

**Комплект оценочных средств**

**для проведения, текущего, рубежного контроля и промежуточной  
аттестации в форме дифференцированного зачета**

**по МДК 03.02 Инструментальные средства разработки программного обеспечения**  
*(индекс и наименование учебной дисциплины(МДК))*

в рамках программы подготовки специалистов среднего звена (ППССЗ)

по специальности СПО

**09.02.03 «Программирование в компьютерных системах»**

*(код и наименование специальности)*

СОГЛАСОВАНО

Зам. директора по УМР

 Е.В. Кужилева  
01 июля 2021 г.

УТВЕРЖДАЮ

Зам. директора по УР

 Т.В. Трусова  
01 июля 2021 г.

Одобрено

УМО общепрофессиональных  
и специальных дисциплин  
специальностей 09.02.03, 10.02.01, 10.01.03

Протокол от 01 июля 2021 г. №11

Председатель УМО

 О.А. Афиногенова

Комплект оценочных средств для проведения промежуточной аттестации по МДК03.02. Инструментальные средства разработки программного обеспечения разработан на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах (утв. приказом Министерства образования и науки Российской Федерации от 28.07.2014 г. № 804, зарегистрирован в Минюст России от 21.08.2014 г. № 33733), рабочей программы ПМ03 Участие в интеграции программных модулей (утв. директором колледжа), Положения по организации текущего контроля успеваемости и промежуточной аттестации обучающихся ГБПОУ КК НКРП (утв. директором колледжа), Положения по формированию КОС по дисциплине, МДК (утв. директором колледжа)

**Организация-разработчик:** ГБПОУ КК «Новороссийский колледж радиоэлектронного приборостроения» (далее ГБПОУ КК НКРП)

**Разработчик:**

преподаватель ГБПОУ КК НКРП  
(должность, место работы)

 Барилова С.В.  
(подпись)

преподаватель ГБПОУ КК НКРП  
(должность, место работы)

 Крайнева А.А.  
(подпись)

**Рецензенты:**



Черникович Д.В., ведущий инженер-программист  
ПАО «Новороссийское морское пароходство»  
(ФИО, должность место работы)



О.А. Афиногенова, преподаватель высшей аттестационной категории  
ГБПОУ КК НКРП  
(ФИО, должность место работы)

## РЕЦЕНЗИЯ

на комплект оценочных средств МДК.03.02.Инструментальные средства разработки программного обеспечения  
Направление подготовки (специальность) 09.02.03 Программирование в компьютерных системах

Комплект оценочных средств подготовлен Бариловой С.В., Крайневой А.А.

КОС МДК.03.02 Инструментальные средства разработки программного обеспечения разработан на основе рабочей программы профессионального модуля.

Предназначен для подготовки оценочных материалов, обеспечивающих проведение дифференцированного зачета по МДК.

КОС состоит из следующих разделов:

- раздел «Паспорт комплекта оценочных средств», характеризующий область применения и нормативные основания разработки КОС; сводные сведения об объектах оценивания, показателях и критериях оценивания, типах заданий; формах аттестации;
- раздел «Комплект оценочных средств», структура которого позволяет разрабатывать и комплектовать разные типы заданий для обучающихся.

В паспорте указаны знания и умения в соответствии с рабочей программой профессионального модуля, показатели и критерии оценки знаний обучающихся по каждому объекту оценивания, а также формы и методы контроля.

Комплект оценочных средств включает в себя типовые задания для проведения дифференцированного зачета по каждому объекту оценивания.

Задания разработаны конкретно, последовательно, технически грамотно и позволяют проверить знания и умения по профессиональному модулю. Предлагаемый программой перечень заданий обеспечивает приобретение умений и навыков у студентов.

Таким образом, КОС МДК.03.02 Инструментальные средства разработки программного обеспечения обеспечивает проведение промежуточного контроля знаний студентов и может быть использована в учебном процессе Новороссийского колледжа радиоэлектронного приборостроения.

**Рецензент:**

Черникевич Д.В.



ведущий инженер-программист  
ЦАО «Новороссийское морское пароходство»  
(должность, место работы)

01 июля 2021 г.

## РЕЦЕНЗИЯ

на комплект оценочных средств МДК.03.02 Инструментальные средства разработки программного обеспечения  
Направление подготовки (специальность) 09.02.03 Программирование в компьютерных системах

Комплект оценочных средств подготовлен Бариловой С.В., Крайневой А.А.

Комплект оценочных средств МДК.03.02 Инструментальные средства разработки программного обеспечения разработан на основе рабочей программы профессионального модуля.

КОС состоит из следующих разделов:

- раздел «Паспорт комплекта оценочных средств», характеризующий область применения и нормативные основания разработки КОС; сводные сведения об объектах оценивания, показателях и критериях оценивания, типах заданий; формах аттестации;
- раздел «Комплект оценочных средств», структура которого позволяет разрабатывать и комплектовать разные типы заданий для обучающихся.

В паспорте указаны знания и умения в соответствии с рабочей программой профессионального модуля, показатели и критерии оценки знаний обучающихся по каждому объекту оценивания, а также формы и методы контроля.

Комплект оценочных средств включает в себя типовые задания для проведения дифференцированного зачета по каждому объекту оценивания.

Задания разработаны конкретно, последовательно, технически грамотно и позволяют проверить знания и умения по профессиональному модулю, а также сформированность соответствующих профессиональных компетенций.

КОС МДК.03.02 Инструментальные средства разработки программного обеспечения может быть использован в учебном процессе ГБПОУ КК «Новороссийского колледжа радиоэлектронного приборостроения».

Рецензент:

  
подпись

О.А. Афиногенова

расшифровка

преподаватель высшей аттестационной категории ГБПОУ КК НКРП

01 июля 2021 г.

# 1 Паспорт комплекта оценочных средств

## 1.1 Область применения комплекта оценочных средств

Комплект оценочных средств (КОС) предназначен для оценки результатов освоения МДК.03.02 Инструментальные средства разработки программного обеспечения.

КОС включает контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме дифференцированного зачета.

<b>Результаты освоения (объекты оценивания)</b>	<b>Основные показатели оценки результата и их критерии</b>	<b>Тип задания; № задания</b>	<b>Форма аттестации (в соответствии с учебным планом)</b>
<b>уметь:</b>			
<i>строить концептуальную модель предметной области</i>	<b>Показатель:</b> построение концептуальной модели предметной области; <b>Критерий:</b> Концептуальная модель соответствует описанию предметной области и требованиям, определенным в практических заданиях	Практические задания	Текущий контроль при выполнении практических занятий №8-19, 23-28 Дифференцированный зачет
<i>проектировать интерфейс пользователя</i>	<b>Показатель:</b> проектирование интерфейса пользователя; <b>Критерий:</b> Разработанный интерфейс соответствует правилам и требованиям, определенным в практических заданиях	Практические задания	Текущий контроль при выполнении практических занятий №20-22 Дифференцированный зачет
<b>знать:</b>			
принципы построения, структуры и приемы работы с	Изложение принципов и приемов работы с инструментальными	Устный опрос	Текущий контроль

инструментальными средствами, поддерживающими создание программного обеспечения;	средствами, поддерживающими создание программного обеспечения;		
методы организации работы в коллективах разработчиков программного обеспечения;	Перечисление методов работы в коллективах разработчиков программного обеспечения;	Тестирование	Текущий контроль
<i>инструментальные средства поддержки тестирования</i>	Перечисление и описание средств поддержки тестирования	Устный опрос	Текущий контроль
<i>инструментальные средства поддержки процесса документирования</i>	Перечисление и описание средств поддержки процесса документирования	Тестирование	Текущий контроль

## 2 Комплект оценочных средств

### 2.1 Задания для проведения контроля практических занятий:

#### Практическое занятие 1 Планирование задач проекта

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков создания проекта, настройки его календаря, ввода перечня работ и задания их параметров

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

#### **Прядок выполнения заданий**

#### **Создание остальных связей проекта Разработка Программы**

Используя рассмотренные выше методы создать остальные связи проекта в соответствии с табл.1.Связи между задачами.

№	Название	Предшественники	Длительность
1	Начало реализации проекта		–
2	<b>Программирование</b>		–
3	Постановка задачи	1	10
4	Разработка интерфейса	3	5
5	Разработка модулей обработки данных	4	7
6	Разработка структуры базы данных	3	6
7	Заполнение базы данных	6	8
8	Программирование завершено	5;7	–
9	<b>Отладка</b>		–
10	Отладка программного комплекса	8	5
11	Тестирование и исправление ошибок	10	10
12	Составление программной документации	10	5

13	Отладка завершена	11;12	–
14	Конец проекта	13	–

### Типы связей, задержки, опережения и ограничения

1. Двойной щелчок мыши по строке задачи Тестирование и исправление ошибок в таблице.
2. В открывшемся окне сведений о задаче выбрать вкладку Предшественники.
3. В строке предшественника Отладка программного комплекса изменить значение поля Тип на Начало-начало, а в поле Запаздывание установить 3д (отрицательное значение означает задержку).
4. Нажать ОК.
5. Двойной щелчок мыши по строке задачи Составление программной документации.
6. В открывшемся окне сведений о задаче выбрать вкладку Дополнительно.
7. В поле Тип выбрать ограничение Как можно позже.
8. Нажать ОК.

### Ввод длительности задач

Ввод длительности задач выполняется в столбце Длительность таблицы диаграммы Ганта или в одноимённом поле, расположенном на вкладке Общие окна сведений о задаче. Необходимо самостоятельно задать длительности задач проекта Разработка Программы в соответствии с табл.1.2.

### Ввод даты начала проекта

1. Выбрать пункт меню Проект/Сведения о проекте.
2. В открывшемся окне в поле ДатаНачала ввести дату 07.09.2021.
3. Нажать ОК.

### Практическая часть:

Создать проект Внедрение бухгалтерской системы, предназначенный для автоматизации бухгалтерии небольшого предприятия, состоящей из 10-ти человек. Дата начала проекта – 1 июля 2021 года. Перечень задач проекта, их связи и длительности приведены в табл.1.3. Фазы выделены полужирным курсивом, а вехи имеют нулевую длину. Названия задач, входящих в фазу, выделены отступом слева.

№	Название задачи	Длит. (дн)	Предшественники
1	Начало проекта	0	
2	<b>Выбор системы</b>		
3	Изучение рынка бухгалтерских систем	7	1
4	Составление требований к бухгалтерским системам	7	1
5	Консультации с фирмами-разработчиками	7	3; 4
6	Принятие окончательного решения	2	5
7	Выбор завершён	0	6
8	<b>Приобретение программного обеспечения</b>		
9	Заклучение договоров	6	7
10	Оплата за ПО	2	9
11	Оформление ПО на баланс	3	10
12	Приобретение ПО завершено	0	11
13	<b>Составление проекта сети</b>		
14	Разработка архитектуры сети	7	7
15	Проработка физического размещения сети	5	14
16	Проект сети завершён	0	15
17	<b>Приобретение компьютеров и сетевого оборудования</b>		
18	Сбор информации о поставщиках и предложениях	7	7
19	Анализ и выбор поставщика	5	14; 18
20	Заклучение договоров	5	19
21	Оплата за оборудование	2	20
22	Оформление оборудования на баланс	3	21

23	Приобретение оборудования завершено	0	22
24	<b>Обучение администратора и программиста</b>		
25	Курсы администраторов	18	16
26	Курсы программистов	18	12
27	Сдача сертификац. экзаменов	3	25; 26
28	Обучение завершено	0	27
№	Название задачи	Длит. (дн)	Предшественники
29	<b>Монтаж локальной сети</b>		
30	Установка компьютеров на рабочих местах	3	23; 28
31	Монтаж кабеля	10	23; 28
32	Монтаж сетевых устройств	10	23; 28
33	Подключение кабеля к компьютерам и сетевым устройствам	5	30; 31; 32
34	Монтаж завершён	0	33
35	<b>Установка ПО на компьютеры</b>		
36	Установка сервера	5	34
37	Создание доменов и пользователей	7	36
38	Проверка и настройка работы сети	5	37
39	Настройка сети завершена	0	38
40	<b>Ввод начальных данных</b>		
41	Ввод справочников	40	39
42	Ввод начальных остатков	40	41
43	Ввод начальных данных завершён	0	42
44	<b>Обучение персонала</b>		
45	Принципы работы системы	3	39
46	Изучение интерфейса	5	45
47	Изучение справочников	20	41; 46
48	Изучение документов и журналов	30	42; 47
49	Обучение завершено	0	48
50	<b>Передача в эксплуатацию</b>		
51	Формирование тестовой отчётности	5	49
52	Акт ввода в эксплуатацию	3	51
53	Передача в эксплуатацию завершена	0	52
54	Конец проекта	0	53

2. Между задачами 10 и 11 установить задержку в 5 дней, необходимую для прохождения безналичной оплаты.

3. Между задачами 21 и 22 установить задержку в 7 дней, необходимую для прохождения безналичной оплаты и доставки оборудования.

4. Установить тип связи между задачами 41 и 47 Начало-начало и задержку в 5 дней.

5. Установить ограничение для задачи 42 ограничение Не ранее 1.01.2021.

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 2      Использование таблиц и представлений

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков использования таблиц и представлений проекта: форматирования, сортировки, группировки и фильтрации таблиц.

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

### Прядок выполнения заданий

#### Создание остальных связей проекта Разработка Программы

Примеры обработки таблиц и представлений иллюстрируются с использованием проекта Разработка программы, содержащегося (практическое занятие 1).

#### Выбор таблицы

1. Переключиться в диаграмму Ганта – Вид/Диаграмма Ганта.
2. Увеличить область отображения таблицы, переместив мышью вправо границу между левой и правой частями представления.
3. По умолчанию на экране расположена таблица Ввод. Выведем таблицу Затраты – Вид/Таблица/Затраты. Эта таблица содержит данные о стоимости работ проекта.
4. Вернём первоначальную таблицу – Вид/Таблица/Ввод.

#### Удаление столбца

1. Щелкнуть правой кнопкой мыши по заголовку столбца идентификаторов задач, отображающему значок .
2. В появившемся контекстном меню выбрать Скрыть столбец.
3. Вернуть этот столбец, отменив последнее действие (Правка/Отменить скрыть столбец или кнопка на панели инструментов).

#### Добавление столбца

1. Щелкнуть правой кнопкой мыши по заголовку столбца Длительность.
2. Из контекстного меню выбрать пункт Вставить столбец.
3. В открывшемся диалоге в поле Имя поля выбрать Затраты.
4. Нажать *ОК*.

#### Назначение стилей текста

1. Выбрать пункт меню Формат/Стили текста.
2. В поле Изменяемый элемент установить элемент таблицы или представления, подлежащий форматированию – Заголовки строк и столбцов.
3. Установить способ начертания – Жирный.
4. Нажать *ОК*. Результат – заголовки строк и столбцов прописаны полужирным шрифтом.
5. Формат/Стили текста.
6. Изменяемый элемент – Суммарные задачи.
7. Цвет – малиновый.
8. Изменяемый элемент – Нижний уровень шкалы времени.
9. Цвет – тёмно-синий.

10. ОК. Результат – различные виды задач отображаются заданными цветами текста.

### **Предопределённая сортировка**

1. Пункт меню Проект/Сортировка/По дате начала.
2. Результат – задачи в таблице упорядочены по датам начала.
3. Отмена сортировки – Проект/Сортировка/По идентификатору.

### **Сортировка пользователя**

1. Выбрать пункт меню Проект/Сортировка/Сортировать по.  
2. В открывшемся окне (рис.1.2) установить уровни сортировки (каждый последующий уровень действует только в пределах одинаковых значений предыдущего): Сортировать по – Критическая задача, Затем по – Длительность.

3. Убрать флажок Сохранить структуру (при установленном значении этого флажка задачи сортируются только в пределах своих фаз).

4. Нажать Сортировать. Результат – все задачи отсортированы по признаку критической задачи, а в пределах этого признака – по длительности.

5. Для проверки правильности сортировки вставим в таблицу столбец Критическая задача. Результат – первая группа задач имеет значение Нет в этом столбце, а вторая группа – Да. В пределах групп задачи упорядочены по значениям длительности.

6. Отменим сортировку – Проект/Сортировка/По идентификатору и удалим столбец Критическая задача.

7. Выполним аналогичную сортировку с установленным флажком Сохранить структуру. Результат – задачи сортируются только в пределах своих фаз.

8. Отменим сортировку.

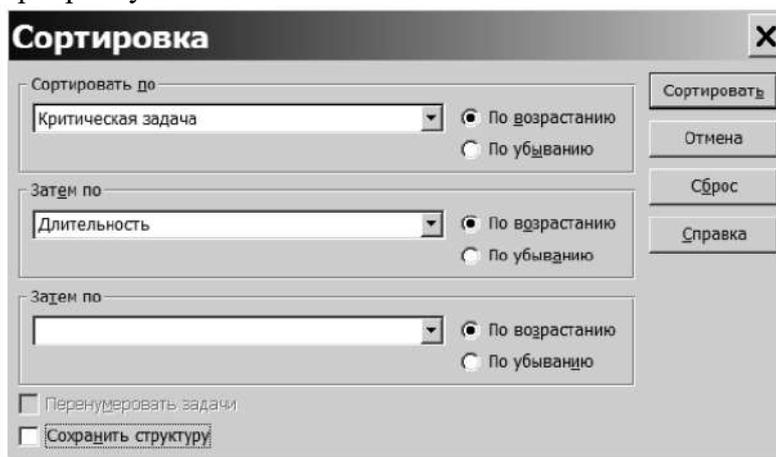


Рис.1.2. Окно параметров сортировки

### **Использование структурного фильтра**

1. Назначим структурный фильтр – Проект/ Структура/Показать/ Уровень 1. Результат – таблица содержит только задачи уровня 1. К ним относятся начальная и конечная вехи и обе фазы.

2. Назначим другой структурный фильтр – Проект/ Структура/ Показать/ Уровень 2. Результат – таблица содержит задачи первого и второго уровней. В этом проекте к ним относятся все задачи.

3. Отменим структурный фильтр – Проект/ Структура/ Показать/ Все подзадачи.

### **Использование автофильтра**

1. Включить автофильтр – Проект/Фильтр/Автофильтр.

Результат – в заголовках столбцов таблицы появились кнопки автофильтра для назначения фильтра по отдельному столбцу. При нажатии мышью такой кнопки появляется список условий фильтрации.

2. Отобразим в таблице только задачи с длительностью от 1 до 5 дней.

а. Щелкнуть мышью по кнопке автофильтра в столбце Длительность.

б. В списке условий выбрать пункт (Условие..). Откроется окно для определения условия фильтрации, изображённое на рис. 1.3.

- c. Заполнить поля этого окна в соответствии с рис.1.3.
- d. Нажать ОК. Результат – таблица содержит только задачи с длительностью от 1 до 5 дней и фазы, в которые входят эти задачи.

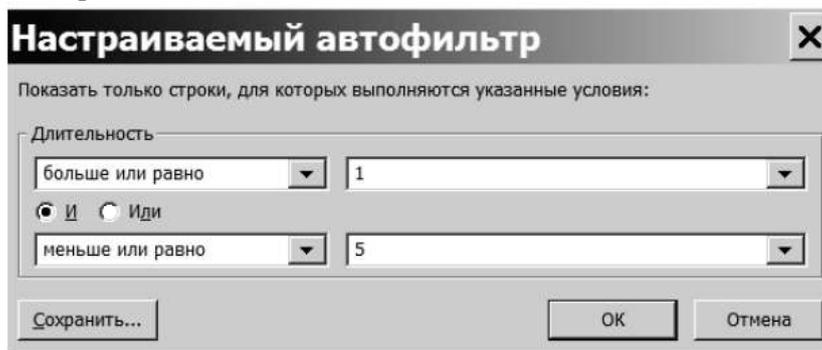


Рис.1.3 Окно условия автофильтра

3. Отменить фильтрацию. В списке условий автофильтра по столбцу Длительность выбрать пункт Все. Результат – таблица содержит все задачи.
4. Назначить автофильтр по столбцу Начало. В качестве условия фильтрации выбрать Текущая неделя. Результат – в таблице нет ни одной задачи, поскольку начало ни одной из задач проекта не приходится на текущую неделю.
5. Отменить фильтрацию и убрать кнопки автофильтра – Проект/Фильтр/Автофильтр. Результат – таблица снова содержит все задачи проекта.

#### **Использование предопределённого фильтра**

1. Отобразить только фазы – Проект/Фильтр/Суммарные задачи. Результат – таблица содержит только фазы Программирование и Отладка.
2. Отобразить только вехи – Проект/Фильтр/Вехи. Результат – таблица содержит только вехи и фазы, в которых эти вехи находятся.
3. Отобразить только задачи критического пути – Проект/Фильтр/Критические задачи.
4. Отобразить только задачи с номерами из заданного диапазона – Проект/Фильтр/Диапазон задач. В открывшемся окне ввести диапазон номеров задач – от 5 до 7. Нажать ОК.
5. Отобразить задачи, которые начинаются или заканчиваются в указанном диапазоне дат – Проект/Фильтр/Диапазон дат. Сначала откроется окно для задания начальной даты диапазона. Установить 01.07.21. ОК. Затем откроется окно для ввода конечной даты диапазона. Установить 10.07.21. ОК. Проанализировать результат.
6. Отменить предопределённый фильтр – Проект/Фильтр/Все задачи.

#### **Использование определённого пользователем фильтра**

1. Создать новый фильтр пользователя.
  - a. Проект/Фильтр/Другие фильтры. Результат – откроется окно со списком определённых в системе фильтров.
  - b. Нажать Создать – откроется окно определения фильтра, изображённое на рис. 1.4.
2. Заполнить поля значениями в соответствии с этим рисунком. ОК.
3. В списке фильтров окна Другие фильтры появилась строка КороткаяКритическаяЗадача. Закрыть это окно кнопкой Закрыть.
4. Назначить созданный фильтр – Проект/ Фильтр/ КороткаяКритическаяЗадача. Результат – таблица содержит только критические задачи длительностью не более 5 дней.
5. Отменить действие фильтра – Проект/Фильтр/Все задачи.
6. Удалить созданный фильтр из системы.
  - a. Проект/Фильтр/Другие фильтры. Откроется окно со списком фильтров.
  - b. Нажать кнопку Организатор.
  - c. В правом списке выделить строку КороткаяКритическаяЗадача.
  - d. Нажать кнопку Удалить, после чего подтвердить удаление.
  - e. Закрыть окна организатора.

f. В окне списка фильтров убедиться в том, что удалённого фильтра нет в списке. Закрывать это окно.

g. Проверить состав предопределённых фильтров в пункте меню Проект/Фильтр – из списка исчез удалённый фильтр.

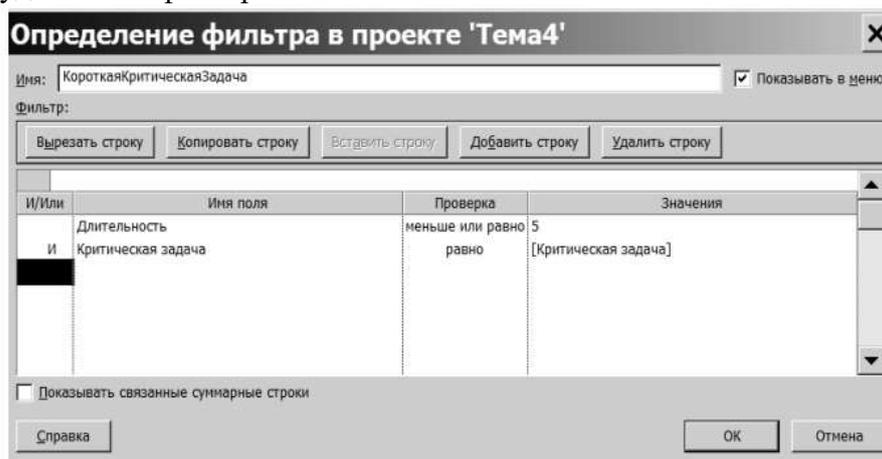


Рис.1.4 Окно определения фильтра

### Использование предопределённой группировки

1. Проект/Группировка/Вехи – задачи таблицы сгруппированы в две группы. Первая содержит вехи, а вторая – задачи, не являющиеся вехами.

2. Проект/Группировка/Критические задачи – задачи сгруппированы в две группы. Первая состоит из задач, не лежащих на критическом пути, а во вторую входят все задачи критического пути.

3. Проект/Группировка/Длительности – задачи сгруппированы в шесть групп в соответствии с их длительностью.

4. Отменить действие группировки – Проект/Группировка/Нет группировки.

### Использование созданной пользователем группировки

1. Создать новую группировку.

a. Проект/Группировка/Другие группы – откроется окно Другие группы со списком определённых в системе группировок.

b. Нажать кнопку Создать – откроется окно определения группы, изображённое на рис. 1.5.

2. Заполнить поля в соответствии с изображением на этом рис. 1.5.

3. Зададим группировку по длительности с интервалом, равным неделе.

a. Выбрать при помощи мыши имя поля Длительность, как это сделано на рис. 1.5.

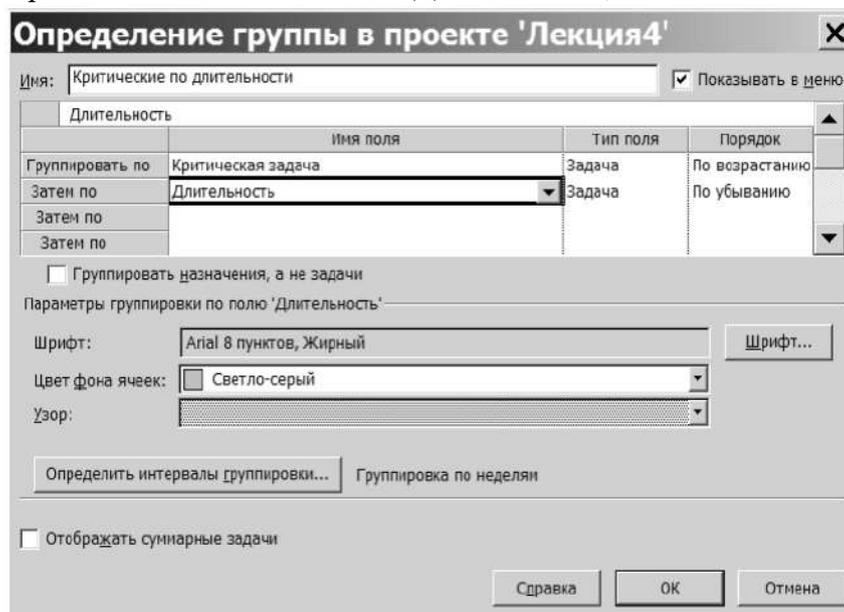


Рис.1.5 Окно создания группировки

b. Нажать кнопку Определить интервалы группировки – откроется окно определения интервала.

c. Установить поле Группировать в значение По неделям. Нажать ОК.

4. Завершить определение группировки – ОК.

5. Убедиться, что в окне Другие группы в списке присутствует созданная группировка. Закрыть это окно.

6. Использовать созданную группировку – Проект/ Группировка/Критические по длительности. Задачи сгруппированы в две группы по признаку Критическая задача, а внутри этих групп – по значению длительности с интервалом в неделю.

7. Отменить действие группировки – Проект/Группировка/Нет группировки.

8. Удалить созданную группировку.

a. Проект/Группировка/Другие группы.

b. Нажать кнопку Организатор.

c. В правом списке выделить название КритическаяДлительностиПоНеделям.

d. Нажать Удалить и подтвердить удаление.

e. Закрыть окно организатора и окно Другие группы.

f. Убедиться, что удалённая группировка отсутствует в списке пункта меню Проект/Группировка.

### **Использование временной группировки**

1. Открыть окно для создания временной группировки – Проект/Группировка/ Настройка группировки.

2. Выполнить пп.2 – 4 предыдущей последовательности действий.

3. Отменить действие группировки – Проект/Группировка/Нет группировки

### **Практическая часть:**

При выполнении заданий используются варианты проектов, созданных при выполнении практического занятия № 1.

1. Выполнить форматирование таблицы ввода диаграммы Ганта ранее созданного проекта согласно своему варианту.

a. Удалить столбец идентификаторов.

b. Добавить столбец Критическая задача.

c. Изменить столбец Критическая задача на столбец Затраты.

d. Назначить стили текста, выделив разными цветами фазы, вехи, критические и некритические задачи. При помощи стилей текста установить для заголовков строк и столбцов жирный шрифт коричневого цвета, а для среднего уровня шкалы времени сиреневого цвета.

e. Сохранить изменения в файле ПЗ2\_1.mpp.

2. В первоначальном файле проекта выполнить сортировку задач таблицы по дате начала, а затем по дате окончания. Результат сохранить в файле ПЗ2\_2.mpp.

3. В первоначальном файле проекта выполнить многоуровневую сортировку таблицы сначала по возрастанию признака критической задачи, а затем по убыванию даты окончания. Результат сохранить в файле ПЗ2\_3.mpp.

4. В первоначальном файле проекта, используя структурный фильтр, отобразить только задачи 1-го уровня. Результат сохранить в файле ПЗ2\_4.mpp.

5. В первоначальном файле проекта, используя автофильтр, отобразить задачи, которые начинаются в следующем месяце и имеют длительность больше 15-ти дней. Результат сохранить в файле ПЗ2\_5.mpp.

6. В первоначальном файле проекта, используя предопределённый фильтр, отобразить только суммарные задачи. Результат сохранить в файле ПЗ2\_6.mpp.

7. В первоначальном файле проекта создать фильтр, который отображает только критические задачи длительностью не более 14-ти дней. Поместить этот фильтр в меню системы. Применить этот фильтр. Результат сохранить в файле ПЗ2\_7.mpp.

### **Критерии оценивания:**

оценка «5» ставится, если:

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

оценка «4» ставится, если:

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

оценка «3» ставится, если:

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

оценка «2» ставится, если:

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

### Практическое занятие 3 Создание ресурсов и назначений

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков формирования списка ресурсов, ввода их параметров и создания назначений ресурсов.

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

#### Прядок выполнения заданий

Примеры создания ресурсов и назначений иллюстрируются с использованием проекта Разработка программы (предыдущие занятия).

**Ввод списка ресурсов**

1. Открыть проект Разработка Программы.
2. Выбрать пункт меню Вид/Лист ресурсов. Результат – откроется одноимённое представление, изображённое на рис. 1.2.

Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на исполыз.	Начисление	Базовый календарь
Постановщик	Трудовой		П		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Программист1	Трудовой		Пр1		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Программист2	Трудовой		Пр2		100%	0,00р./ч	0,00р./ч	0,00р.	Пропорциональное	Стандартный
Бумага	Материальный		Б			0,00р.		0,00р.	Пропорциональное	
Междугородные переговоры	Затраты		М						Пропорциональное	

Рис.1.2 Лист ресурсов проекта

3. Щёлкнуть мышью по полю Название ресурса первой пустой строки и ввести название Постановщик.
4. Аналогично добавить в таблицу ресурсы Программист1 и Программист2.
5. Добавить в таблицу ресурс Бумага и CD-матрица и выбрать для них тип Материальный.
6. Добавить в таблицу ресурс Междугородные переговоры и выбрать для него тип

Затраты.

### Ввод свойств ресурса Постановщик

1. Дважды щёлкнуть мышью по строке Постановщик таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие
3. В поле Краткое название ввести Пост.
4. В таблицу доступности ресурса ввести две строки:
  - Доступен с – НД; Доступен по – 27.09.21; Единицы – 100 %.
  - Доступен с – 19.10.21; Доступен по – НД; Единицы – 100 %.
5. Нажать кнопку Изменить рабочее время – откроется индивидуальный календарь ресурса.
6. Установим четырёхдневную рабочую неделю. Для этого выберем вкладку Рабочие недели, в списке рабочих недель – строку По умолчанию, и нажмём кнопку Подробности.

7. Откроется окно сведений о рабочей неделе (рис. 1.3), в котором нужно выделить день недели Пятница и для него выбрать переключатель Задать нерабочие дни. Закрыть оба окна нажатием ОК. Результат – по пятницам Постановщик является выходным и в работах проекта не участвует.

8. Выбрать вкладку Затраты. В столбец Стандартная ставка первой строки ввести 50000 р./мес, а в столбец Ставка сверхурочных – 300 р./ч. Предположим, что с 01.01.22 зарплата данного работника должна быть увеличена. Во второй строке задать дату начала действия новой ставки – 01.01.22, стандартная ставка – 60000 р./мес, ставка сверхурочных – 350 р./ч. Для ввода значения удобнее всего пользоваться строкой ввода, расположенной над таблицей.

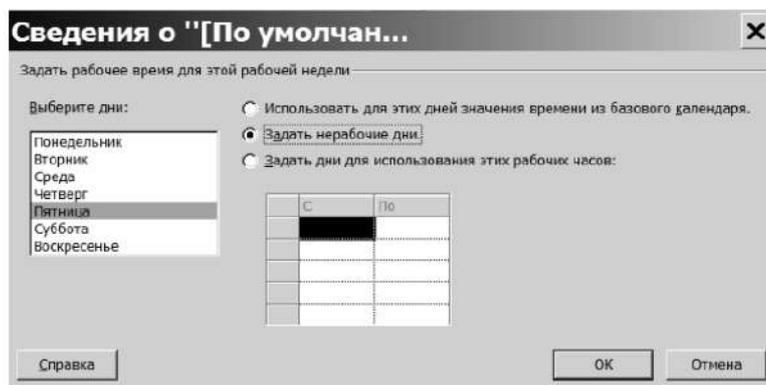


Рис.1.3 Вкладка Общие окна свойств ресурса

9. Выбрать таблицу норм затрат *B* и установить затраты на использование – 25000 р. Эту норму можно использовать для работ с фиксированной суммой оплаты работника.

10. Нажать *ОК*.

### Ввод свойств ресурса Программист1

1. Дважды щёлкнуть мышью по строке Программист1 таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие.
3. В поле Краткое название ввести Прог1.
4. Выбрать вкладку Затраты. Предположим, что 01.01.2022 оплата этого ресурса будет увеличена. Мы должны ввести две строки в таблицу норм затрат *A*:
  - Дата действия – «-», Стандартная ставка – 65000 р./мес, Ставка сверхурочных – 500 р./ч.
  - Дата действия – 01.01.22, Стандартная ставка – 70000 р./мес, Ставка сверхурочных – 510 р./ч.

5. Выбрать таблицу норм затрат *B* и установить затраты на использование – 35000р. Эту норму можно использовать для работ с фиксированной суммой оплаты работника. Нажать *ОК*.

### Ввод свойств ресурса Программист2

1. Дважды щёлкнуть мышью по строке Программист2 таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие.
3. В поле Краткое название ввести Прог2.
4. Выбрать вкладку Затраты. В столбец Стандартная ставка ввести 70000 р./мес, а в столбец Ставка сверхурочных – 500 р./ч.
5. Выбрать таблицу норм затрат *B* и установить затраты на использование – 35000 р. Нажать *OK*.

#### **Ввод свойств ресурса Бумага**

1. Дважды щёлкнуть мышью по строке Бумага таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие.
3. В поле Краткое название ввести Бум, а в поле Единицы измерения – пачка.
4. Выбрать вкладку Затраты. В столбец Стандартная ставка ввести 200 р. Нажать

*OK*.

#### **Ввод свойств ресурса CD-матрица**

1. Дважды щёлкнуть мышью по строке CD-матрица таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие.
3. В поле Краткое название ввести CD-R, а в поле Единицы измерения – штука.
4. Выбрать вкладку Затраты. В столбец Стандартная ставка ввести 10 р. Нажать

*OK*.

#### **Ввод свойств ресурса Междугородные переговоры**

1. Дважды щёлкнуть мышью по строке Бумага таблицы ресурсов.
2. В открывшемся окне выбрать вкладку Общие.
3. В поле Краткое название ввести Межгород. Нажать *OK*.

#### **Ввод назначений для задач проекта**

1. Выбрать пункт меню Вид/Диаграмма Ганта.
2. Двойной щелчок мыши по строке Постановка задачи.
3. В открывшемся окне свойств задачи выбрать закладку Ресурсы.
4. В таблицу ресурсов добавить записи: Постановщик – 100, Бумага–1, CD-матрица – 1штука/н, Междугородные переговоры – 1000 р. (рис. 1.4).
5. Нажать *OK*.
6. Ввести назначения в соответствии с табл. 1.1

#### **Назначение профиля загрузки**

1. Выбрать пункт меню Вид/Использование задач.
2. Двойной щелчок мыши по назначению Постановщик в задаче Постановка задачи.
3. В открывшемся окне выбрать вкладку Общие и в поле Профиль загрузки установить значение Загрузка в конце.
4. Нажать *OK*.
5. Результат – длительность задачи выросла с 10 до 16,67 дней.

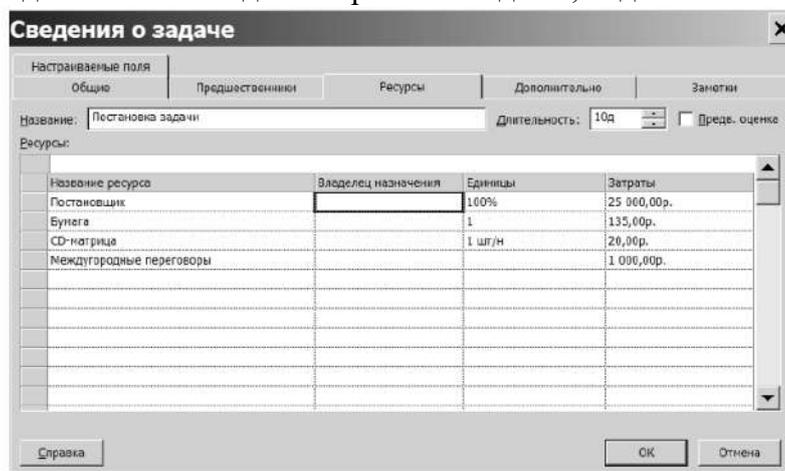


Рис.1.4 Пример назначения материальных и затратных ресурсов

Таблица 1.1 Назначения ресурсов для проекта примера

№	Название задачи	Ресурсы	Единицы %	Затраты руб
4	Разработка интерфейса	Программист1 CD-матрица М. переговоры	100 1шт./д	500
5	Разработка модулей обработки данных	Программист1 CD-матрица	100 1шт./д	
6	Разработка структуры базы данных	Программист2 CD-матрица М. переговоры	100 1шт./д	1500
7	Заполнение базы данных	Программист2 CD-матрица	100 1шт./д	
10	Отладка программного комплекса	Постановщик Программист1 Программист2 CD-матрица	100 100 100 2шт./д	
11	Тестирование и исправление ошибок	Постановщик Программист1 Программист2 CD-матрица	100 100 100 2шт./д	
12	Составление программной документации	Постановщик CD-матрица М. переговоры	100 10 шт.	2000

### Ручное изменение графика трудозатрат

1. Выбрать пункт меню Вид/Использование задач.
2. Проанализируем график трудозатрат постановщика при работе над постановкой задачи. В последний день работы (5 октября) ему запланировано 5,33 ч (рис. 1.5). Распределим эти трудозатраты с 1-го по 5-й дни работы, где первоначально запланировано, соответственно, 0,8 ч, 0,93 ч, 1,2 ч, 1,73 ч, 2 ч (т.е. на 7, 8, 9, 10, 14 сентября соответственно).
3. Щелчок мышью по ячейке трудозатрат 5.10.09. Ввести 0, нажать Enter.
4. Ввести в ячейки трудозатрат 1 – 5-го дней работы (т.е. 7, 8, 9, 10, 14 сентября) цифры: 1,8; 1,93; 2,2; 2,73; 3,33. Результат – длительность задачи сократилась до 16 дней, трудозатраты с 80 ч уменьшились до 79,98. Снижение трудозатрат в колонке Трудозатраты на 0,02 связано с ошибками округления.
5. Ввести в ячейку трудозатрат 14 сентября число 3,34 для компенсации ошибок округления. Результат – суммарные трудозатраты – 80 ч.
6. Общим результатом выполненной коррекции является уменьшение длительности задачи Постановка задачи на один день при сохранении структуры профиля загрузки Загрузка в конце.

Название задачи	Трудозатраты	Длительность	Начал	Подробности	05 Окт '09							
					С	Ч	П	С	В	П	В	
Начало реализации прое	0 ч	0 дней	Пн 07.0	Трудозатр.								
▣ Программирование	288 ч	35,67 дней	Пн 07.0	Трудозатр.	8ч	8ч				10,67ч	16ч	
▣ Постановка задачи	80 ч	16,67 дней	Пн 07.0	Трудозатр.	8ч	8ч				5,33ч		
Постановщик	80 ч		Пн 07.0	Трудозатр.	8ч	8ч				5,33ч		
Бумага	1 пачка		Пн 07.0	Трудозатр. (п.								
Междугородные			Пн 07.0	Трудозатр.								
CD-матрица	2 штука		Пн 07.0	Трудозатр. (ц.								
▣ Разработка интерфей	40 ч	5 дней	Пн 05.1	Трудозатр.						2,67ч	8ч	
Программист1	40 ч		Пн 05.1	Трудозатр.						2,67ч	8ч	
Междугородные			Пн 05.1	Трудозатр.								
CD-матрица	5 штука		Пн 05.1	Трудозатр. (ц.						0,33	1	
▣ Разработка модулей	48 ч	6 дней	Ср 14.1	Трудозатр.								
Программист1	48 ч		Ср 14.1	Трудозатр.								
CD-матрица	6 штука		Ср 14.1	Трудозатр. (ц.								
▣ Разработка структуры	56 ч	7 дней	Пн 05.1	Трудозатр.						2,67ч	8ч	
Программист2	56 ч		Пн 05.1	Трудозатр.						2,67ч	8ч	
Междугородные			Пн 05.1	Трудозатр.								

Рис.1.5. Редактирование графика загрузки

### Изменение норм затрат в назначении

1. В представлении Использование задач дважды щёлкнуть мышью по назначению Постановщик в задаче Тестирование и исправление ошибок. Откроется окно свойств назначения.

2. На вкладке Общие выбрать Таблица норм затрат – В.

3. Нажать ОК.

4. Аналогично установить таблицу норм затрат В для назначений Программист1 и Программист 2 этой же задачи.

5. Результат – расчёт затрат этих ресурсов для выполнения данной задачи выполняется по таблице норм В, в которой предусмотрена сдельная расценка для всей задачи: Постановщик – 25000 р, Программист1 и Программист2 – по 35000 р.

### Практическая часть:

1. Для проекта Внедрение бухгалтерской системы, созданного ранее при выполнении практического занятия 1, создать список ресурсов в соответствии с параметрами, перечисленными в табл.1.2.

Таблица 1.2 Назначения самостоятельного создания

Название ресурса	Тип	Затраты			Затраты на исп.
		Таблица норм	Станд. ставка	Ставка сверхур.	
Главбух	Т	А В	90000 р./мес	500 р./ч	30000 р
Администратор	Т	А В	70000 р./мес	450 р./ч	40000 р
Программист	Т	А В	60000 р./мес	400 р./ч	50000 р
Техник	Т	А	40000 р./мес	250 р./ч	
Расчётчик1	Т	А	40000 р./мес	250 р./ч	–
Расчётчик2	Т	А	40000 р./мес	250 р./ч	–
Расчётчик3	Т	А	40000 р./мес	250 р./ч	–
Бухгалтер мат. учёта1	Т	А	40000 р./мес	250 р./ч	–
Бухгалтер мат. учёта2	Т	А	40000 р./мес	250 р./ч	–
Бухгалтер учёта ОС и НМА	Т	А	40000р./мес	250р./ч	–
Бухгалтер учёта ОС	Т	А	40000 р./мес	250 р./ч	–
Бухгалтер учёта реализации	Т	А	40000 р./мес	250р./ч	–
Бухгалтер учёта ОС	Т	А	40000 р./мес	250 р./ч	–
Бухгалтер учёта реализации	Т	А	40000 р./мес	250р./ч	–
Бухгалтер производственного учёта	Т	А	40000 р./мес	250 р./ч	–
Компьютер	М	А	15000 р.		–
Сервер	М	А	50000 р.		–
Принтер	М	А	5000 р.		–
МФУ	М	А	7000 р.		–
Сетевой кабель	М	А	–		15000 р.

Название ресурса	Тип	Затраты			
		Таблица норм	Станд. ставка	Ставка сверх.	Затраты на исп.
Сетевой концентратор	М	А	3000 р.		–
Панель	М	А			10000 р.
Разъёмы и розетки	М	А	–		15000 р.
Бухгалтерская система	М	А	–		100000 р.
Офисный пакет	М	А	–		70000 р.
ОС рабочей станции	М	А	–		60000 р.
Серверная ОС	М	А	–		30000 р.
DVD-матрица	М	А	10		–
Интернет	З				
Междугородние переговоры	З				
Оплата курсов	З				

2. Создать назначения ресурсов в соответствии с табл. 1.3.  
Таблица 1.3. Создаваемые назначения

Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
Изучение рынка бухгалтерских систем	Администратор Интернет	100	А 1500 р.
Составление требований к бухгалтерским системам	Администратор Главбух	100 20	А А
Консультации с фирмами-разработчиками	Администратор Междугородние переговоры Интернет	100	А 2000 р. 1000 р.
Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
Принятие окончательного решения	Администратор	100	А
	Главбух	100	А
Заключение договоров	Администратор	100	А
	Программист	100	А
	Главбух	100	А
Оплата за ПО	Главбух	10	А
	Бухгалтерская система		А
	Офисный пакет		А
	ОС рабочей станции		А
	Серверная ОС		А
Оформление ПО на баланс	Бухгалтер учёта ОС и НМА	30	А
Разработка архитектуры сети	Администратор	100	А
	Программист	100	А
	Техник	50	А
Проработка физического размещения сети	Администратор	100	А
	Программист	100	А
	Техник	100	А
Сбор информации о поставщиках и предложениях	Администратор	50	А
	Интернет		1000 р.
	Междугородние переговоры		1500 р.
Анализ и выбор поставщика	Администратор	50 20	А
	Главбух		А
	Интернет		1000 р.
Заключение договоров	Администратор	100	А
	Главбух	50	А
Оплата за оборудование	Главбух	30	А
	Компьютер	12	А
	Сервер	1	А
	Принтер	2	А
	МФУ	2	А

Оплата за оборудование	Сетевой концентратор Панель Разъёмы и розетки	2	A A A
Оформление оборудования на баланс	Бухгалтер учёта ОС	70	A
Курсы администраторов	Администратор Оплата курсов	100	A 25000 р.
Курсы программистов	Программист Оплата курсов	100	A 20000 р.
Сдача сертификационных экзаменов	Администратор	100	A
	Программист	100	A
Установка компьютеров на рабочих местах	Техник	100	A
Монтаж кабеля	Техник	100	A
Монтаж сетевых устройств	Техник	100	A
Подключение кабеля к компьютерам и сетевым устройствам	Техник	100	A
Установка сервера	Администратор	100	A
Создание доменов и пользователей	Администратор	100	A
Создание доменов и пользователей	Администратор	100	A
Проверка и настройка работы сети	Администратор	100	A
	Программист	100	A
Ввод справочников	Администратор	100	B
	Программист	100	B
	Расчётчик1	30	A
	Расчётчик2	30	A
	Расчётчик3	30	A
	Бухгалтер мат. учёта1	50	A
	Бухгалтер мат. учёта2	50	A
	Бухгалтер учёта ОС и НМА	50	A
	Бухгалтер учёта ОС	50	A
	Бухгалтер учёта реализации	50	A
	Бухгалтер производственного учёта	50	A
	DVD-матрица	10	A
	Ввод начальных остатков	Администратор	100
Программист		100	B
Главбух		50	A
DVD-матрица		10	A
Принципы работы системы	Администратор	50	A
	Главбух	50	A
	Расчётчик1	50	A
	Расчётчик2	50	A
	Расчётчик3	50	A
	Бухгалтер мат. учёта1	50	A
	Бухгалтер мат. учёта2	50	A
	Бухгалтер учёта ОС и НМА	50	A
	Бухгалтер учёта ОС	50	A
	Бухгалтер учёта реализации	50	A
	Бухгалтер производственного учёта	50	A

Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
Изучение интерфейса	Программист	50	А
	Главбух	50	А
	Расчётчик1	50	А
	Расчётчик2	50	А
	Расчётчик3	50	А
	Бухгалтер мат. учёта1	50	А
	Бухгалтер мат. учёта2	50	А
	Бухгалтер учёта ОС и НМА	50	А
	Бухгалтер учёта ОС	50	А
	Бухгалтер учёта реализации	50	А
Бухгалтер производственного учёта	50	А	
Изучение справочников	Программист	50	А
	Главбух	50	А
	Расчётчик1	50	А
	Расчётчик2	50	А
	Расчётчик3	50	А
	Бухгалтер мат. учёта1	50	А
	Бухгалтер мат. учёта2	50	А
	Бухгалтер учёта ОС и НМА	50	А
	Бухгалтер учёта ОС	50	А
	Бухгалтер учёта реализации	50	А
Бухгалтер производственного учёта	50	А	
Изучение справочников	Программист	50	А
	Главбух	50	А
	Расчётчик1	50	А
	Расчётчик2	50	А
	Расчётчик3	50	А
	Бухгалтер мат. учёта1	50	А
	Бухгалтер мат. учёта2	50	А
	Бухгалтер учёта ОС и НМА	50	А
	Бухгалтер учёта ОС	50	А
	Бухгалтер учёта реализации	50	А
Бухгалтер производственного учёта	50	А	
Изучение документов и журналов	Программист	50	А
	Главбух	50	А
	Расчётчик1	50	А
	Расчётчик2	50	А
	Расчётчик3	50	А
	Бухгалтер мат. учёта1	50	А
	Бухгалтер мат. учёта2	50	А
	Бухгалтер учёта ОС и НМА	50	А
Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
Изучение документов и журналов	Бухгалтер учёта ОС	50	А
	Бухгалтер учёта реализации	50	А
	Бухгалтер производственного учёта	50	А
Формирование тестовой отчётности	Администратор	100	А
	Программист	100	А
Акт ввода в эксплуатацию	Главбух	100	А
	Администратор	50	А
	Главбух	50	А

3. Установить различные профили загрузки для ресурса Техник.

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 4 Анализ проекта

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков анализа проекта, выполнения параметрического и PERT-анализа, анализа рисков.

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

### Прядок выполнения заданий

Примеры анализа проекта иллюстрируются с использованием проекта Разработка программы, содержащегося в файле *Лаба4.mpp*, расположенном в кафедральной локальной сети.

### Создание настраиваемого поля

1. Открыть проект Разработка Программы, содержащийся в файле *Лаба4.mpp*.
2. Откроем окно настройки полей: Сервис/Настройка/Поля (рис.2).
3. Установить: тип поля – флаг, поле – Задача.
4. Выбрать из списка Флаг 1.
5. Кнопка Переименовать – ввести Параметрическая Задача – ОК.

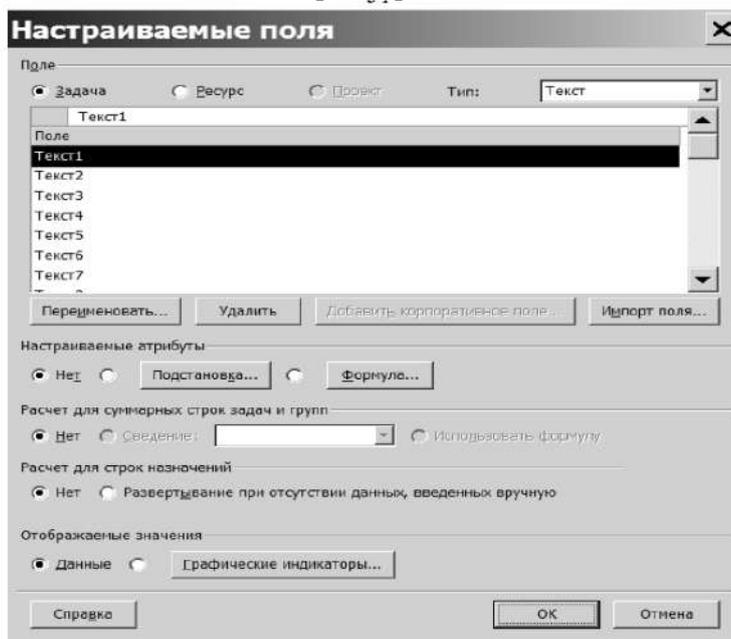


Рис.2 Окна настройки полей

6. Кнопка Графические индикаторы.

7. В открывшейся таблице заполнить 2 строки:

- a) равно – Да – зелёный индикатор;
- b) равно – Нет – красный индикатор (рис. 3).

8. Нажать два раза ОК.

**Заполнение настраиваемого поля**

1. Откроем диаграмму Ганта: Вид/Диаграмма Ганта.

2. В таблицу диаграммы Ганта вставим столбец ПараметрическаяЗадача: Вставка/Столбец – выбрать название столбца – ОК.

3. В появившемся столбце для всех задач задано значение Нет (красный индикатор). Установить Да (зелёный индикатор) для задач Разработка модулей обработки данных, Заполнение базы данных, Составление программной документации (рис. 4).

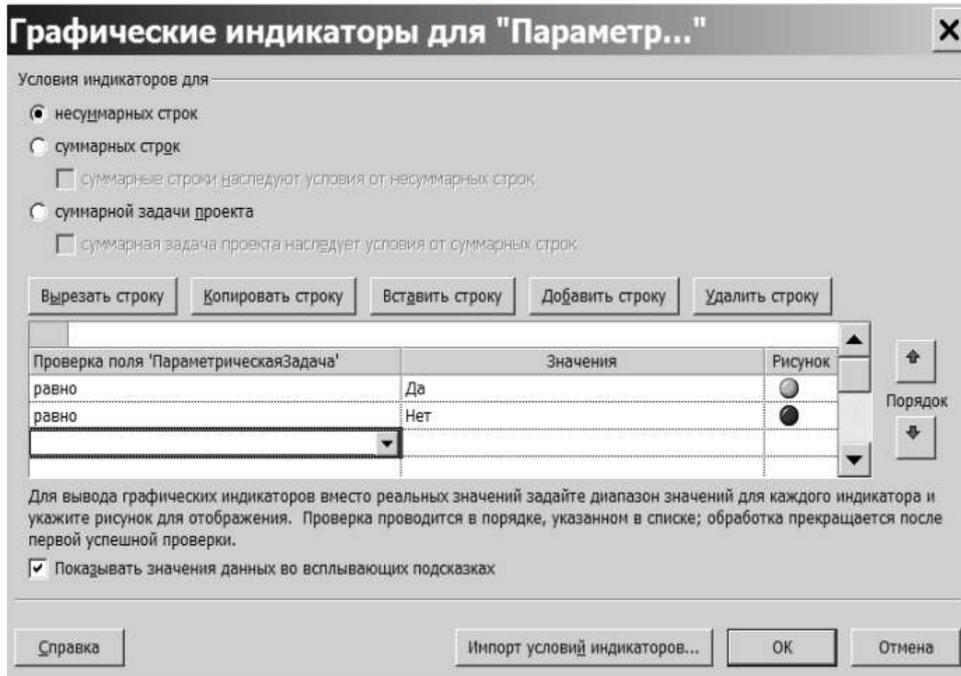


Рис.3 Окно настройки графических индикаторов

	Название	ПараметрическаяЗадача	УсловныйОбъемРаботы	НормативнаяДлительность	ОценкаДлительности	Длительность
0	Лекцияб		0	0 дней	24,5 дней	40 дней
1	Начало реализации проекта	●	0	0 дней	0 дней	0 дней
2	Программирование		0	0 дней	20,5 дней	25 дней
3	Постановка задачи	●	0	0 дней	0 дней	9,88 дней
4	Разработка интерфейса	●	0	0 дней	0 дней	5 дней
5	Разработка модулей обработки дан	●	20	0,4 дней	8 дней	6 дней
6	Разработка структуры базы данных	●	0	0 дней	0 дней	7 дней
7	Заполнение базы данных	●	250	0,05 дней	12,5 дней	8 дней
8	Программирование завершено	●	0	0 дней	0 дней	0 дней
9	Отладка		0	0 дней	4 дней	15 дней
10	Отладка программного комплекса	●	0	0 дней	0 дней	5 дней
11	Тестирование и исправление ошибок	●	0	0 дней	0 дней	10 дней
12	Составление программной докумен	●	20	0,2 дней	4 дней	5 дней
13	Отладка завершена	●	0	0 дней	0 дней	0 дней
14	Конец проекта	●	0	0 дней	0 дней	0 дней

Рис. 4 Пример таблицы параметрический анализ

**Параметрический анализ длительностей задач**

1. Предположим, что длительность программирования модулей обработки данных, заполнения базы данных и составления документации зависит от количества таблиц в базе данных. Тогда нам потребуется два настраиваемых поля: КолТабл (число) и НормаДлит (длительность). В первое мы занесём количество таблиц, а во второе – длительность обработки одной таблицы.

2. Сервис/Настройка/Поля.
3. Тип поля – число, поле – Задача. Выбрать Число10 – кнопка Переименовать – ввести КолТабл – ОК.
4. Тип поля – длительность, поле – Задача. Выбрать Длительность10 – кнопка Переименовать – ввести НормаДлит – ОК.
5. Нам также понадобится поле, в которое будет записан результат оценки длительности. Тип поля – длительность, поля – задач. Выбрать Длительность9 – кнопка Переименовать – ввести ОценкаДлит – ОК.
6. Зададим формулу расчёта оценки: кнопка Формула – ввести выражение [Длительность10]\*[Число10] – ОК.
7. ОК.
8. Вид/Диаграмма Ганта – вставим в таблицу поля КолТабл, НормаДлит и ОценкаДлит (Вставка/Столбец – имя поля – ОК).
9. Установить фильтр по значению поля ПараметрическаяЗадача и отобразить только задачи со значением Да: Проект/Фильтр/Автофильтр – столбец ПараметрическаяЗадача – выбрать Да.
10. Ввести в поле КолТабл количество модулей для параметрических задач в соответствии с рис. 4, а в поле НормаДлит – длительность реализации модуля.
11. Поле ОценкаДлит содержит длительность, вычисленную параметрическим методом. Сравним её с ранее заданным полем Длительность.
12. Скрыть столбцы всех настраиваемых полей: выделить столбец – Правка/Скрыть столбец.
13. Отменить автофильтр: Проект/Фильтр/Автофильтр.

№	Название задачи	Параметрическая	КолТабл	НормаДлит	ОценкаДлит	Длительность
2	Программирование		0	0 дней	0 дней	27 дней
5	Разработка модулей обработки данных	<input checked="" type="radio"/>	10	0,5 дней	5 дней	6 дней
7	Заполнение базы данных	<input checked="" type="radio"/>	30	0,25 дней	7,5 дней	8 дней
9	Отладка		0	0 дней	0 дней	21 дней
12	Составление программной документации	<input checked="" type="radio"/>	40	0,1 дней	4 дней	2 дней

Рис.4 Результат параметрического анализа деятельности задач

#### PERT-анализ длительностей задач

1. Сохраним проект в файл Темаб\_Анализ.mpp.
2. Выведем на экран панель инструментов для PERT-анализа: Вид/Панели инструментов/Анализ по методу PERT.
3. Нажать кнопку Лист ввода PERT – на экране таблица ввода оценок длительности (рис. 5).
4. Введем значения согласно табл. 1.
5. Сохраним проект в файл Тема4\_Анализ.mpp.
6. Сохраним проект в файл Тема4\_PERT.mpp.
7. Нажать кнопку Вычислить по методу PERT на панели инструментов. Результат – изменятся значения длительностей всех задач.

Таблица 1 Оценки длительностей для PERT-анализа

Задача	Оптим. длит.	Ожид. длит.	Пессим. длит.
Постановка задачи	7	10	13
Разработка интерфейса	5	5	5
Разработка модулей обработки данных	6	7	10
Разработка структуры базы данных	4	5	8
Заполнение базы данных	5	7	10
Отладка программного комплекса	5	6	7
Тестирование и исправление ошибок	7	10	14
Составление программной документации	5	5	5

	Название задачи	Длительность	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
0	▣ Лекция6_1	40 дней	0 дней	0 дней	0 дней
1	Начало реализац	0 дней	0 дней	0 дней	0 дней
2	▣ Программирова	25 дней	0 дней	0 дней	0 дней
3	Постановка за	10 дней	7 дней	10 дней	13 дней
4	Разработка ин	5 дней	5 дней	5 дней	5 дней
5	Разработка м	6 дней	4 дней	5 дней	8 дней
6	Разработка ст	7 дней	6 дней	7 дней	10 дней
7	Заполнение б	8 дней	5 дней	7 дней	10 дней
8	Программиро	0 дней	0 дней	0 дней	0 дней
9	▣ Отладка	15 дней	0 дней	0 дней	0 дней
10	Отладка прог	5 дней	5 дней	6 дней	7 дней
11	Тестирование	10 дней	7 дней	10 дней	14 дней
12	Составление и	5 дней	3 дней	5 дней	7 дней
13	Отладка заве	0 дней	0 дней	0 дней	0 дней
14	Конец проекта	0 дней	0 дней	0 дней	0 дней

Рис.5 Лист ввода PERT

8. Проанализировать оптимистический, ожидаемый и пессимистический варианты проекта последовательным нажатием кнопок Диаграмма Ганта – оптимистическая оценка – Диаграмма Ганта – ожидаемая оценка – диаграмма Ганта – пессимистическая оценка.

9. Закрыть проект с сохранением результатов PERT-анализа.

#### Анализ критического пути

1. Открыть проект из файла Тема4\_Анализ.mpp.

2. Открыть диаграмму Ганта. Запустить мастер форматирования диаграммы Ганта (Формат/Мастер диаграмм Ганта).

3. Кнопка Далее – установить переключатель Критический путь –кнопка Готово –кнопка Форматировать. Результат – все критические работы выделены красным цветом.

4. Критическими оказались все задачи проекта. Причина – нарушен крайний срок для задачи Тестирование и исправление ошибок. Он установлен на 9.11.21, в то время как задача заканчивается 12.11.21.

5. Изменим крайний срок. Двойной щелчок мыши по названию задачи – в окне сведений о задаче перейти на вкладку Дополнительно – в поле Крайний срок ввести 16.11.21 – ОК. Результат – задачи Разработка интерфейса и Разработка модулей обработки данных перестали быть критическими, поскольку у них имеется временной резерв.

#### Анализ стоимости проекта

1. Добавим в проект суммарную задачу: Сервис/Параметры – установить флаг Суммарная задача проекта – ОК.

2. Переключимся в диаграмму Ганта, а в ней выберем таблицу затрат: Вид/Таблица/Затраты. Результат – в столбце Общие затраты суммарная задача проекта содержит общую его стоимость, а фазы – стоимость каждой фазы.

#### Анализ стоимости задач разного вида

1. Проанализируем соотношение стоимости задач двух групп со значениями Да и Нет настраиваемого поля ПараметрическаяЗадача.

2. Для этого сгруппируем задачи по значению этого поля: Вид/Диаграмма Ганта – Вид/Таблица/Затраты – Проект/Группировка/Настройка группировки – группировать по ПараметрическаяЗадача – ОК.

3. Результат – общие стоимости задач разных видов указаны в итоговых строках групп (рис. 6).

4. Отменим группировку: Проект/Группировка/Без группировки.

	Название задачи	Фиксированные затраты	Начисление фикс. затрат	Общие затраты	Базовые	Отклонение
	<input type="checkbox"/> ПараметрическаяЗада	0,00р.		210 640,00р.	0,00р.	210 640,00р.
1	Начало реализации пр	0,00р.	Пропорциональное	0,00р.	0,00р.	0,00р.
3	Постановка задачи	0,00р.	Пропорциональное	26 220,00р.	0,00р.	26 220,00р.
4	Разработка интерфей	0,00р.	Пропорциональное	16 800,00р.	0,00р.	16 800,00р.
6	Разработка структуры	0,00р.	Пропорциональное	26 070,00р.	0,00р.	26 070,00р.
8	Программирование за	0,00р.	Пропорциональное	0,00р.	0,00р.	0,00р.
10	Отладка программног	0,00р.	Пропорциональное	46 350,00р.	0,00р.	46 350,00р.
11	Тестирование и испра	0,00р.	Пропорциональное	95 200,00р.	0,00р.	95 200,00р.
13	Отладка завершена	0,00р.	Пропорциональное	0,00р.	0,00р.	0,00р.
14	Конец проекта	0,00р.	Пропорциональное	0,00р.	0,00р.	0,00р.
	<input type="checkbox"/> ПараметрическаяЗада	0,00р.		66 440,00р.	0,00р.	66 440,00р.
5	Разработка модулей с	0,00р.	Пропорциональное	19 560,00р.	0,00р.	19 560,00р.
7	Заполнение базы дан	0,00р.	Пропорциональное	28 080,00р.	0,00р.	28 080,00р.
12	Составление програм	0,00р.	Пропорциональное	18 800,00р.	0,00р.	18 800,00р.

Рис.6 Анализ стоимости по группам задач

### Анализ стоимости ресурсов разного вида

1. Проанализируем распределение стоимости по трудовым, материальным и затратным ресурсам (сколько стоят работа, материалы и прочие затраты). Для этого переключимся в представление использования ресурсов: Вид/Использование ресурсов.

2. Выбрать таблицу Затраты: Вид/Таблица/Затраты.

3. Сгруппируем ресурсы: Проект/Группировка/Тип ресурса.

4. Результат – стоимость разного вида ресурсов указана в итоговых строках группировок.

5. Результат со свёрнутыми подуровнями ресурсов изображён на рис. 7.

6. Удалим группировку: Проект/Группировка/Без группировки.

### Анализ сверхурочных затрат

1. В диаграмме Ганта выберем таблицу затрат: Вид/Диаграмма Ганта – Вид/Таблица/Затраты.

2. В эту таблицу вставим столбец Затраты на сверхурочные: Вставка/Столбец – имя столбца Затраты на сверхурочные – ОК.

3. Результат – в этом столбце отображена стоимость сверхурочных работ по каждой задаче, фазе и всему проекту.

	Название ресурса	Затраты	Базовые затраты	Отклонение
	<input type="checkbox"/> Тип: Трудовой	271 200,00р.	0,00р.	271 200,00р.
	<input type="checkbox"/> Не назначен	0,00р.	0,00р.	0,00р.
1	<input type="checkbox"/> Постановщик	63 700,00р.	0,00р.	63 700,00р.
2	<input type="checkbox"/> Программист1	94 500,00р.	0,00р.	94 500,00р.
3	<input type="checkbox"/> Программист2	113 000,00р.	0,00р.	113 000,00р.
7	<input type="checkbox"/> Инженер по документам	0,00р.	0,00р.	0,00р.
	<input type="checkbox"/> Тип: Материальный	880,00р.	0,00р.	880,00р.
4	<input type="checkbox"/> Бумага	200,00р.	0,00р.	200,00р.
6	<input type="checkbox"/> CD-матрица	680,00р.	0,00р.	680,00р.
	<input type="checkbox"/> Тип: Затраты	5 000,00р.	0,00р.	0,00р.
5	<input type="checkbox"/> Междугородные пере	5 000,00р.	0,00р.	0,00р.

Рис.7 Результат анализа стоимости ресурсов разного вида

### Анализ рисков задач с предварительными длительностями

1. В диаграмме Ганта выберем таблицу ввода: Вид/Диаграмма Ганта – Вид/Таблица/Ввод.

2. Отообразим только задачи с оценкой длительности: Проект/Фильтр/Задачи с оценкой длительности. Результат – нет таких задач, риск отсутствует.

3. Отменить фильтр – Проект/Фильтр/Все задачи.

### Анализ рисков задач со слишком короткой длительностью

1. Будем считать рискованными задачи длительностью два и менее дня.
2. Определим фильтр Короткая задача: Проект/Фильтр/Другие фильтры/Создать.
3. Заполнить поля в соответствии с рис. 8 – ОК.

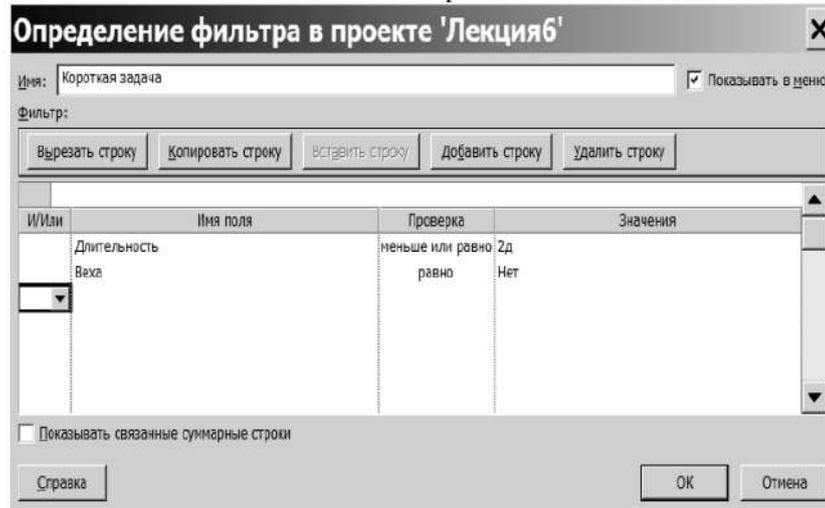


Рис.8 Фильтр для поиска коротких задач

4. Закрыть окно Другие фильтры.
  5. Назначить созданный фильтр: Проект/Фильтр/Короткая задача. Результат – в таблице осталась задача Составление программной документации, которая является рискованной.
  6. Удалить фильтр: Проект/Фильтр/Все задачи.
- Анализ рисков слишком длинных задач с большим количеством ресурсов**
1. Создадим числовое вычисляемое поле:
    - a. Сервис/Настройка/Поля.
    - b. В открывшемся окне настраиваемых полей выберем: тип –число, поле – задача.
    - c. Выбрать число2, кнопка Переименовать.
    - d. В окне переименования ввести КоличествоРесурсов – ОК.
    - e. Нажать кнопку Формула – откроется построитель формул.
    - f. Нажать кнопку Функция и из категории Текст выбрать Len(строка) – эта функция впишется в поле формулы.
    - g. Выделить надпись строка и нажать кнопку Поле.
    - h. Из категории Текст выбрать Названия ресурсов – формула примет вид Len([Названия ресурсов]).
    - i. Нажать ОК и закрыть окно настройки полей.
  2. Перейти в представление Диаграмма Ганта и настроить его соответствующим образом для целей анализа.
    - a. Отобразить таблицу ввода – Вид/Таблица/Ввод.
    - b. Добавить в таблицу столбец КоличествоРесурсов: щелчок правой кнопкой мыши по столбцу Длительность; Имя поля – КоличествоРесурсов – ОК.
    - c. Окно/Разделить – в нижней части представления будет отображаться форма.
    - d. Щелчок правой кнопкой мыши по форме, из контекстного меню выбрать тип формы Ресурсы и последователи.
    - e. Теперь при выборе задачи в диаграмме Ганта форма показывает все её ресурсы и всех её последователей.
  3. Отсортируем таблицу диаграммы Ганта
    - a. Проект/Сортировка/Сортировать по.
    - b. Сортировать по – Длительность (по убыванию), Затем по – Количество ресурсов.
    - c. Снять флажок Сохранить структуру.
    - d. Кнопка Сортировать
  4. Анализируем таблицу и содержимое формы. Вывод: наиболее рискованными

являются задачи Тестирование и исправление ошибок (как наиболее длительная и имеющая три трудовых ресурса) и Составление программной документации (как наиболее ресурсоёмкая).

5. Убрать форму: Окно/Снять разделение.

6. Убрать сортировку: Проект/Сортировка/По идентификатору. 7. Скрыть столбец КоличествоРесурсов.

#### Анализ рисков задач с календарными ограничениями

1. В соответствии с описанной в предыдущем примере технологией создать настраиваемые вычисляемые числовые поля, описанные в табл. 2.

2. Создать фильтр для задач с ограничением Окончание не позднее или с ненулевым значением крайнего срока.

a. Проект/Фильтр/Другие фильтры/Создать.

b. Заполнить окно определения фильтра на основании рис. 10.

c. Нажать ОК.

Таблица 2 Параметры настраиваемых полей

Имя поля	На базе поля	Формула
ЗапасДоОграничения	Число3	[Дата ограничения]-[Окончание]
ЗапасДоКрайнегоСрока	Число4	[Крайний срок]-[Окончание]
Запас	Число5	Иф([ЗапасДоОграничения]<[ЗапасДоКрайнегоСрока]; [ЗапасДоОграничения]; [ЗапасДоКрайнегоСрока])

3. Добавить в таблицу диаграммы Ганта поле Запас.

4. Применить созданный фильтр: Проект/Фильтр/ОграниченияПоВремени.

5. Результат анализа – рискованной является задача Тестирование и исправление ошибок с запасом всего 4 дня.

Рис.10 Параметры фильтра

#### Практическая часть:

Для проекта Внедрение бухгалтерской системы, созданного при выполнении практического занятия №3, выполнить следующие виды анализа.

1. Параметрический анализ длительностей задач. При этом нужно самостоятельно назначить некоторые задачи как параметрические; создать настраиваемые поля для

параметра, нормы и оценки длительности; ввести значения параметра и нормы длительности и вычислить оценку длительности. Результаты анализа сохранить в файле Лаба4\_Параметри-ческийАнализ.mpp.

2. PERT-анализ длительностей задач. При этом самостоятельно задать в таблице ввода PERT значения оптимистической, ожидаемой и пессимистической длительностей. Результаты сохранить в файл Лаба4\_PERTАнализ.mpp.

3. Анализ критического пути. Результат сохранить в файл Лаба4\_АнализКрит.mpp.

4. Анализ стоимости проекта. Результат сохранить в файл Лаба4\_АнализСтоим.mpp.

5. Анализ стоимости параметрических и непараметрических задач. Результат сохранить в файл Лаба4\_ЗадачиРазногоВида.mpp.

6. Анализ стоимости ресурсов разного вида. Результат сохранить в файл Лаба4\_РесурсыРазногоВида.mpp.

7. Анализ сверхурочных затрат. Результат сохранить в файл Лаба4\_Сверхур.mpp.

8. Анализ рисков задач со слишком короткой длительностью. Результат сохранить в файл Лаба4\_Короткие.mpp.

9. Анализ рисков слишком длинных задач с большим количеством ресурсов. Результат сохранить в файл Лаба4\_Длинные.mpp.

10. Анализ рисков задач с календарными ограничениями. Результат сохранить в файл Лаба4\_Ограничения.mpp.

11. Анализ рисков привлечения к работам неопытных сотрудников. Результат сохранить в файл Лаба4\_Неопытные.mpp.

12. Анализ рисков использования трудовых ресурсов с большим объёмом трудозатрат. Результат сохранить в файл Лаба4\_Загруженные.mpp.

13. Анализ рисков использования ресурсов со сверхурочной работой. Результат сохранить в файл Лаба4\_Сверхурочные.mpp.

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## **Практическое занятие 5                      Выравнивание ресурсов**

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков отслеживания хода выполнения проекта, ввода фактических данных и использования методики освоенного объёма.

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

### **Пряжок выполнения заданий**

Примеры отслеживания иллюстрируются с использованием проекта Разработка программы, содержащегося в файле ПЗ6.mpp.

Сохранение базового плана

1. Переключиться в диаграмму Ганта.
2. Выбрать пункт Сервис/Отслеживание/Задать базовый план.
3. Установить переключатели Задать базовый план и всего проекта – ОК.
4. Выбрать Вид/Диаграмма Ганта с отслеживанием. Результат – текущий и базовый планы совпадают. Текущий план изображён цветными отрезками, а базовый – отрезками серого цвета.

### **Настройка представления Использование задач**

1. Переключиться в представление Использование задач.
2. В контекстном меню правой таблицы представления выбрать Трудозатраты, Фактические трудозатраты и Фактические затраты.
3. Вставить в левую таблицу столбцы Остановка, Возобновление, Оставшиеся трудозатраты, % завершения: Вставка/Столбец – выбрать имя столбца –ОК.

### **Ввод повременных данных ресурсов**

1. Введём повременные данные ресурса Постановщик по выполнению задачи Постановка задачи. Ему запланированы работы в течение 10 рабочих дней общей трудоёмкостью 80 ч.

2. В строке Постановщик в ячейки Фактические трудозатраты введём значения по 8 ч в течение первых 10-ти рабочих дней.

3. В строке Бумага в ячейку Фактические трудозатраты введём число 1 в первый день работы постановщика 7.11.09.

4. В строке CD-матрица в ячейку Фактические трудозатраты введём число 2 в этот же день 7.11.21. Результат – процент завершения задачи равен 100% и она отмечается знаком в столбце индикаторов.

5. В строке Междугородные переговоры в ячейку Фактические затраты введём числа 100, 200 и 200.

6. Переключиться в диаграмму Ганта с отслеживанием. Результат – проект выполняется точно в соответствии с первоначальным планом.

### **Ввод повременных данных задач**

1. Переключиться в представление использования задач (Вид/Использование задач).

2. Найти строку задачи Разработка интерфейса. На неё назначен Программист1, Междугородные переговоры и CD-матрица. Трудозатраты программиста составляют 40 ч (5 дней по 8 ч), а CD-матриц расходуется по одной штуке в день (всего 5шт).

3. В строку Фактические трудозатраты правой таблицы, расположенной напротив названия задачи, введём трудозатраты задачи в рабочие дни, начиная с запланированного дня начала работ (23.11.21): 4 ч, 8 ч, 8 ч, 8 ч, 8 ч, 4 ч. Результат – система сама распределяет фактические трудозатраты задачи по выполняющим её трудовым ресурсам. Поскольку такой ресурс единственный (Программист1), все вводимые значения переносятся в его трудозатраты.

4. В строке CD-матрица для этой же задачи в первый день работы программиста введём фактические трудозатраты 5.

5. В строке Междугородные переговоры для этой же задачи в строке Фактические затраты введём числа 200 в каждый день работы Программист1.

6. Найти строку задачи Разработка структуры базы данных. На неё назначен Программист2, Междугородные переговоры и CD-матрица. Трудозатраты программиста

составляют 56 ч (7 дней по 8 ч), а CD-матриц расходуется по 1 штуке в день.

7. В строку Фактические трудозатраты правой таблицы, расположенной напротив названия задачи, введём трудозатраты задачи в рабочие дни, начиная с запланированного дня начала работ (23.11.09): 8 ч, 8 ч, 8 ч, 8 ч, 8 ч, 8 ч, 8 ч. Результат – система аналогично распределяет фактические трудозатраты задачи по выполняющим её трудовым ресурсам. Поскольку такой ресурс единственный (Программист2), все вводимые значения переносятся в его трудозатраты.

8. В строке CD-матрица для этой же задачи в первый день работы программиста введём фактические трудозатраты 7.

9. В строке Междугородные переговоры для этой же задачи в строке Фактические затраты введём числа 150 в каждый день работы Программист2. Результат изображён на рис. 2.

Рис. 2 Результат ввода повременных данных задач

10. Введём таким же способом фактические трудозатраты задач Разработка модулей обработки данных (4, 4, 4, 4, 8, 8, 8, 8 с 1.10.21) и Заполнение базы данных (8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4 с 2.10.21).

11. Введём фактические трудозатраты CD-матриц по 7 и 8 штук соответственно 1.10.21 и 2.10.21.

12. Переключиться в диаграмму Ганта с отслеживанием. Результат – в связи с тем, что фактическое выполнение задач отстало от ранее запланированного графика, выполнение проекта в целом отстаёт от базового плана (время окончания последних цветных отрезков находится позже времени окончания последних отрезков серого цвета, что изображено на рис. 3).

### Ввод фактических трудозатрат

1. Выбрать представление Использование задач.
2. Добавить в таблицу представления столбец Фактические трудозатраты.
3. Найти строку задачи Отладка программного комплекса, а в ней – ячейки столбцов Фактические трудозатраты и Оставшиеся трудозатраты. В них соответственно записаны 0 ч и 120 ч. Эти трудозатраты делятся на двух программистов и постановщика, у которых значения фактических и оставшихся трудозатрат равны соответственно 0 ч и 40 ч.

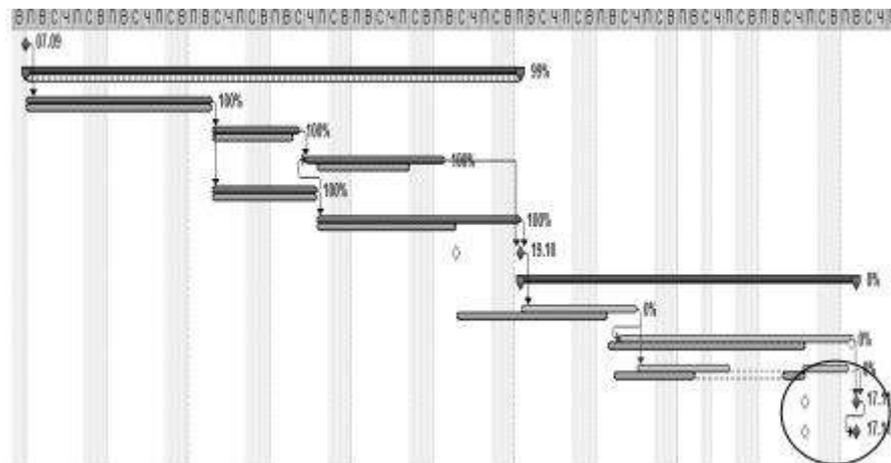


Рис. 3 Отставание выполнения проекта от плана

4. Ввести в поле Фактические трудозатраты задачи Отладка программного комплекса значение 32 ч. Результат – введённые трудозатраты пропорционально распределены между программистами (поскольку степень их участия в работе одинакова, произошло равномерное распределение). Постановщику не досталось фактических трудозатрат, поскольку в первые дни он не участвует в этой задаче. Фактические трудозатраты распределены, начиная с запланированного дня начала работ (с 19.10.21). Оставшиеся трудозатраты равны 88 ч. В графе Остановка для задачи Комплексная отладка модулей указана дата прекращения работ, а в графе Возобновление – дата, с которой работы будут продолжены. Обе даты равны 19.10.21.

5. Предположим, что работы были возобновлены 23.10.21 и велись ежедневно в полном объёме вплоть до завершения задачи. Установить в графе Возобновление дату 23.10.21. Ввести в поле Фактические трудозатраты задачи Отладка программного комплекса значение 120ч. Результат – оставшиеся трудозатраты равны 0, работа выполнена полностью.

#### Ввод процента завершения

1. Перейти в представление Использование задач.  
 2. Найти строку задачи Тестирование и исправление ошибок.  
 3. В левой таблице представления в этой строке найти поле %завершения и занести в него число 30. Результат – система перенесла 30 % запланированных трудозатрат в фактические трудозатраты.

4. Открыть окно свойств задачи Тестирование и исправление ошибок: двойной щелчок мыши по её названию.

5. В открывшемся окне перейти на вкладку Общие, найти поле Процент завершения и занести в него число 50. Результат – 50% запланированных трудозатрат ресурсов перенесены в фактические трудозатраты.

6. Перейти в представление Диаграмма Ганта и найти на диаграмме отрезок задачи Тестирование и исправление ошибок. Чёрная линия на этом отрезке соответствует фактическому ходу выполнения работ по задаче. Поскольку линия не достигла конца отрезка, задача является незавершённой.

7. Переместить при помощи мыши правую границу линии фактического выполнения задачи примерно на 80 % отрезка. Результат – система определит точный процент завершения и в соответствии с ним перенесёт запланированные трудозатраты в фактические.

#### Анализ хода выполнения проекта

1. Перейти в представление Использование задач.  
 2. В правой части представления убрать строки Фактические трудозатраты и Фактические затраты.  
 3. Вывести таблицу для анализа освоенного объёма: Вид/Таблица/Другие таблицы – выбрать из списка Освоенный объём – Применить.

4. Вставить в таблицу суммарную задачу проекта: Сервис/Параметры – вкладка

Вид – установить флаг Суммарная задача проекта.

5. В строке суммарной задачи выведены основные показатели:

- a. БСЗР – затраты на текущий момент по базовому плану;
- b. ФСВР – сколько фактически затрачено на проект к текущему моменту;
- c. БСВР – сколько планировалось затратить на те работы, которые уже выполнены.

6. Установим дату отчёта, на которую будем выполнять анализ. В качестве таковой используем 19.11.21, когда проект должен быть завершён: Проект/Сведения о проекте, Дата отчёта – 19.11.21.

7. Анализируем индексы суммарной задачи проекта:

- a. ОКП < 0 – проект выполняется с отставанием от плана (значение этого индекса – стоимость тех работ, которые мы выполнили сверх плана к текущему моменту);
- b. ПОПЗ = БПЗ – затраты по проекту ожидаются в пределах плановых показателей (значение ПОПЗ – ожидаемая стоимость проекта после его завершения);
- c. ОПЗ = 0 – величина, на которую увеличатся затраты (значение ОПЗ – размер ожидаемой экономии).

8. Выберем таблицу для анализа скорости выполнения работ: Вид/Таблица/Другие таблицы – выбрать из списка Показатели календ. плана (освоенный объём) – Применить.

9. Проанализируем индексы суммарной задачи проекта, которых не было в предыдущей таблице:

- a. ООКП < 0 – проект отстаёт от плана (значение ООКП показывает процент опережения плана);
- b. ИОКП < 1 – проект отстаёт от плана.

10. Выберем таблицу для анализа затрат проекта:

Вид/Таблица/Другие таблицы – выбрать строку Показатели затрат (освоенный объём) – Применить.

11. Проанализируем показатели суммарной задачи проекта, которых не было в двух предыдущих таблицах:

- a. ООПС = 0 – проект не экономит средства (значение ООПС – процент сэкономленных средств);
- b. ИОС = 1 – проект не экономит средства;
- c. ПЗВ = 1 – средства не экономятся, но и нет перерасхода.

### Практическая часть:

1. Для ранее созданного проекта Внедрение бухгалтерской системы самостоятельно ввести фактические данные о выполнении работ различными способами для задач, принадлежащих различным фазам в соответствии с табл. 1.

Таблица 1 Виды фактических данных

Фаза	Способ ввода фактических данных	Файл сохранения результата
Выбор системы	Процент завершения	<i>Тема8_Процент.mpp</i>
Приобретение программного обеспечения	Ввод фактических или оставшихся трудозатрат с перерывом в работе	<i>Тема8_Фактич.mpp</i>
Составление проекта сети	Ввод повременных данных задач	<i>Тема8_Задачи.mpp</i>
Приобретение компьютеров и сетевого оборудования	Ввод повременных данных ресурсов	<i>Тема8_Ресурсы.mpp</i>

2. Выполнить анализ хода выполнения проекта по методике освоенного объёма на предполагаемую дату окончания работ и сделать вывод о скорости выполнения работ (раньше или позже плана) и о предполагаемом объёме фактических затрат (больше или меньше плановых).

### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 6                      Отслеживание проекта

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков отслеживания хода выполнения проекта, ввода фактических данных и использования методики освоенного объёма.

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

### **Прядок выполнения заданий**

Примеры отслеживания иллюстрируются с использованием проекта Разработка программы, содержащегося в файле ПЗ6.mpp.

Сохранение базового плана

1. Переключиться в диаграмму Ганта.
2. Выбрать пункт Сервис/Отслеживание/Задать базовый план.
3. Установить переключатели Задать базовый план и всего проекта – ОК.
4. Выбрать Вид/Диаграмма Ганта с отслеживанием. Результат – текущий и базовый планы совпадают. Текущий план изображён цветными отрезками, а базовый – отрезками серого цвета.

**Настройка представления Использование задач**

1. Переключиться в представление Использование задач.
2. В контекстном меню правой таблицы представления выбрать Трудозатраты, Фактические трудозатраты и Фактические затраты.
3. Вставить в левую таблицу столбцы Остановка, Возобновление, Оставшиеся трудозатраты, % завершения: Вставка/Столбец – выбрать имя столбца – ОК.

**Ввод повременных данных ресурсов**

1. Введём повременные данные ресурса Постановщик по выполнению задачи Постановка задачи. Ему запланированы работы в течение 10 рабочих дней общей трудоёмкостью 80 ч.
2. В строке Постановщик в ячейки Фактические трудозатраты введём значения по 8 ч в течение первых 10-ти рабочих дней.
3. В строке Бумага в ячейку Фактические трудозатраты введём число 1 в первый

день работы постановщика 7.11.09.

4. В строке CD-матрица в ячейку Фактические трудозатраты введём число 2 в этот же день 7.11.21. Результат – процент завершения задачи равен 100% и она отмечается знаком в столбце индикаторов.

5. В строке Междугородные переговоры в ячейку Фактические затраты введём числа 100, 200 и 200.

6. Переключиться в диаграмму Ганта с отслеживанием. Результат – проект выполняется точно в соответствии с первоначальным планом.

#### Ввод повременных данных задач

1. Переключиться в представление использования задач (Вид/Использование задач).

2. Найти строку задачи Разработка интерфейса. На неё назначен Программист1, Междугородные переговоры и CD-матрица. Трудозатраты программиста составляют 40 ч (5 дней по 8 ч), а CD-матриц расходуется по одной штуке в день (всего 5шт).

3. В строку Фактические трудозатраты правой таблицы, расположенной напротив названия задачи, введём трудозатраты задачи в рабочие дни, начиная с запланированного дня начала работ (23.11.21): 4 ч, 8 ч, 8 ч, 8 ч, 8 ч, 4 ч. Результат – система сама распределяет фактические трудозатраты задачи по выполняющим её трудовым ресурсам. Поскольку такой ресурс единственный (Программист1), все вводимые значения переносятся в его трудозатраты.

4. В строке CD-матрица для этой же задачи в первый день работы программиста введём фактические трудозатраты 5.

5. В строке Междугородные переговоры для этой же задачи в строке Фактические затраты введём числа 200 в каждый день работы Программист1.

6. Найти строку задачи Разработка структуры базы данных. На неё назначен Программист2, Междугородные переговоры и CD-матрица. Трудозатраты программиста составляют 56 ч (7 дней по 8 ч), а CD-матриц расходуется по 1 штуке в день.

7. В строку Фактические трудозатраты правой таблицы, расположенной напротив названия задачи, введём трудозатраты задачи в рабочие дни, начиная с запланированного дня начала работ (23.11.09): 8 ч, 8 ч, 8 ч, 8 ч, 8 ч, 8 ч, 8 ч. Результат – система аналогично распределяет фактические трудозатраты задачи по выполняющим её трудовым ресурсам. Поскольку такой ресурс единственный (Программист2), все вводимые значения переносятся в его трудозатраты.

8. В строке CD-матрица для этой же задачи в первый день работы программиста введём фактические трудозатраты 7.

9. В строке Междугородные переговоры для этой же задачи в строке Фактические затраты введём числа 150 в каждый день работы Программист2. Результат изображён на рис. 2.

	23.11.09	24.11.09	25.11.09	26.11.09	27.11.09	28.11.09	29.11.09	30.11.09
Трудозатр.	4ч	8ч	8ч	8ч	8ч	8ч	4ч	
Факт труд.	4ч	8ч	8ч	8ч	8ч	8ч	4ч	
Факт затраты	1 875,00р	3 450,00р	3 450,00р	0,00р	0,00р	3 450,00р	3 450,00р	1 625,00р
Трудозатр.	4ч	8ч	8ч			8ч	8ч	4ч
Факт труд.	4ч	8ч	8ч			8ч	8ч	4ч
Факт затраты	1 625,00р	3 250,00р	3 250,00р	0,00р	0,00р	3 250,00р	3 250,00р	1 625,00р
Трудозатр.								
Факт труд.								
Факт затраты	200,00р	200,00р	200,00р	0,00р	0,00р	200,00р	200,00р	200,00р
Трудозатр. (в)	5							
Факт труд. (в)	5							
Факт затраты	50,00р							
Трудозатр.								
Факт труд.								
Факт затраты								
Трудозатр.								
Факт труд.								
Факт затраты								
Трудозатр. (в)								
Факт труд. (в)								
Факт затраты								
Трудозатр.	8ч	8ч	8ч			8ч	8ч	8ч
Факт труд.	8ч	8ч	8ч			8ч	8ч	8ч
Факт затраты	3 720,00р	3 650,00р	3 650,00р	0,00р	0,00р	3 650,00р	3 650,00р	3 650,00р
Трудозатр.	8ч	8ч	8ч			8ч	8ч	8ч
Факт труд.	8ч	8ч	8ч			8ч	8ч	8ч
Факт затраты	3 500,00р	3 500,00р	3 500,00р			3 500,00р	3 500,00р	3 500,00р
Трудозатр.								

Рис. 2 Результат ввода повременных данных задач

10. Введём таким же способом фактические трудозатраты задач Разработка модулей обработки данных (4, 4, 4, 4, 8, 8, 8, 8 с 1.10.21) и Заполнение базы данных (8, 8, 8, 8, 4, 4, 4, 4, 4, 4, 4, 4 с 2.10.21).

11. Введём фактические трудозатраты CD-матриц по 7 и 8 штук соответственно 1.10.21 и 2.10.21.

12. Переключиться в диаграмму Ганта с отслеживанием. Результат – в связи с тем, что фактическое выполнение задач отстало от ранее запланированного графика, выполнение проекта в целом отстаёт от базового плана (время окончания последних цветных отрезков находится позже времени окончания последних отрезков серого цвета, что изображено на рис. 3).

### Ввод фактических трудозатрат

1. Выбрать представление Использование задач.
2. Добавить в таблицу представления столбец Фактические трудозатраты.
3. Найти строку задачи Отладка программного комплекса, а в ней – ячейки столбцов Фактические трудозатраты и Оставшиеся трудозатраты. В них соответственно записаны 0 ч и 120 ч. Эти трудозатраты делятся на двух программистов и постановщика, у которых значения фактических и оставшихся трудозатрат равны соответственно 0 ч и 40 ч.

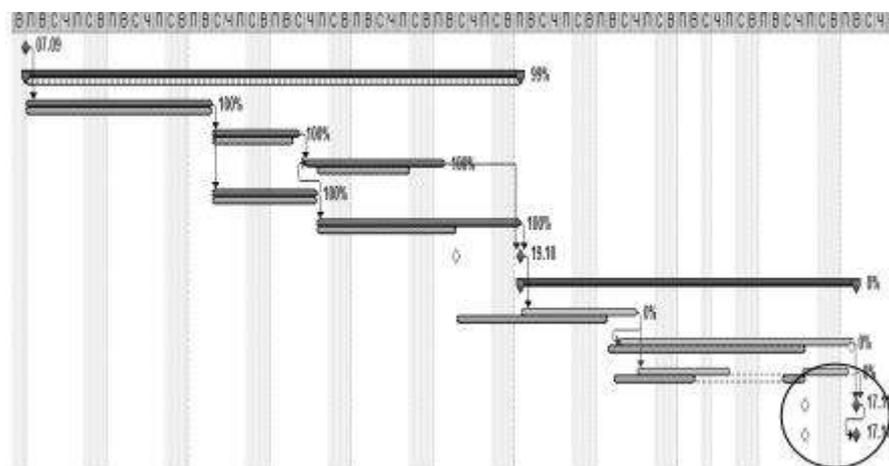


Рис. 3 Отставание выполнения проекта от плана

4. Ввести в поле Фактические трудозатраты задачи Отладка программного комплекса значение 32 ч. Результат – введённые трудозатраты пропорционально распределены между программистами (поскольку степень их участия в работе одинакова, произошло равномерное распределение). Постановщику не досталось фактических трудозатрат, поскольку в первые дни он не участвует в этой задаче. Фактические трудозатраты распределены, начиная с запланированного дня начала работ (с 19.10.21). Оставшиеся трудозатраты равны 88 ч. В графе Остановка для задачи Комплексная отладка модулей указана дата прекращения работ, а в графе Возобновление – дата, с которой работы будут продолжены. Обе даты равны 19.10.21.

5. Предположим, что работы были возобновлены 23.10.21 и велись ежедневно в полном объёме вплоть до завершения задачи. Установить в графе Возобновление дату 23.10.21. Ввести в поле Фактические трудозатраты задачи Отладка программного комплекса значение 120ч. Результат – оставшиеся трудозатраты равны 0, работа выполнена полностью.

### Ввод процента завершения

1. Перейти в представление Использование задач.
2. Найти строку задачи Тестирование и исправление ошибок.
3. В левой таблице представления в этой строке найти поле %завершения и занести в него число 30. Результат – система перенесла 30 % запланированных трудозатрат в фактические трудозатраты.
4. Открыть окно свойств задачи Тестирование и исправление ошибок: двойной щелчок мыши по её названию.

5. В открывшемся окне перейти на вкладку Общие, найти поле Процент завершения и занести в него число 50. Результат – 50% запланированных трудозатрат ресурсов перенесены в фактические трудозатраты.

6. Перейти в представление Диаграмма Ганта и найти на диаграмме отрезок задачи Тестирование и исправление ошибок. Чёрная линия на этом отрезке соответствует фактическому ходу выполнения работ по задаче. Поскольку линия не достигла конца отрезка, задача является незавершённой.

7. Переместить при помощи мыши правую границу линии фактического выполнения задачи примерно на 80 % отрезка. Результат – система определит точный процент завершения и в соответствии с ним перенесёт запланированные трудозатраты в фактические.

#### **Анализ хода выполнения проекта**

1. Перейти в представление Использование задач.

2. В правой части представления убрать строки Фактические трудозатраты и Фактические затраты.

3. Вывести таблицу для анализа освоенного объёма: Вид/Таблица/Другие таблицы – выбрать из списка Освоенный объём – Применить.

4. Вставить в таблицу суммарную задачу проекта: Сервис/Параметры – вкладка Вид – установить флаг Суммарная задача проекта.

5. В строке суммарной задачи выведены основные показатели:

a. БСЗР – затраты на текущий момент по базовому плану;

b. ФСВР – сколько фактически затрачено на проект к текущему моменту;

c. БСВР – сколько планировалось затратить на те работы, которые уже выполнены.

6. Установим дату отчёта, на которую будем выполнять анализ. В качестве таковой используем 19.11.21, когда проект должен быть завершён: Проект/Сведения о проекте, Дата отчёта – 19.11.21.

7. Анализируем индексы суммарной задачи проекта:

a. ОКП < 0 – проект выполняется с отставанием от плана (значение этого индекса – стоимость тех работ, которые мы выполнили сверх плана к текущему моменту);

b. ПОПЗ = БПЗ – затраты по проекту ожидаются в пределах плановых показателей (значение ПОПЗ – ожидаемая стоимость проекта после его завершения);

c. ОПЗ = 0 – величина, на которую увеличатся затраты (значение ОПЗ – размер ожидаемой экономии).

8. Выберем таблицу для анализа скорости выполнения работ: Вид/Таблица/Другие таблицы – выбрать из списка Показатели календ. плана (освоенный объём) – Применить.

9. Проанализируем индексы суммарной задачи проекта, которых не было в предыдущей таблице:

a. ООКП < 0 – проект отстаёт от плана (значение ООКП показывает процент опережения плана);

b. ИОКП < 1 – проект отстаёт от плана.

10. Выберем таблицу для анализа затрат проекта:

Вид/Таблица/Другие таблицы – выбрать строку Показатели затрат (освоенный объём) – Применить.

11. Проанализируем показатели суммарной задачи проекта, которых не было в двух предыдущих таблицах:

a. ООПС = 0 – проект не экономит средства (значение ООПС – процент сэкономленных средств);

b. ИОС = 1 – проект не экономит средства;

c. ПЗВ = 1 – средства не экономятся, но и нет перерасхода.

#### **Практическая часть:**

1. Для ранее созданного проекта Внедрение бухгалтерской системы самостоятельно ввести фактические данные о выполнении работ различными способами для задач, принадлежащих различным фазам в соответствии с табл.1.

Таблица 1 Виды фактических данных

Фаза	Способ ввода фактических данных	Файл сохранения результата
Выбор системы	Процент завершения	<i>Тема8_Процент.mpp</i>
Приобретение программного обеспечения	Ввод фактических или оставшихся трудозатрат с перерывом в работе	<i>Тема8_Фактич.mpp</i>
Составление проекта сети	Ввод повременных данных задач	<i>Тема8_Задачи.mpp</i>
Приобретение компьютеров и сетевого оборудования	Ввод повременных данных ресурсов	<i>Тема8_Ресурсы.mpp</i>

2. Выполнить анализ хода выполнения проекта по методике освоенного объёма на предполагаемую дату окончания работ и сделать вывод о скорости выполнения работ (раньше или позже плана) и о предполагаемом объёме фактических затрат (больше или меньше плановых).

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

**Практическое занятие 7                      Отчетность по проекту**

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков формирования отчетов по проекту..

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Project

**Прядок выполнения заданий**

Примеры иллюстрируются с использованием проекта Разработка программы, содержащегося в файле ПЗ7.mpp.

**Формирование статистики проекта**

1. Проект/Сведения о проекте – откроется окно сведений о проекте.

2. Кнопка Статистика – на экране окно статистики проекта.
3. Анализ содержимого окна показывает, что завершение проекта ожидается через 3,28 дня, фактические затраты составили 258030р. и стоимость оставшихся работ равна 19425 р.

#### **Формирование стандартного отчёта**

1. Отчёт/Отчёты – открылось окно с категориями стандартных отчётов.
2. Двойным щелчком мыши выбрать категорию Назначения – откроется окно со списком отчётов этой категории.
3. Двойным щелчком мыши выбрать отчёт Дела по исполнителям и времени – первая страница отчёта выводится на экран. Отчёт показывает ежедневное распределение трудозатрат исполнителей по задачам.
4. При помощи кнопок листаем страницы отчёта по горизонтали.
5. Выход из отчёта – кнопка Закрыть.

#### **Настройка стандартного отчёта**

1. Отчёт/Отчёты – открылось окно с категориями стандартных отчётов.
2. Двойным щелчком мыши выбрать категорию Назначения – откроется окно со списком отчётов этой категории.
3. Щелчком мыши выбрать отчёт Дела по исполнителям и времени и нажать кнопку Изменить.
4. В открывшемся окне свойств перекрёстного отчёта изменить в поле Столбец значение Дни на Недели. ОК. На экране снова список отчётов категории Назначения.
5. Двойным щелчком мыши открыть отчёт Дела по исполнителям и времени.
6. Результат – в отличие от предыдущего варианта отчёта трудозатраты исполнителей сгруппированы по неделям.
7. Закрыть отчёт и окно категорий отчётов.

#### **Создание настраиваемого отчёта**

1. Открыть список категорий отчётов: Отчёт/Отчёты.
2. Двойным щелчком мыши открыть категорию Настраиваемые.
3. Кнопка Создать – откроется список типов отчётов.
4. Выбрать Перекрёстная таблица. ОК.
5. Заполнить окно свойств отчёта в соответствии с изображением на рис. 2. ОК.
6. Нажать кнопку Просмотр – формируется отчёт.

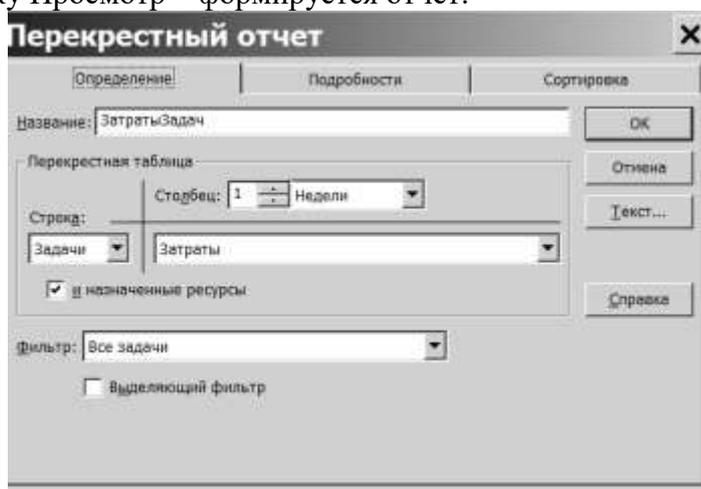


Рис.2 Параметры создаваемого отчета

#### **Удаление отчёта**

1. Открыть список категорий отчётов: Отчёт/Отчёты.
2. Двойным щелчком мыши открыть категорию Настраиваемые.
3. Нажать кнопку Организатор.
4. В правом списке выделить название удаляемого отчёта ЗатратыЗадач и нажать кнопку Удалить, подтвердить удаление.
5. Закрыть все ранее открытые окна (организатор, список отчётов, перечень

категорий отчётов).

### Формирование наглядного отчёта

1. Отчёт/Наглядные отчёты.
2. В открывшемся окне на вкладке Все выбрать отчёт Сводный отчёт о затратах ресурсов и нажать кнопку Просмотреть.
3. Отчёт формируется в системе Excel и имеет вид круговой диаграммы.
4. Закрыть Excel.

### Создание наглядного отчёта

1. Отчёт/Наглядные отчёты.
2. Нажать кнопку Создать шаблон.
3. В поле Выберите данные, по которым требуется создать отчёт установить Сводка по задачам. ОК.
4. В системе Excel откроется макет отчёта, изображённый на рис.4.

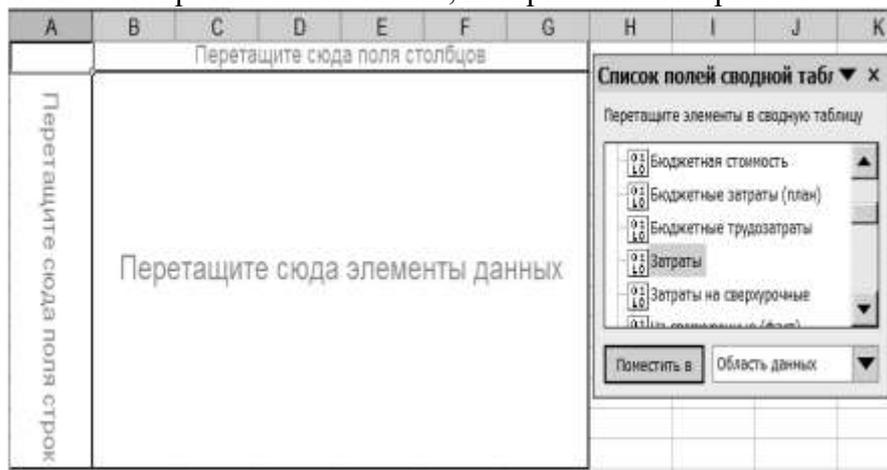


Рис.4 Макет наглядного отчета

5. Перетащить из списка полей Задачи в область полей строк, а Затраты – в область элементов данных.
6. При помощи кнопки выбора поля Задачи отобразим в отчёте задачи второго уровня. Результат изображён на рис. 5.
7. Закрыть Excel.

A	B	C	D
Затраты			
Задачи	Задачи	Задачи 02	Итого
Тема9_рабочий	Програм	Программирование	0
		Постановка задачи	26345
		Разработка интерфейса	17500
		Разработка модулей обработки данных	19560
		Разработка структуры базы данных	25620
		Заполнение базы данных	28080
		Программирование завершено	0
		Программирование Итого	117105
	Отладки	Отладка	0
		Отладка программного комплекса	46350
		Тестирование и исправление ошибок	95200
		Составление программной документации	18800
		Отладка завершена	0
		Отладка Итого	160350
	Тема9_рабочий	Итого	277455
	Общий итог		277455

Рис.5 Построенный наглядный отчет

### Практическая часть:

Для проекта Внедрение бухгалтерской системы, созданного ранее при выполнении

практического занятия 6, сформировать следующие виды отчётов.

1. Статистика проекта.
2. Любой из стандартных отчётов.
3. Любой из стандартных отчётов с предварительно изменёнными параметрами.
4. Свои собственные настраиваемые отчёты всех четырёх видов.
5. Предопределённый наглядный отчёт. Результат сохранить в файл

ПЗ7\_Отчёт1.xls.

6. Самостоятельно созданный наглядный отчёт. Результат сохранить в файл ПЗ7\_Отчёт2.xls.

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 8 Изучение возможностей MS Visio

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков работы с MS Visio

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

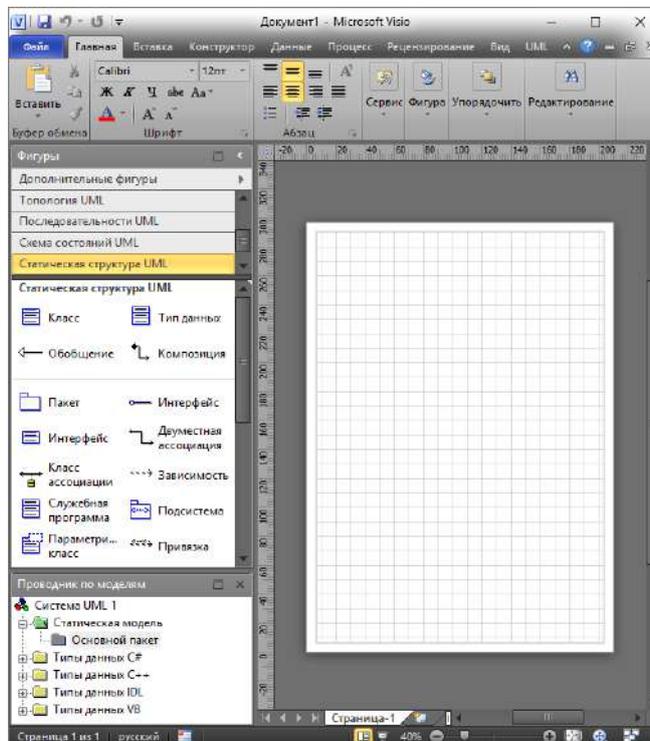
Программное обеспечение: Microsoft Visio

**Прядок выполнения заданий**

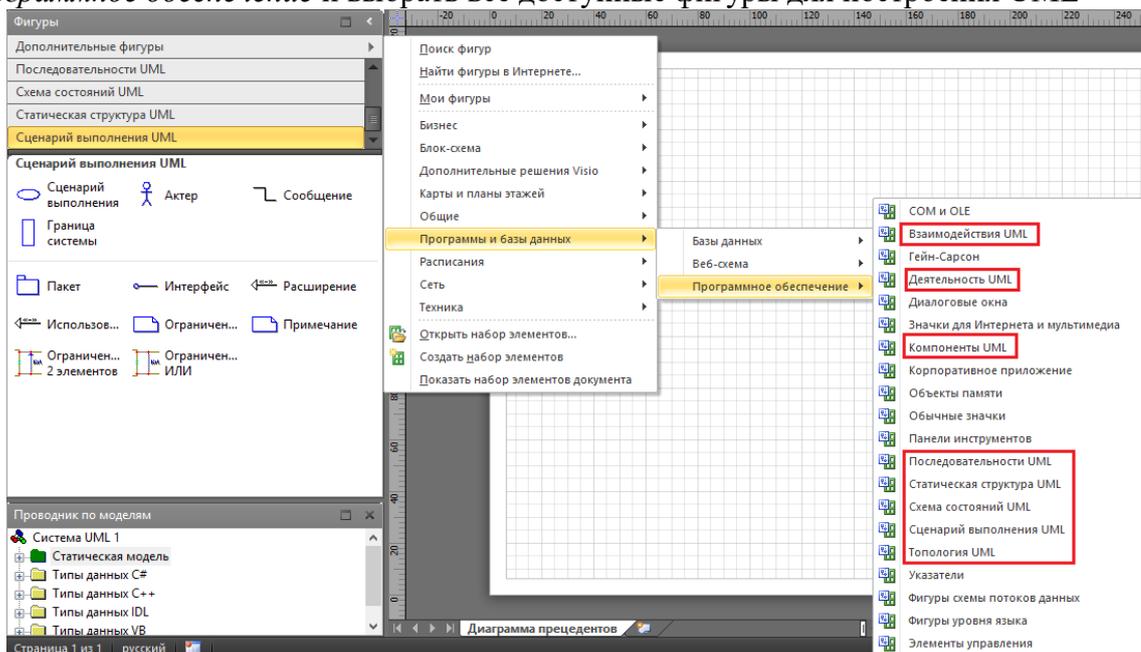
В качестве примера рассматривается моделирование системы продажи товаров по каталогу.

1. Запустите MS Visio.
2. На экране выбора шаблона выберите категорию Программы и БД и в ней элемент Схема модели UML.

Нажмите кнопку Создать в правой части экрана.



3. Далее необходимо открыть все фигуры, необходимые для построения UML-диаграмм. Для этого в левой части экрана необходимо нажать кнопку *Дополнительные фигуры*. В открывшемся вспомогательном меню выбрать *Программы и БД* -> *Программное обеспечение* и выбрать все доступные фигуры для построения UML

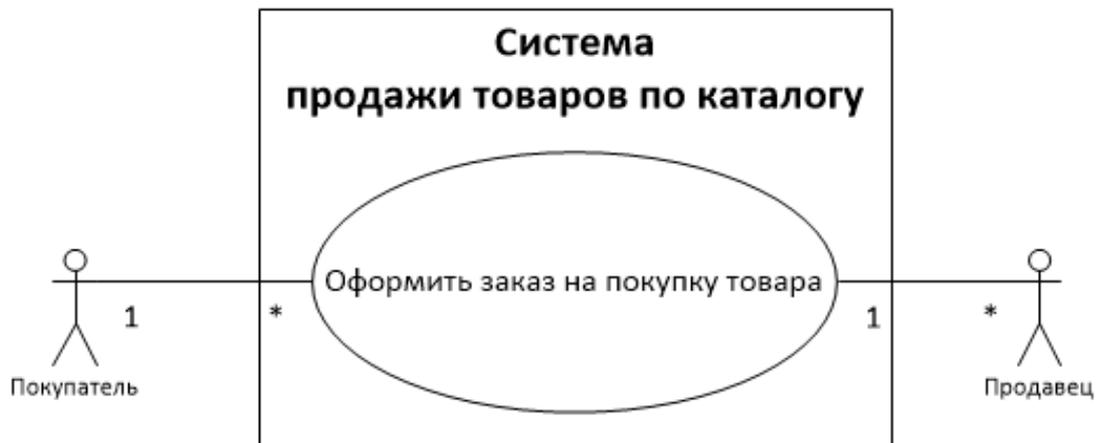
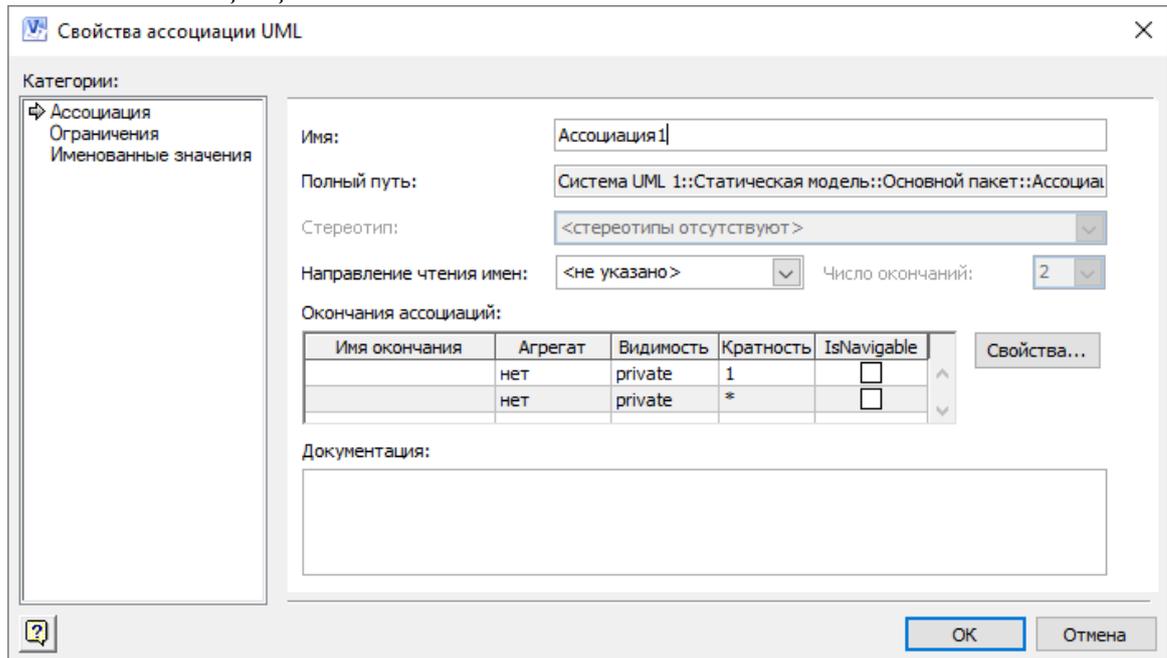


#### 4. Выбор актеров.

В качестве актеров данной системы могут выступать два субъекта, один из которых является продавцом, а другой – покупателем. Каждый из этих актеров взаимодействует с рассматриваемой системой продажи товаров по каталогу и является ее пользователем, т. е. они оба обращаются к соответствующему сервису «Оформить заказ на покупку товара». Как следует из существа выдвигаемых к системе требований, этот сервис выступает в качестве варианта использования разрабатываемой диаграммы, первоначальная структура которой может включать в себя только двух указанных актеров и единственный вариант использования.

– В группе фигур *Сценарий выполнения UML* выбрать блок *Граница системы* и добавить его на лист.

- Внутри границы системы добавить блок *Сценарий выполнения* и добавить к нему название, дважды щелкнув внутри блока.
- Добавить два блока *Актер* – покупатель и продавец.
- С помощью блока *Сообщение* установите связь актеров и варианта использования. Двойным щелчком правой кнопки мыши по блоку Сообщение откройте окно *Свойств ассоциации UML*.



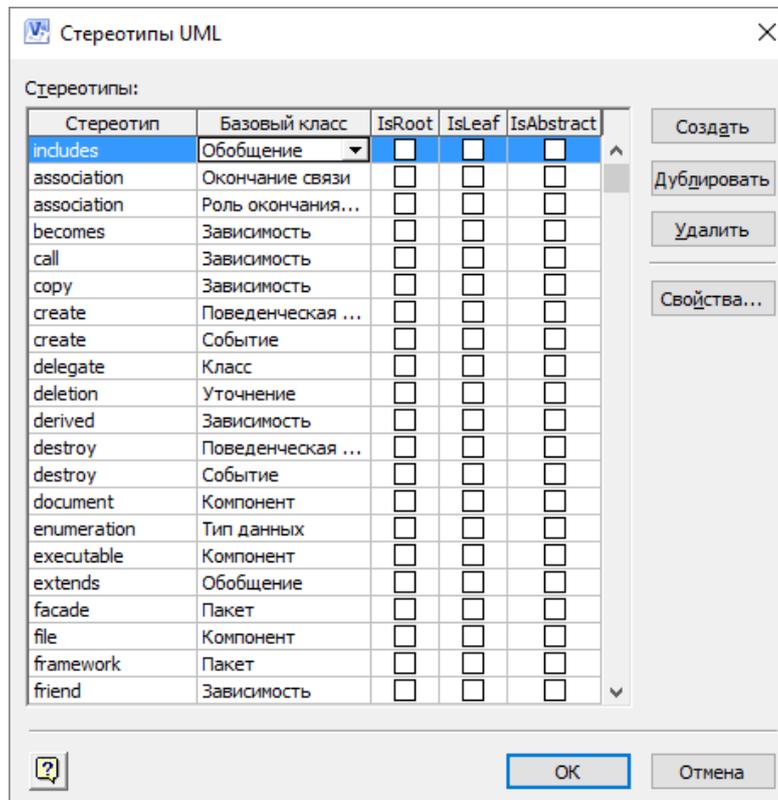
### 5. Выделение дополнительных вариантов использования.

Детализировать вариант использования «Оформить заказ на продажу товара» можно выделив следующие

дополнительные варианты использования:

- обеспечить покупателя информацией – является отношением включения;
- согласовать условия оплаты – является отношением включения;
- заказать товар со склада – является отношением включения;
- запросить каталог товаров – является отношением расширения.

Так как в MS Visio отсутствует отношение включения, его необходимо добавить самостоятельно. Для этого перейти на вкладку *UML* -> в группе *Модель* выбрать пункт *Стереотипы*. В открывшемся окне нажать кнопку *Создать* и настроить стереотип



Далее на новом листе необходимо добавить границу системы и все варианты использования. После чего соединить варианты использования с помощью блока *Расширение*.

Для того, чтобы изменить тип отношения дважды щелкните по стрелке и окне свойств задайте необходимые параметры.

Дополненная диаграмма вариантов использования примет вид:



## 6. Написание описательной спецификации для каждого варианта использования.

Спецификация для варианта использования «Оформить заказ на покупку компьютера» приведена в таблице 1.

Таблица 1 – Спецификация варианта использования

Раздел	Описание
Краткое описание	Покупатель желает оформить заказ на покупку компьютера, который он выбрал в каталоге товаров. При условии, что клиент зарегистрирован и выбранный компьютер есть в наличии оформляется заказ. Если клиент не зарегистрирован, то предлагается ему пройти регистрацию, и после этого заказать выбранный компьютер. Если компьютера нет в наличии, то предлагается заказать товар со склада в течении заданного срока поставки.
Субъекты	Продавец, Покупатель
Предусловия	В каталоге товаров имеются компьютеры, которые можно заказать. У покупателей есть доступ к системе для регистрации. Продавцы умеют пользоваться рассматриваемой системой продажи. У покупателя есть бонусы.
Основной поток	<p>Зарегистрированный покупатель имеет возможность заказать любой компьютер из каталога товаров. В случае наличия выбранного компьютера оформляется заказ с присвоением ему уникального номера. После этого покупателю предлагается выбрать способ оплаты и способ получения компьютера.</p> <p>В случае отсутствия компьютера в наличии предлагается оформить заказ со склада и ожидания его поставки в рамках указанного срока или выбрать другой компьютер.</p>
Альтернативный поток	<p>Покупатель не зарегистрирован. В этом случае, прежде чем оформить заказ на компьютер, ему предлагается пройти регистрацию.</p> <p>Попытка заказать товар, который отсутствует на складе</p> <p>Начисление бонусов</p>
Постусловия	Заказ оформлен и определен срок поставки компьютера и место его получения

### Практическая часть:

Построить диаграмму прецедентов (вариантов использования) в соответствии с вариантом. Составить спецификацию.

Отчет по практическому занятию выполняется в формате MS Word, который содержит пошаговое описание процесса построения организационной диаграммы, а также скриншоты результатов согласно заданию.

#### Варианты:

1. «Агентство аренды»;
2. «Ателье»;
3. «Аэропорт»;
4. «Библиотека»;
5. «Кинотеатр»;
6. «Поликлиника»;
7. «Автосалон»;
8. «Таксопарк».

#### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое

представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 9 Построение ER-диаграмм в MS Visio

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

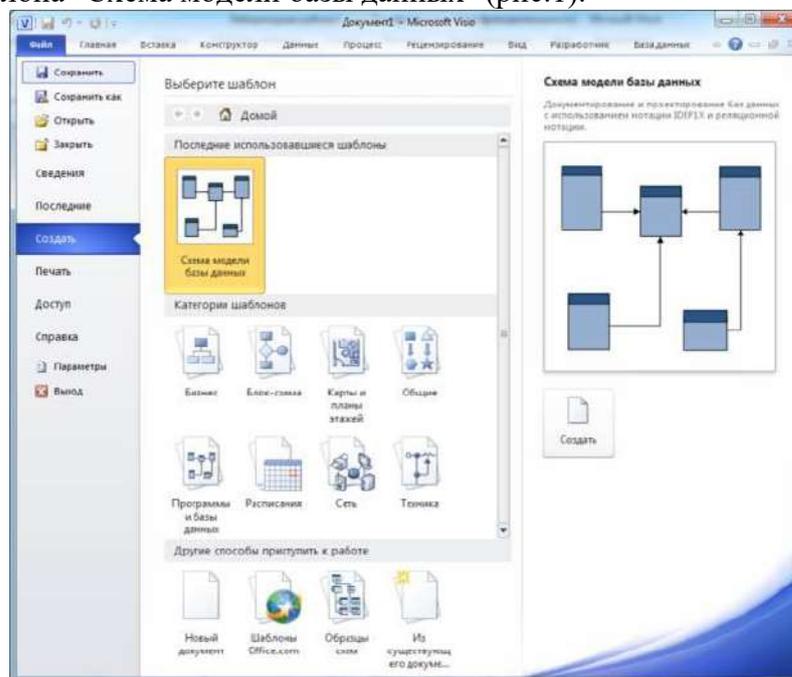
**Целью** работы является получение навыков работы с MS Visio

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio

### Практическая часть:

Создать новую или реконструировать существующую базу данных в модель можно с помощью шаблона "Схема модели базы данных" (рис.1).



**Рисунок 1. Создание документов**

### Анализ предметной области

В качестве примера можно использовать список всех сущностей, информацию о которых, потребуется сохранять в базе данных, причем, для каждой сущности определен необходимый список атрибутов и список всех связей, которые потребуются, в формате, указанном в таблице 1.

Таблица 1

Номер	Сущность	Атрибуты
1	Меры	дата/время, адрес, базовая станция, попыток установки соединений, успешно установленных соединений, исполнитель
2	ДатаВремяИзмерения	дата/время измерения

3	Адрес	город, улица, дом
4	Станция	номер
5	Исполнитель	фамилия, имя, отчество, отдел, должность

Примечание 1: информацию о дате и времени всегда лучше выносить за пределы таблицы фактов (главной таблицы) по нескольким причинам: размерность числового ключа обычно меньше, чем размерность типа данных для даты/времени - экономится место в главной таблице; в этом случае удобно использовать дополнительные флаги, например, рабочие/нерабочие дни, сезонности и т.п.

Примечание 2: разбиение значения адреса на более мелкие удобно, если, например, потребуется поднять результаты измерений для определенного города

Номер связи	Первая сущность	Вторая сущность	Тип связи (один-к-одному, один-ко-многим, многие-к-одному, многие-ко-многим)
1	Меры	ДатаВремяИзмерения	Многие-к-одному
2	Меры	Адрес	Многие-к-одному
3	Меры	Станция	Многие-к-одному
4	Меры	Исполнитель	Многие-к-одному

### Создание новой схемы модели базы данных

Если существующая база данных, которую требуется использовать в качестве основы, отсутствует, можно начать с пустой модели базы данных и добавить собственные таблицы и связи.

#### Параметры документа базы данных

1. Откройте вкладку **Файл**.
2. Выберите команду **Создать** и пункт **Программное обеспечение и базы данных**, а затем дважды щелкните элемент **Схема модели базы данных**.
3. На вкладке **База данных** в группе **Управление** нажмите кнопку **Показать параметры**.
4. В диалоговом окне **Параметры документа базы данных** выберите нужный набор символов и другие параметры таблицы и связи (рис.2), а затем нажмите кнопку **ОК**.

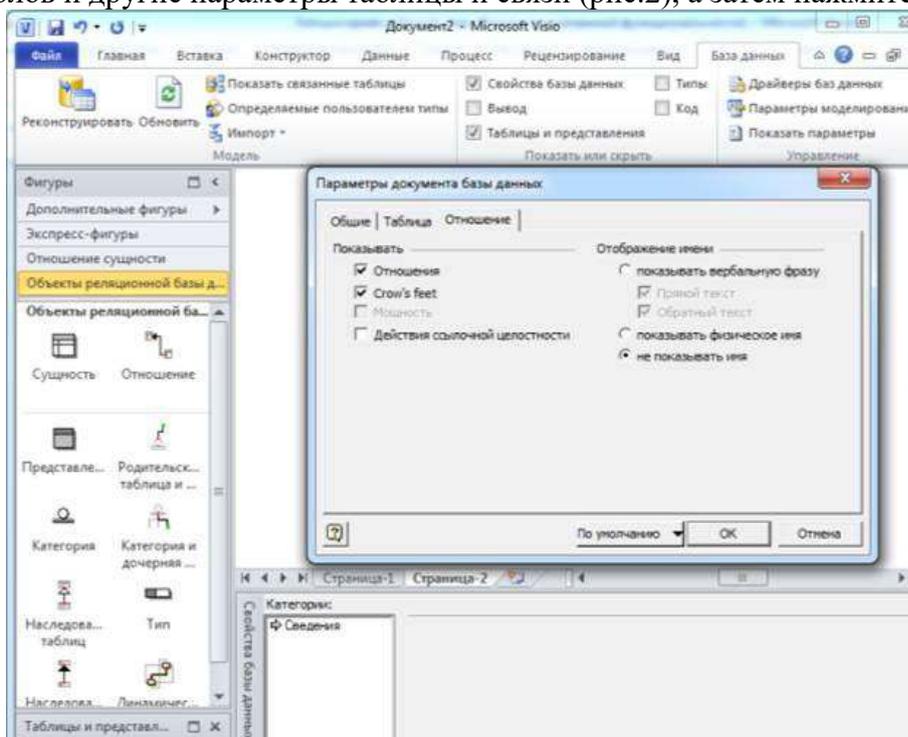


Рисунок 2. Параметры документа базы данных

## Создание таблиц

1. Установите удобный вам размер схемы (например, 100 %), а затем из контейнера **Объекты реляционной базы данных** перетащите на диаграмму элемент **Сущность**.
2. В списке **Категории** убедитесь, что выбрана строка **Определение** и введите следующие значения (табл.1):  
**Физическое имя**– *Меры*,  
**Концептуальное имя**– *Меры*,
3. Перейдите на строку **Столбцы**, убедитесь, что переключатель стоит в положении **Физический тип данных Microsoft Access**, введите имя и выберите тип данных, как показано на рисунке 3. Чтобы изменить тип данных для столбца, щелкните его поле **Тип данных**, а затем из списка выберите тип данных или введите тип данных в список. Поле **Заметки** заполнять не обязательно, значения в нем генерируются автоматически.
4. Установите флажок **Обязательное** для столбцов, которые не могут иметь значения NULL.
5. Установите флажок **PK** (первичный ключ) для столбцов, однозначно определяющих каждую строку таблицы базы данных.
6. В группе **Категории** выберите вариант **Индексы, Триггеры, Проверка** или **Дополнительные**, чтобы создать эти дополнительные элементы.
7. На вкладке **Конструктор** в группе **Темы** можно выбрать стиль оформления таблиц, например, **Цветов Яркая**, эффект **Простая тень** (рис.3)

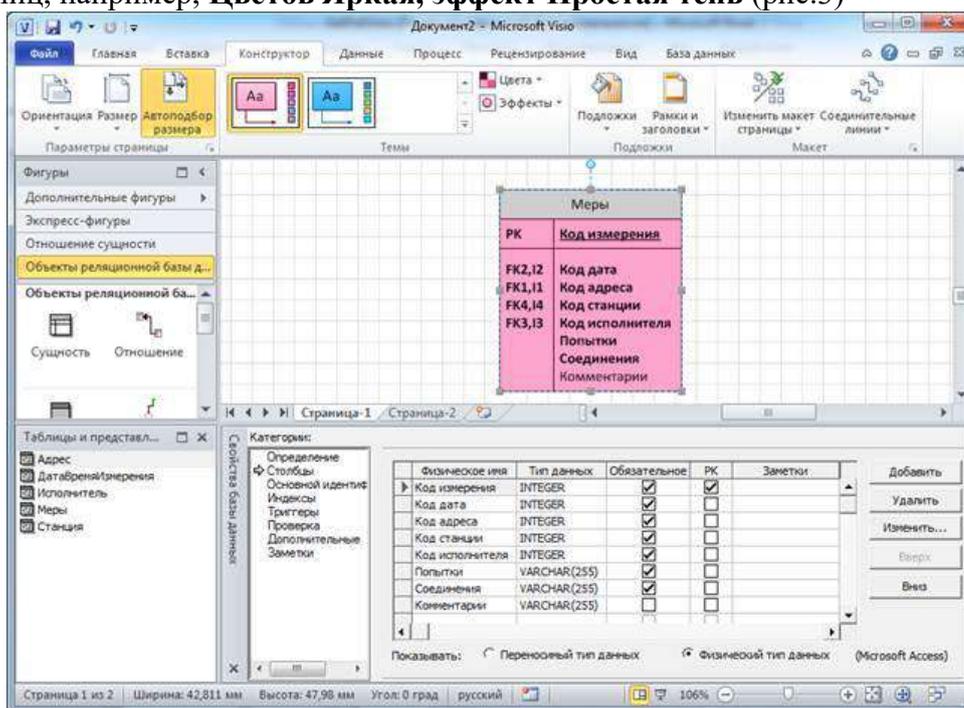


Рисунок 3.

## Проектирование сущности Меры

Точно также создайте еще сущности **Адрес**, **Станция**, **Исполнитель** и **ДатаВремяИзмерения** со столбцами, как показано на рисунке 4



а)



б)



в)



г)

**Рисунок 4. Проектирование сущностей: а) Адрес, б) Станция, в) Исполнитель, г) ДатаВремяИзмерения**

### Создание связей

После того, как создание всех объектов сущностей завершено, необходимо создать отношения между таблицами. Эта операция производится так:

1. Убедитесь, что в схеме отображены все таблицы.
2. Нажмите на кнопку **Соединительная линия** на вкладке **Главная** в группе **Сервис** (соседняя кнопка с указателем мыши не должна быть нажата).
3. Наведите указатель мыши на таблицу (родительскую) с первичным ключом (например, **ДатаВремяИзмерения**). Таблица будет выделена красным.
4. Перетащите таблицу (родительскую) с первичным ключом **ДатаВремяИзмерения** на таблицу (дочернюю) с внешним ключом **Меры**. Если все сделано правильно, то на схеме появится стрелка с красными квадратами на конце и в начале.

5. Если размерность связи не устраивает, то дважды щелкните связь. В окне **Свойства базы данных** в группе **Категории** выберите вариант **Прочее**. В группе **Мощность** выберите размерность, наиболее подходящую к связи. Для отношения «один-ко-многим» наилучшим вариантом будет **0 или более** или **1 или более** (Для отношения «один-к-одному» — **0 или 1** или **ровно 1**).

Создание связей в первый раз получается не всегда. Обратите внимание, что в Visio для столбцов, между которыми создаются отношения, должен совпадать тип данных и название (с учетом регистра). Иначе будет создано дополнительное поле с внешним ключом **FK 5 Код даты** (рис.5), а поле **FK 2 Код дата** останется без соединения. В СУБД (например, Access) такое условие не является обязательным. В итоге схема может выглядеть так, как показано на рисунке 5.

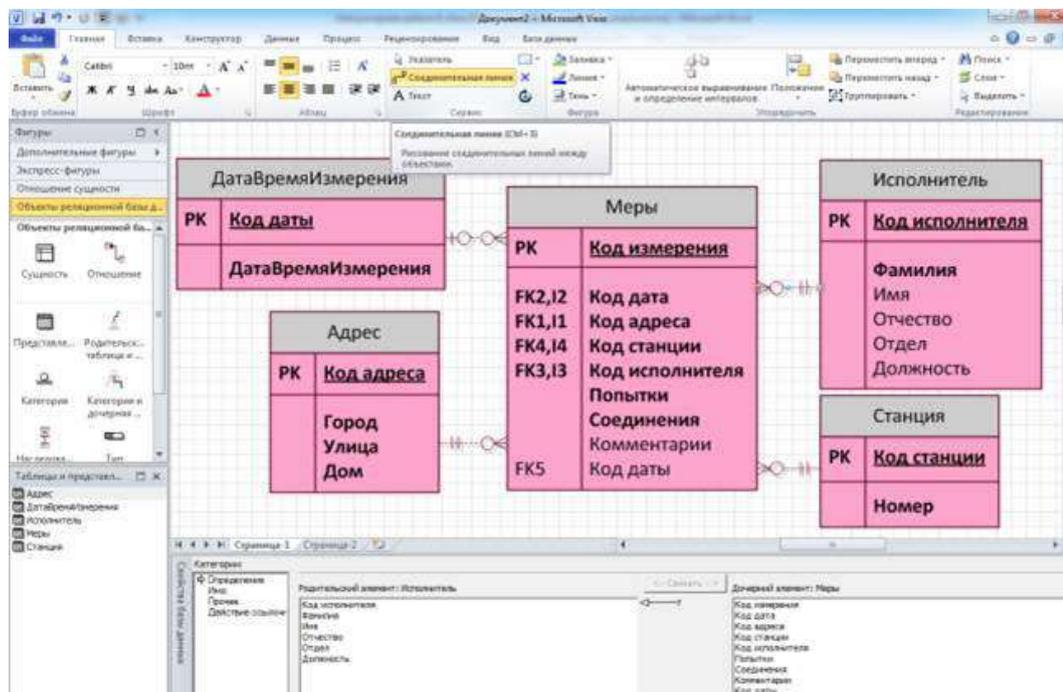


Рисунок 5. Схема Отношения между таблицами в Visio

### Создание индексов

Использование индексов повышает производительность или скорость работы базы данных при выполнении запроса.

1. Дважды щелкните таблицу (**Меры**), к которой требуется добавить индекс, а затем в окне **Свойства базы данных** в списке **Категории** выберите вариант **Основной идентификатор** и включите флажок **Создать индекс**.

2. Затем в окне **Свойства базы данных** в списке **Категории** выберите вариант **Индексы** и нажмите кнопку **Создать**.

3. В диалоговом окне **Создать индекс** введите имя для индекса (**Меры**), а затем нажмите кнопку **ОК**.

4. В списке **Тип индекса** выберите тип создаваемого индекса — **Только неуникальный**.

5. В списке **Доступные столбцы** выберите имя столбца **Код даты**, который требуется включить в этот индекс, а затем нажмите кнопку **Добавить**.

6. В списке **Индексированные столбцы** снимите флажок **Убыwanie**, чтобы создать индекс с возрастающим порядком сортировки (рис.6).

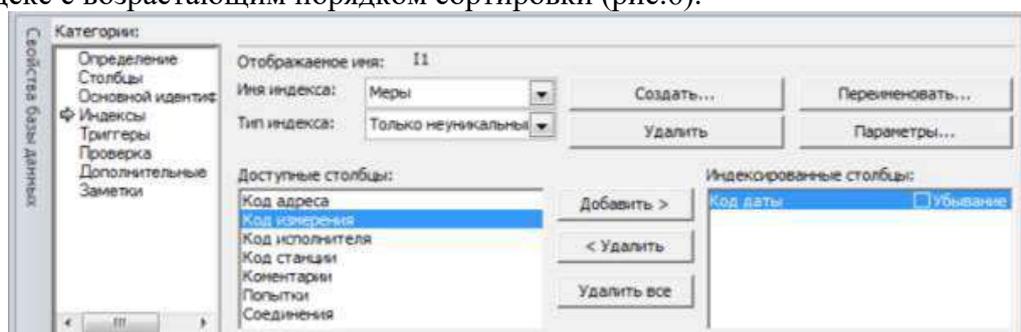


Рисунок 6. Создание индексов

### Создание проверки

Использование проверки позволяет контролировать работу базы данных при выполнении запроса.

Применение предложений проверки гарантирует, что данные, введенные в столбец, находятся в пределах конкретного диапазона значений. Например, можно создать предложение проверки того, что значения столбца «Возраст» превышают 65.

1. Дважды щелкните таблицу, чтобы открыть окно **Свойства базы данных**.

2. В группе **Категории** выберите вариант **Столбцы**, а затем выберите столбец **Соединение**, которому требуется добавить предложение проверки.
3. Нажмите кнопку **Изменить**.
4. На вкладке **Проверка** диалогового окна **Свойства столбца** введите ограничения, указанные на рисунке 7.

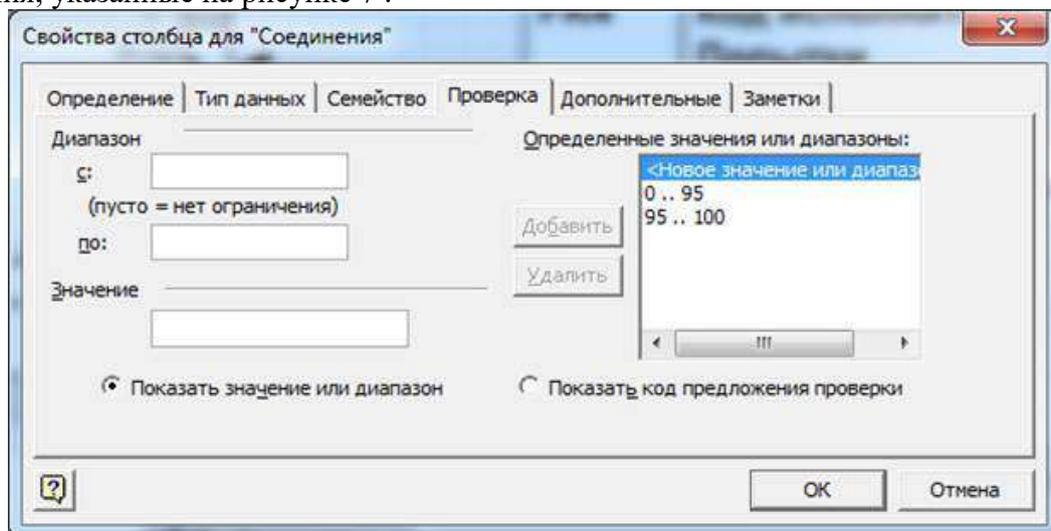


Рисунок 7. Создание диапазонов проверки

### Реконструирование схемы модели базы данных

Если существует база данных, которую нужно смоделировать для лучшего понимания ее структуры или применения в качестве основы создания новой модели, то для извлечения схемы или структуры базы данных можно воспользоваться мастером реконструирования. В качестве примера можно использовать базу данных созданную в Access 2010, по исходным данным, приведенным в таблице 1 (рис.8).

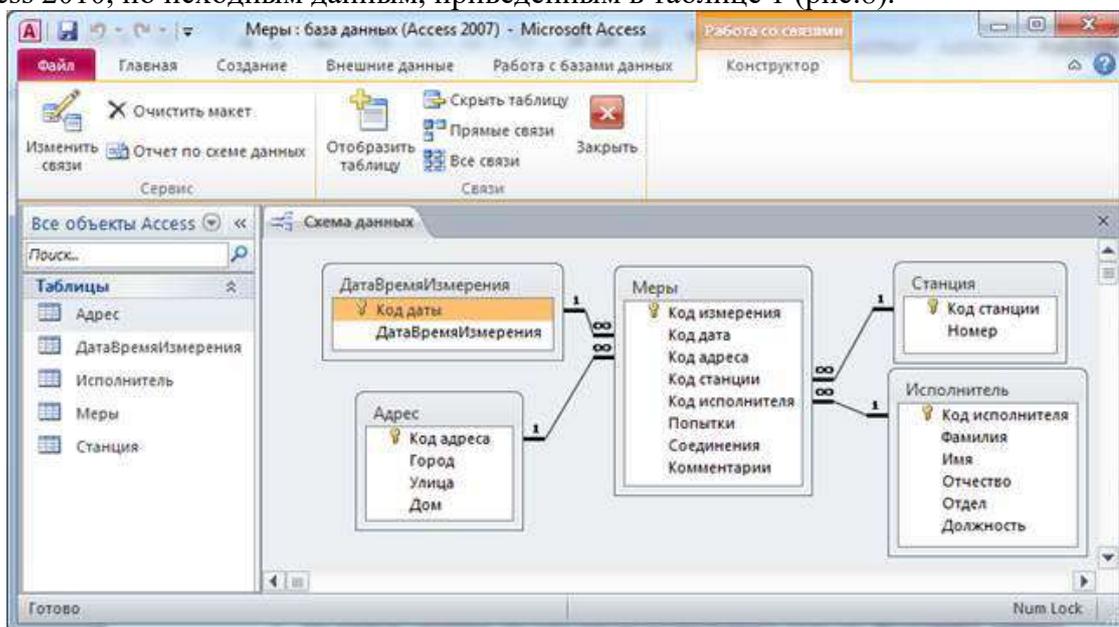


Рисунок 8. Схема Отношения между таблицами в Access

1. Выберите команду **Создать** и пункт **Программное обеспечение и базы данных**, а затем дважды щелкните элемент **Схема модели базы данных**.
2. На вкладке **База данных** в группе **Модель** нажмите кнопку **Реконструирование**.
3. На первой странице мастера реконструирования в качестве **Источника данных** укажите **Microsoft Access Database** и нажмите кнопку **Далее**.
4. Укажите свой **Логин**, **Пароль** и нажмите **ОК**.
5. Укажите **Имя базы данных (Меры. accdb)** и нажмите **ОК**.

6. Типы объектов для реконструирования оставьте по умолчанию и нажмите кнопку **Далее**.

7. Выберите все таблицы и представления для реконструирования, нажав кнопку **Выделить все** и нажмите кнопку **Далее**.

8. Выберите параметр **Да, добавить фигуры на текущую страницу** и нажмите кнопку **Далее**, затем **Да** и **Готово**.

В результате будут извлечены выбранные сведения, и в окне «Вывод» будут отображены примечания о процессе извлечения, т.е. должна получиться схема модели базы данных аналогичная схеме, приведенной на рисунке 9.

#### **Контрольные вопросы:**

1. Что такое ER-диаграмма?
2. Назначение ER-диаграммы?

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 10 Изучение возможностей EDRAW UML DIAGRAMM 7.9

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков работы с EDRAW UML DIAGRAMM

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: EDRAW UML DIAGRAMM

#### **Практическая часть:**

*Построить диаграмму вариантов использования модели работы банкомата.*

Выполните следующие действия:

1. Добавить актера с именем Клиент банкомата.
2. Добавить вариант использования Снятие наличных по кредитной карте.
3. Добавить направленную ассоциацию от бизнес-актера Клиент Банкомата к варианту использования Снятие наличных по кредитной карте
4. Добавить вариант использования Проверка ПИН-кода
5. Добавить актера с именем Банк.

6. Добавить вариант использования Получение справки о состоянии счета.
7. Добавить вариант использования Блокирование кредитной карточки.
8. Добавить направленную ассоциацию от бизнес-актера Клиент Банкомата к варианту использования Получение справки о состоянии счета.
9. Добавить направленную ассоциацию от варианта использования Снятие наличных по кредитной карточке к сервису Банк.
10. Добавить направленную ассоциацию от варианта использования Получение справки о состоянии счета к сервису Банк.
11. Добавить отношение зависимости со стереотипом «include», направленное от варианта использования Снятие наличных по кредитной карте к варианту использования Проверка Пин-кода.
12. Добавить отношение зависимости со стереотипом «include», направленное от варианта использования Получение справки о состоянии счета к варианту использования Проверка Пин-кода.
13. Добавить отношение зависимости со стереотипом «extend», направленное от варианта использования Блокирование кредитной карточки к варианту использования Проверка Пин-кода.

**Задание:**

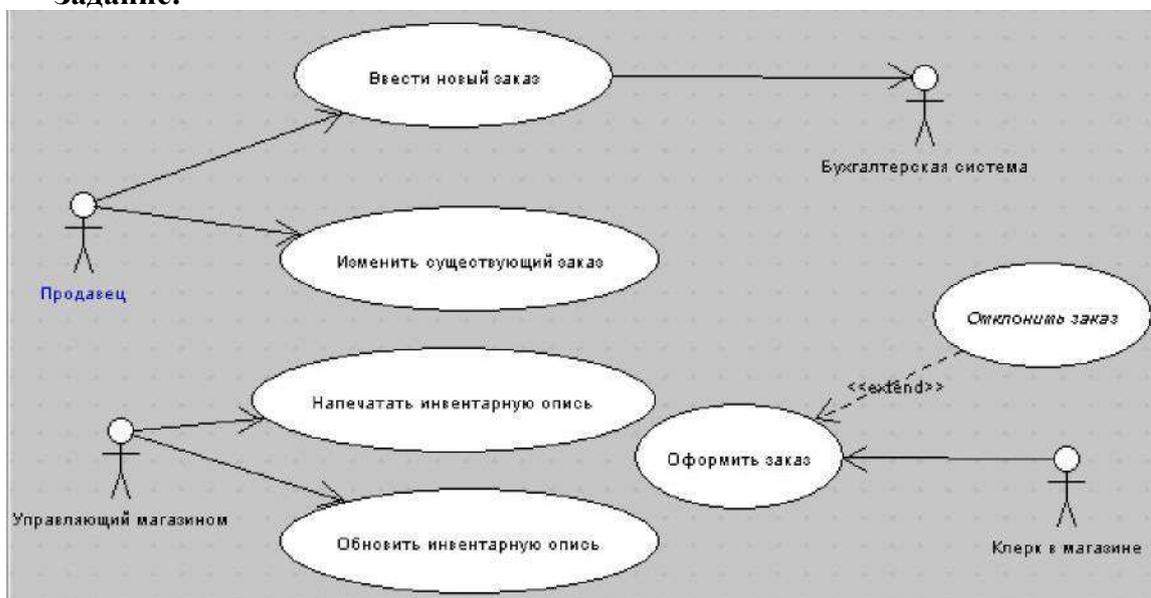


Рисунок 1 – Диаграмма вариантов использования для системы обработки заказов

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 11 Диаграмма классов (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы классов с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

### **Практическая часть:**

Построить диаграмму классов в соответствии с вариантом.

Отчет по практическому занятию выполняется в формате MS Word, который содержит пошаговое описание процесса построения диаграммы, а также скриншоты результатов согласно заданию.

1. «Агентство аренды»;
2. «Ателье»;
3. «Аэропорт»;
4. «Библиотека»;
5. «Кинотеатр»;
6. «Поликлиника»;
7. «Автосалон»;
8. «Таксопарк».

### **Контрольные вопросы**

1. Каково назначение диаграммы классов?
2. Назовите основные элементы диаграммы классов.
3. Какие виды связей доступны в диаграмме классов?
4. Для чего используется каждый вид связи?

### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;  
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 12 Диаграмма пакетов (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы пакетов с использованием инструментальных средств

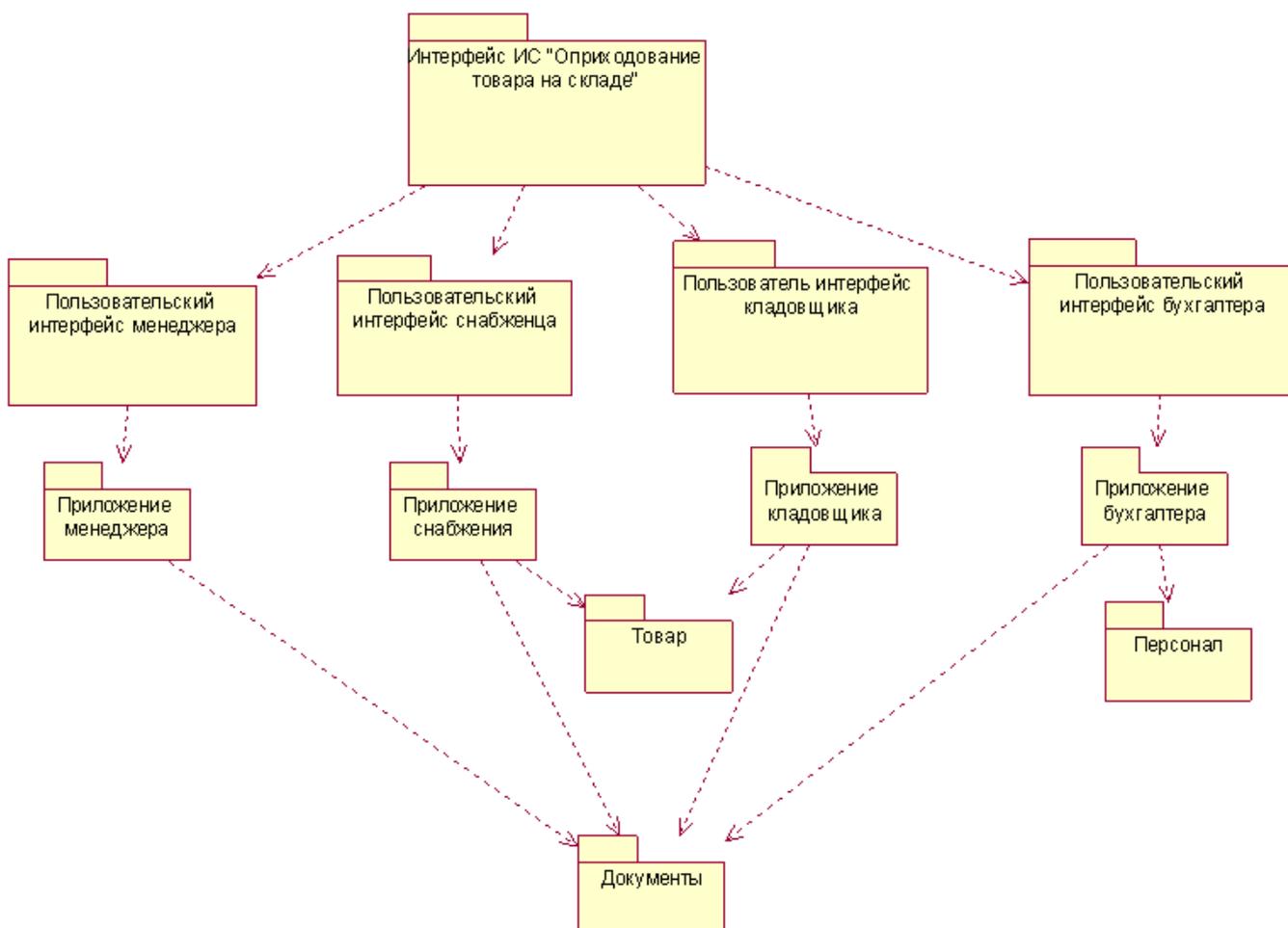
**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

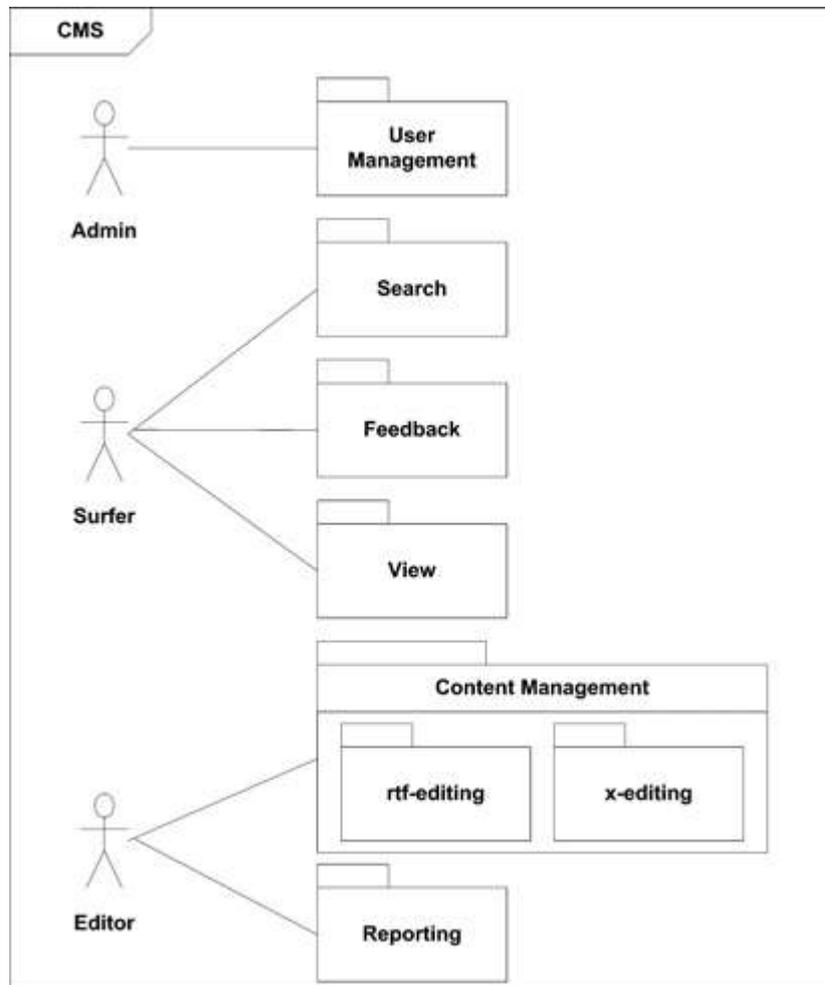
**Практическая часть:**

Построить диаграммы пакетов:

1.



2.



### Контрольные вопросы

1. Каково назначение диаграммы пакетов?
4. Для чего используется диаграмма пакетов?

#### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 13 Диаграмма объектов (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы объектов с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

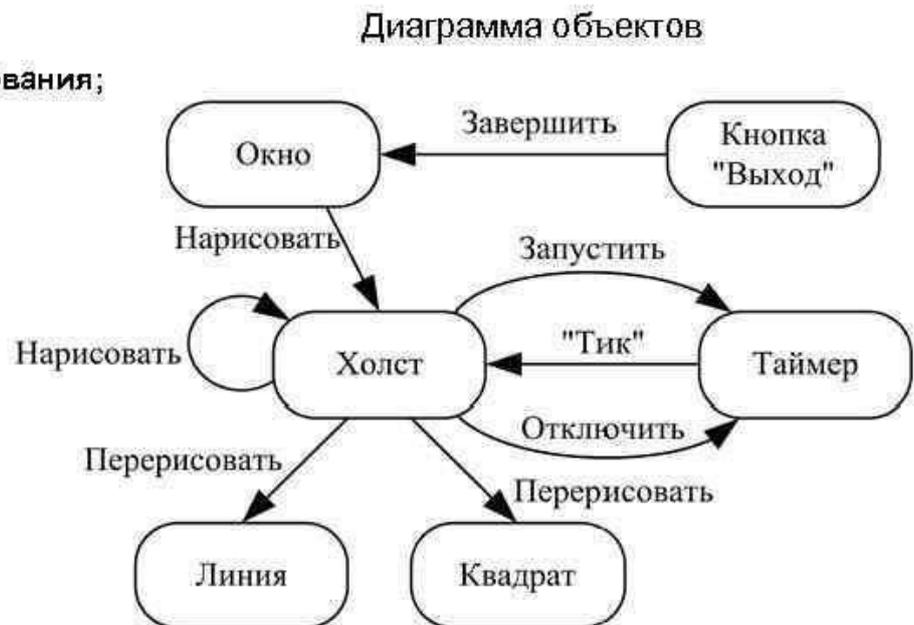
**Практическая часть:**

1. Нарисовать диаграмму объектов

### Диаграмма объектов приложения

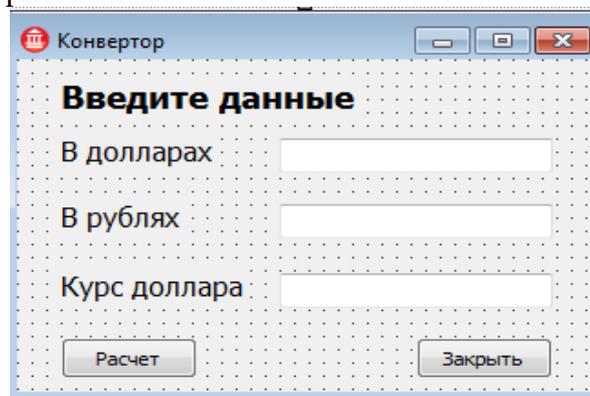
Приложение включает 6 объектов:

- окно;
- кнопка Выход;
- холст для рисования;
- таймер;
- линия;
- квадрат.



19

2. Создать приложение



3. Разработать диаграмму объектов для приложения.

## Контрольные вопросы

1. Для чего используются диаграммы объектов?

### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 14      Диаграмма      последовательностей (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы последовательностей с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

### **Практическая часть:**

Нарисовать диаграмму последовательности для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.

15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы**

1. Для чего используются диаграммы последовательности?

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

### Практическое занятие 15 Диаграмма состояний (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы состояний с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

#### **Практическая часть:**

Нарисовать диаграмму состояний для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компаний.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубежом).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы**

1. Для чего используются диаграммы состояний?

##### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 16 Диаграмма деятельности (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы деятельности с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

### **Практическая часть:**

Нарисовать диаграмму деятельности для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

### **Контрольные вопросы**

1. Для чего используются диаграммы деятельности?

### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 17      Диаграмма компонентов (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы компонентов с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

**Практическая часть:**

Нарисовать диаграмму компонентов для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.

18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы**

1. Для чего используются диаграммы компоненты?

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое

представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

### Практическое занятие 18 Диаграмма развертывания (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы компонентов с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

#### **Практическая часть:**

Нарисовать диаграмму развертывания для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.

4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы**

1. Для чего используются диаграммы развертывания?

##### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 19 Диаграмма коопераций (EDRAW UML DIAGRAMM 7.9 или MS VISIO 2010)

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения диаграммы компонентов с использованием инструментальных средств

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

**Практическая часть:**

Нарисовать диаграмму коопераций для варианта:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубежом).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

**Контрольные вопросы**

1. Для чего используются диаграммы кооперации?

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 20      Проектирование      графического интерфейса пользователя

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения графического интерфейса пользователя

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

**Практическая часть:**

1 Создайте карту навигации для выбранной системы. На карте в зависимости от специфики системы выделите разделы, доступные различным пользователям в зависимости от роли, опишите условия перехода из различных разделов (при необходимости)

2 Используя графический редактор на выбор, создайте макеты графического интерфейса пользователя. Предлагаемые системы:

- Microsoft Visio 2010

3 Для разработанных макетов подготовьте их текстовое описание в следующем виде:

Название поля	Тип	Условия видимости	Условия доступности	Описание
				Формат, допустимые значения, макс. и мин. длина, поведение

Создать карту навигации для выбранной системы:

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .

6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы**

1. Что такое графический интерфейс пользователя?
2. Какие бывают виды графического интерфейса?
3. Что такое карта навигации.

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое

представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 21 Элементы графического интерфейса

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

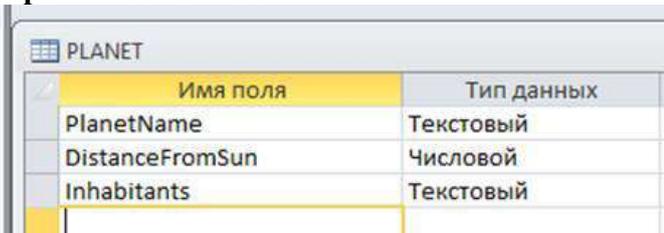
- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения графического интерфейса пользователя

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

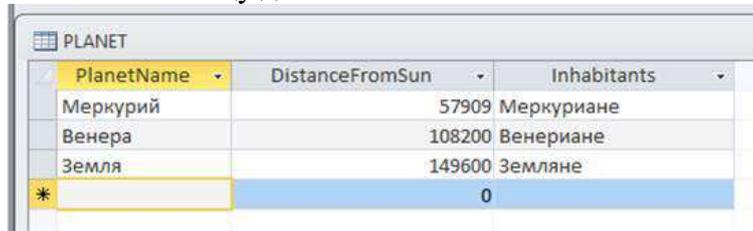
Программное обеспечение: Microsoft Visio, EDRAW UML DIAGRAMM

**Практическая часть:**



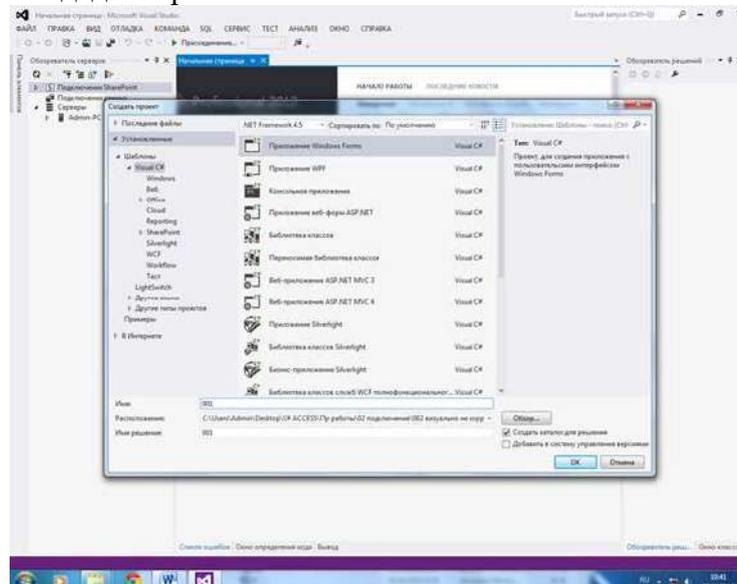
Имя поля	Тип данных
PlanetName	Текстовый
DistanceFromSun	Числовой
Inhabitants	Текстовый

Заполним таблицу данными

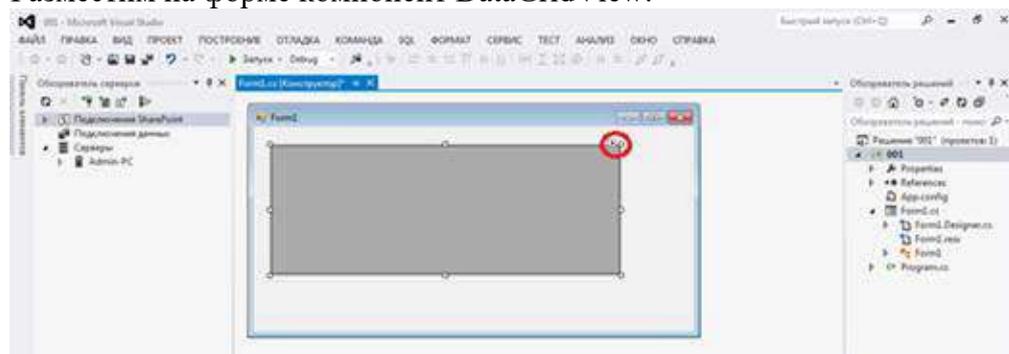


PlanetName	DistanceFromSun	Inhabitants
Меркурий	57909	Меркуриане
Венера	108200	Венериане
Земля	149600	Земляне
*	0	

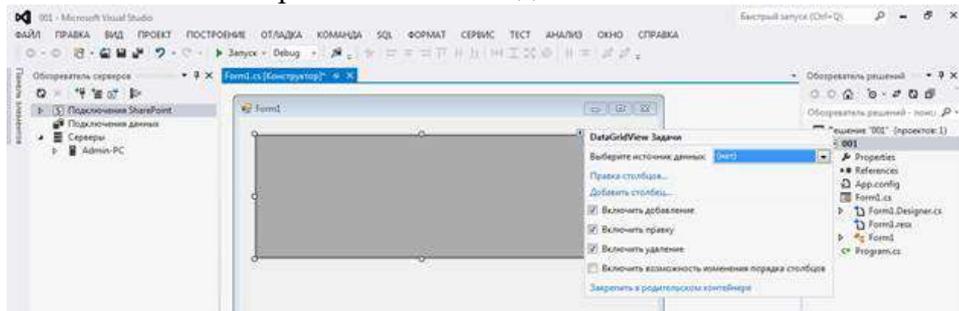
Создадим проект в Visual Studio C#.



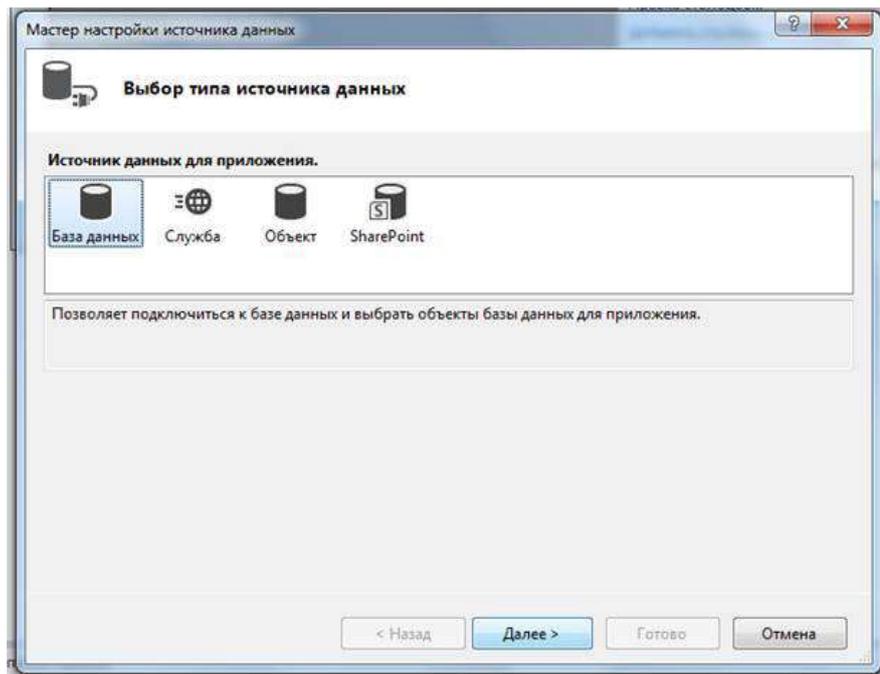
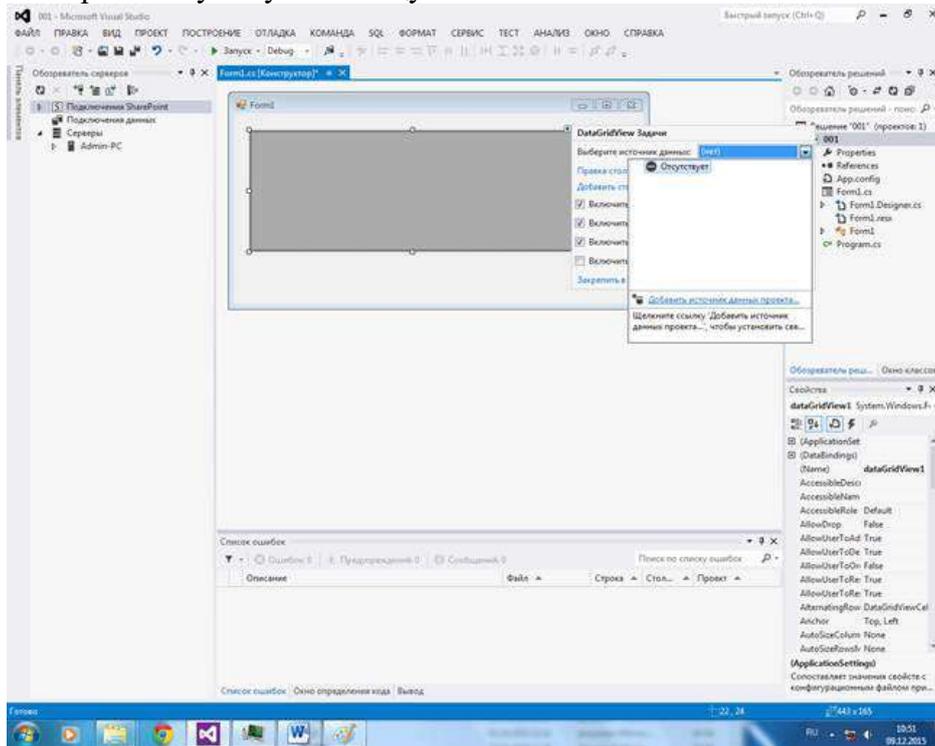
Разместим на форме компонент DataGridView:

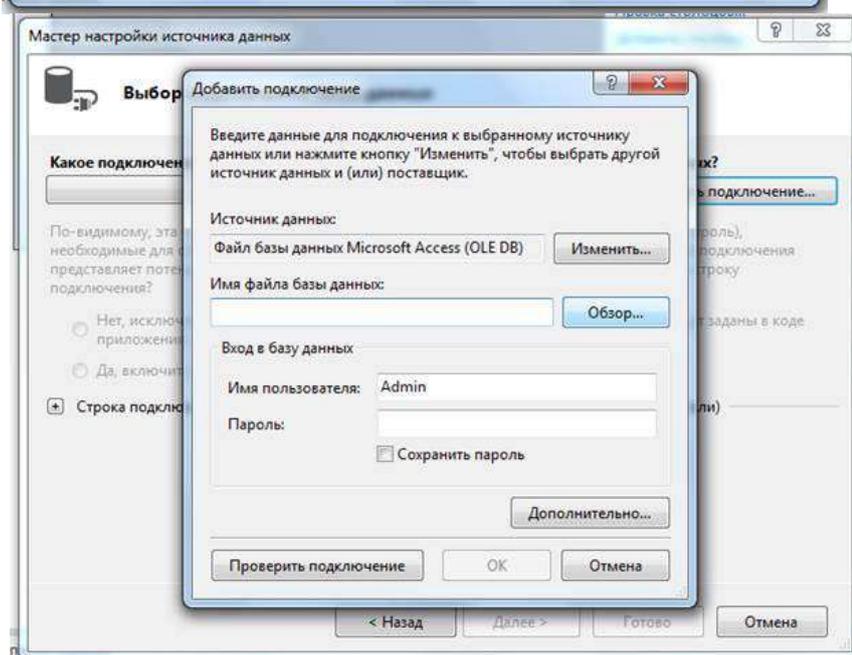
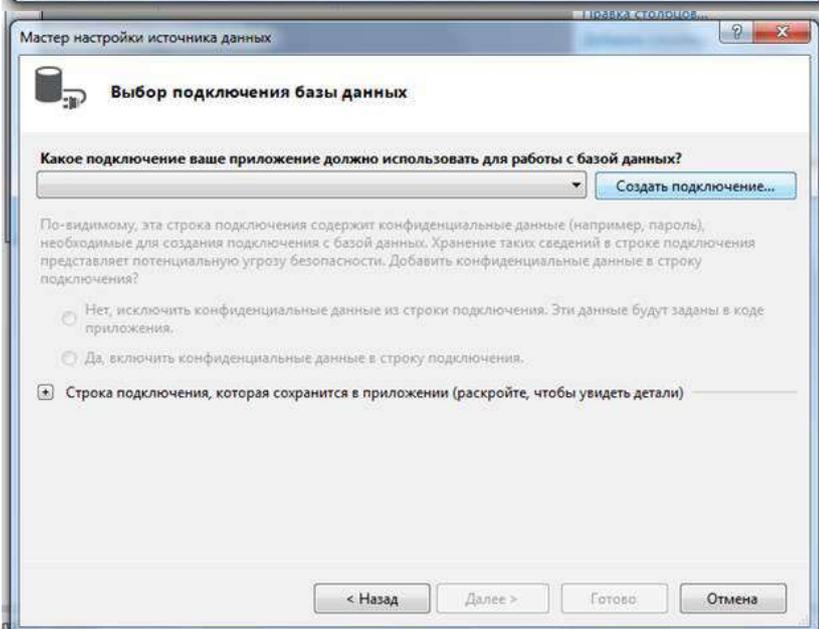
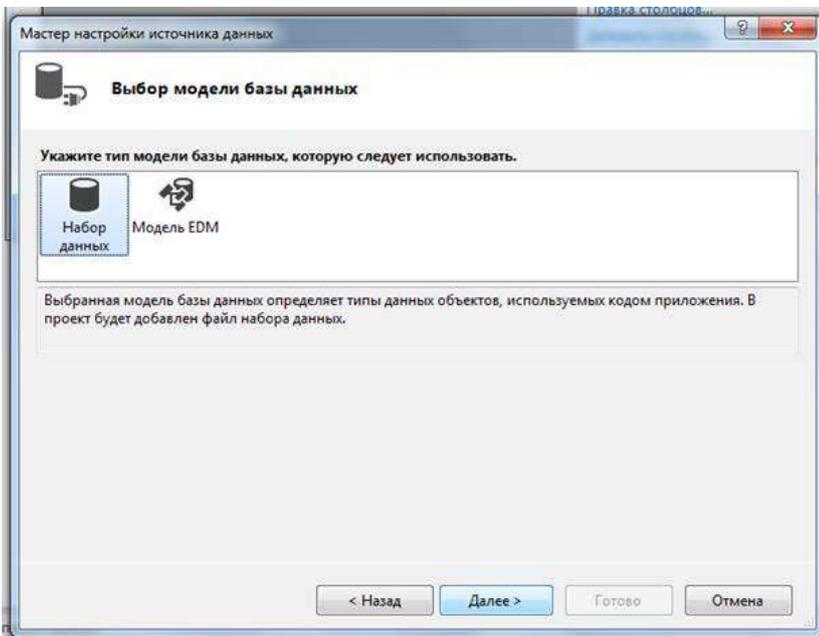


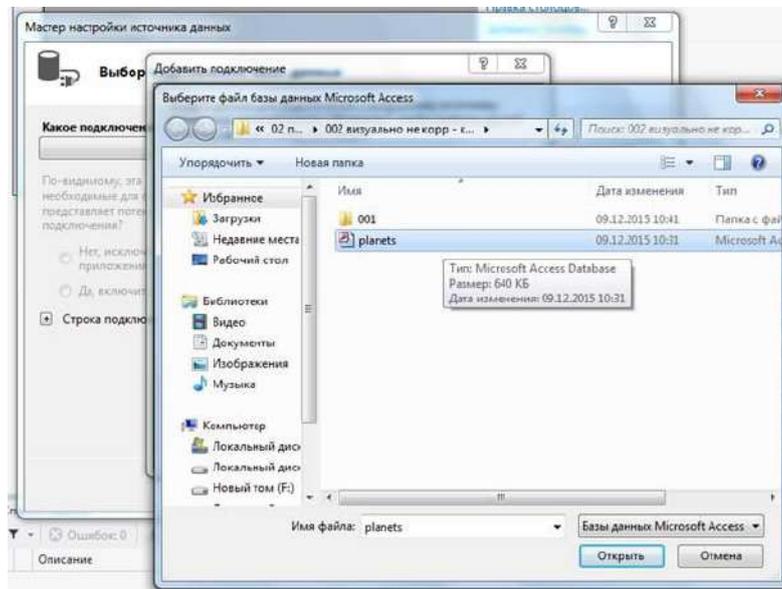
Нажмем на черную стрелочку в правом верхнем углу компонента DataGridView. Появится окно настройки источника данных:



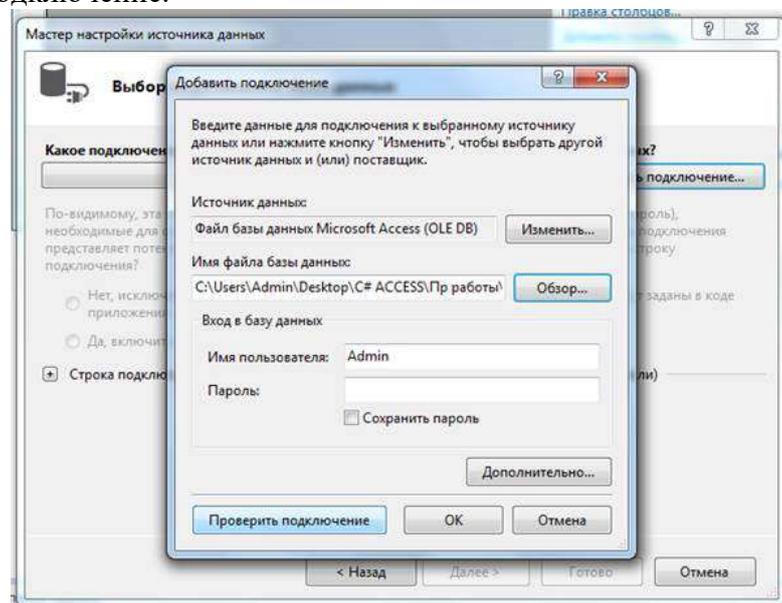
Выберем нашу базу и таблицу PLANET.



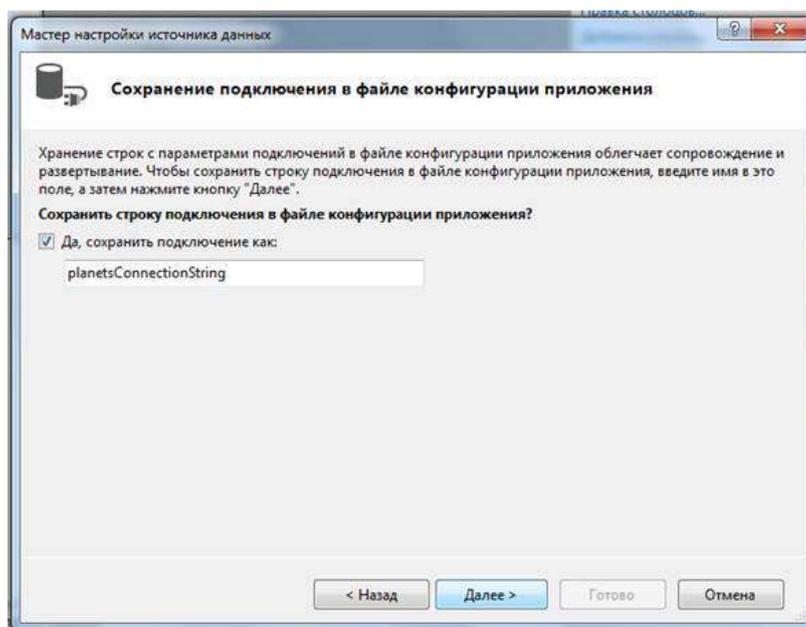
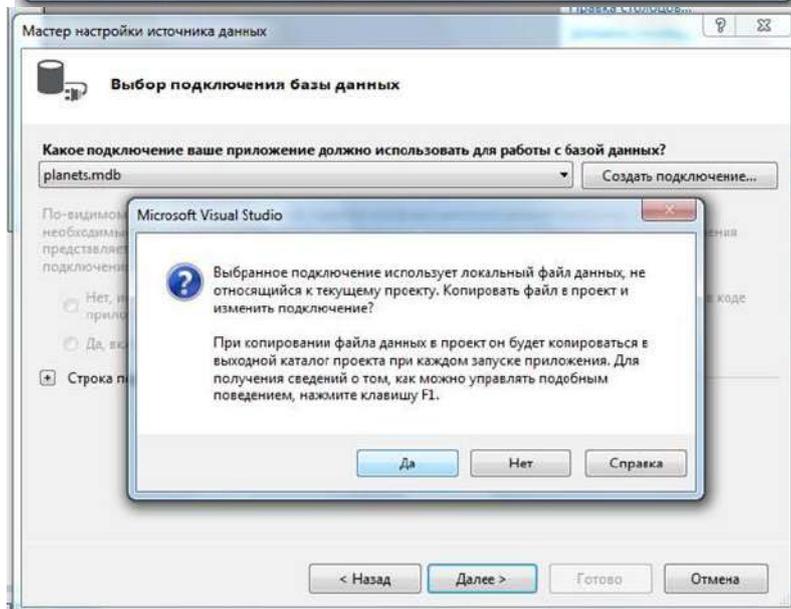
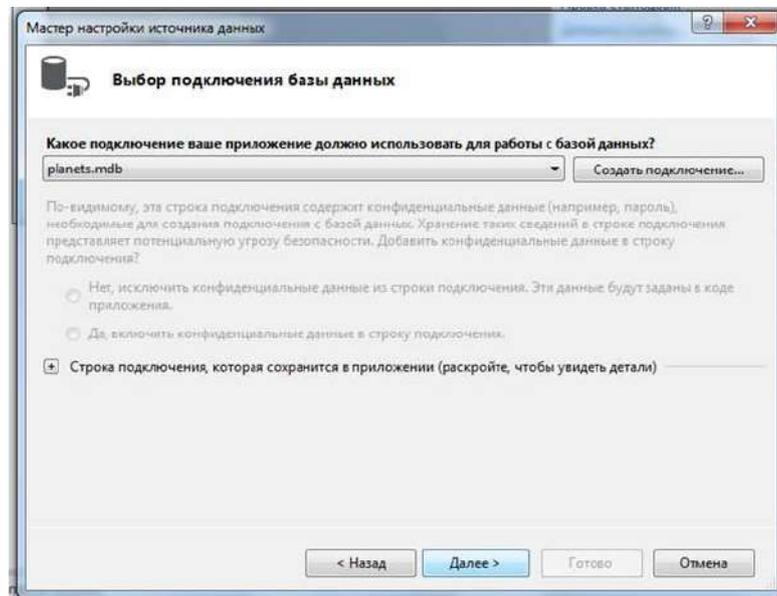




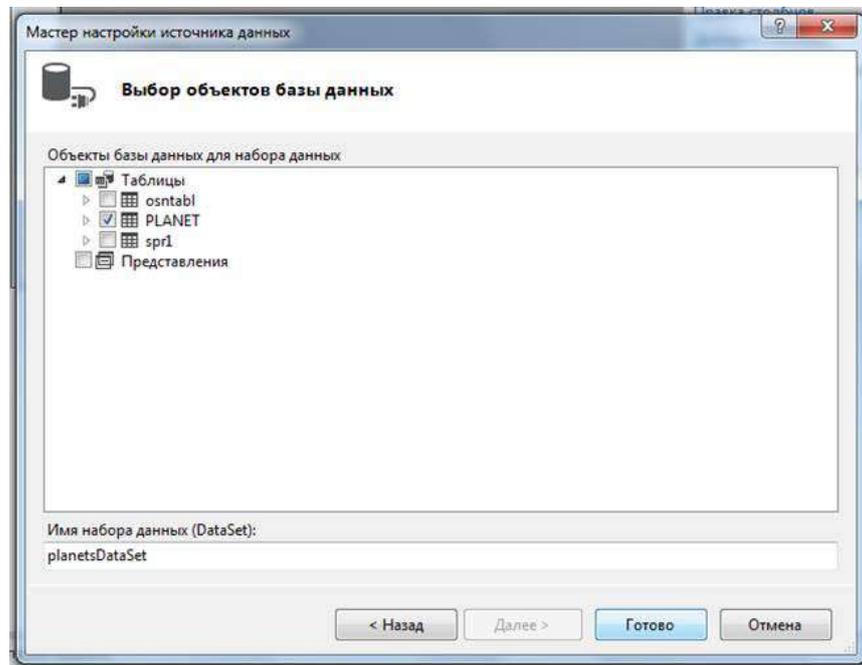
Проверить подключение.



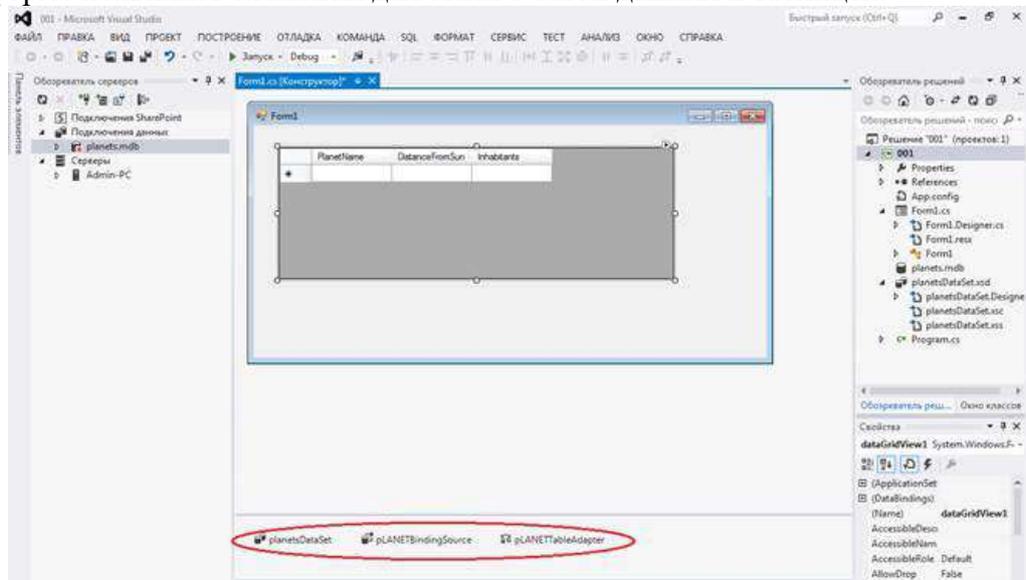
ОК, ОК.



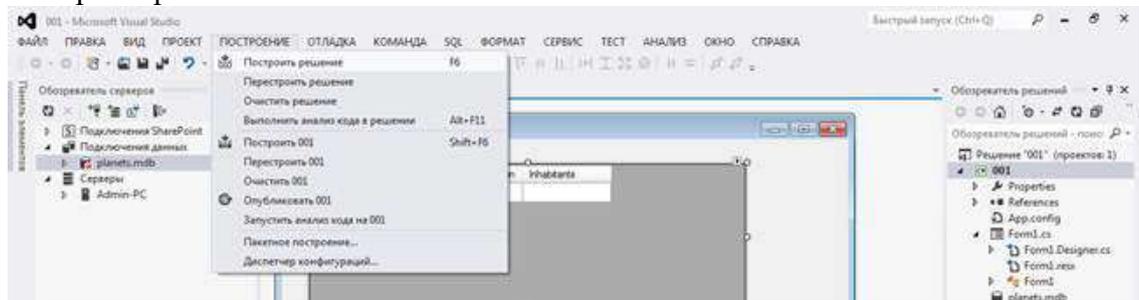
Выбрать таблицу PLANET.



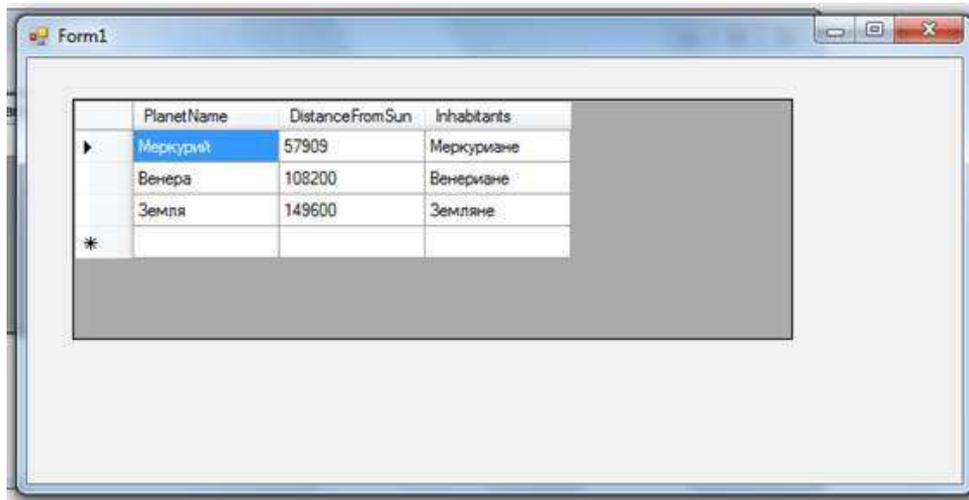
На форме появились элементы для связи с базой данных и таблицей.



Построим решение



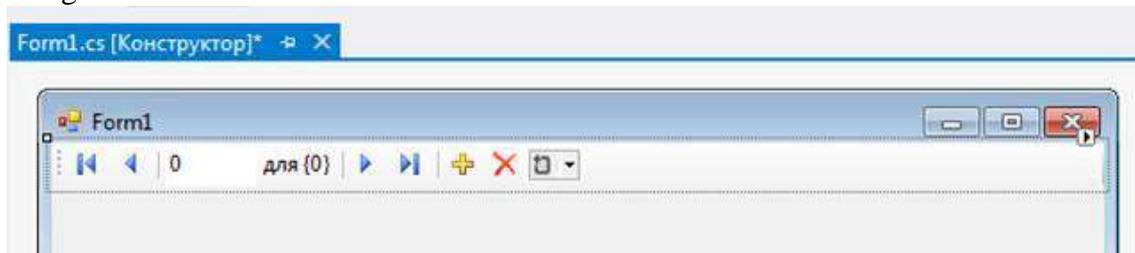
Запустим проект на выполнение:



Протестируем полученное приложение.

В базе данных не сохраняются внесенные изменения (вставка, удаление, корректировка).

Можно проделать следующий эксперимент: установить на форме компонент BidingNavigator.



Результат будет такой же – изменения не сохраняются в базе данных.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace _002
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 'planetsDataSet.PLANET' table. You can
            // move, or remove it, as needed.
            this.pLANETTableAdapter.Fill(this.planetsDataSet.PLANET);
        }
    }
}

```

### **Задание**

Создать базу данных MySQL согласно своему варианту задания.

Создать приложение для демонстрации подключения базы данных в Visual Studio C# по технологии ADO.NET/ Протестировать приложение.

1. Автострахование.
2. Агентство по сдаче автомобилей в аренду.
3. Аренда коньков, роликов, велосипедов, лыж.
4. Аэропорт – пассажирское расписание и перевозки.
5. Банковская система вкладов (физических и юридических лиц) .
6. Банковская система кредитования (физических и юридических лиц).
7. Биллинг сотовой компании.
8. Ветеринарная лечебница.
9. Клуб обучения танцам.
10. Магазин косметики.
11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).
12. Нефтеперерабатывающая компания.
13. Парикмахерская.
14. Поставка вин.
15. Приемная комиссия ВУЗа.
16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).
17. Рекламное агентство.
18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубежом).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 22 Элементы графического интерфейса-2

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** работы является получение навыков проектирования и построения графического интерфейса пользователя

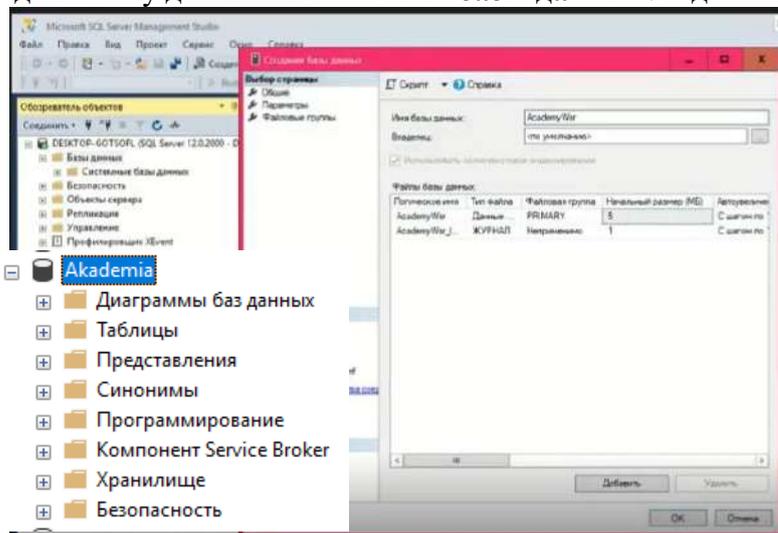
**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio,

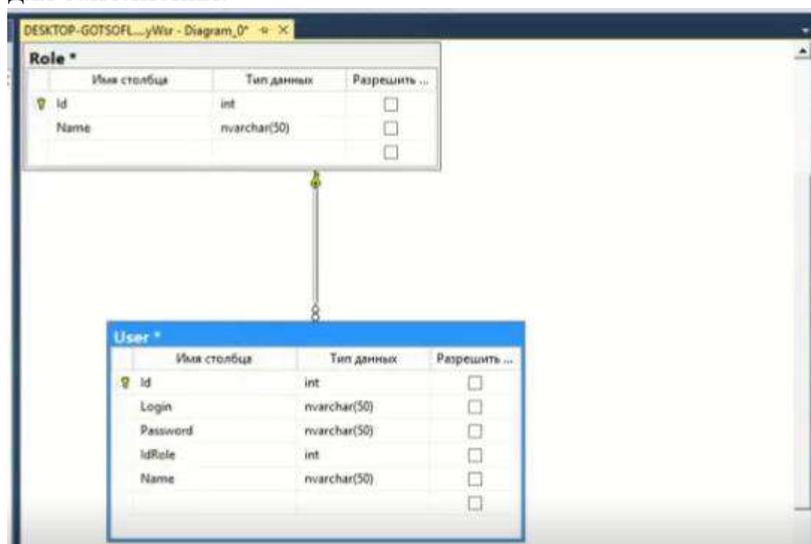
**Практическая часть:**

### Создание БД

1. Запустите MS SQL Server Management Studio.
2. Создайте базу данных. В поле **Имя базы данных:** задайте имя БД.



3. Создайте в БД две таблицы в соответствии с диаграммой (**User** и **Role**), необходимые для реализации в приложении авторизации. Задайте для них ключевое поле в виде счетчика (**Спецификация идентификатора - Да**). Все поля должны быть обязательными для заполнения.



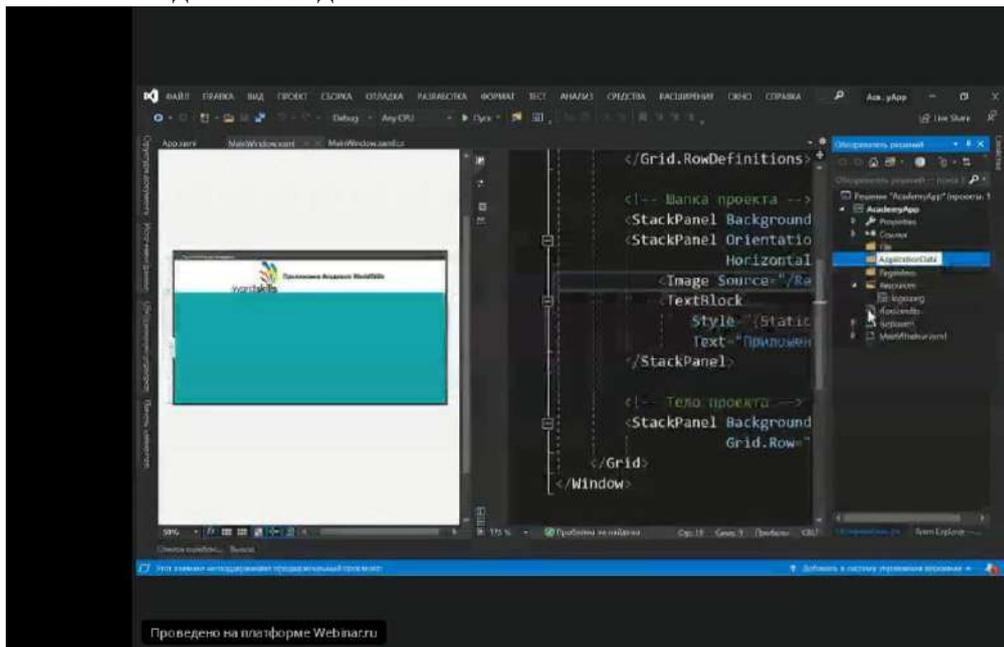
4. Заполните таблицы данными. Предполагаем, что у нас будет два пользователя, один из них Администратор (Степан), другой – Ученик (Наталья). Других пользователей будем добавлять через приложение.

Id	Name
1	Администратор
2	Ученик
NULL	NULL

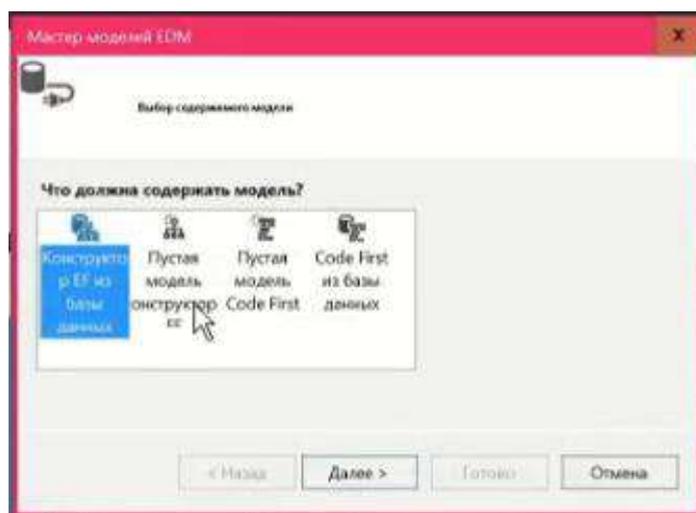
Id	Login	Password	IdRole	Name
1	1	1	1	Степан
2	2	2	2	Наталья
NULL	NULL	NULL	NULL	NULL

### Создание модели БД

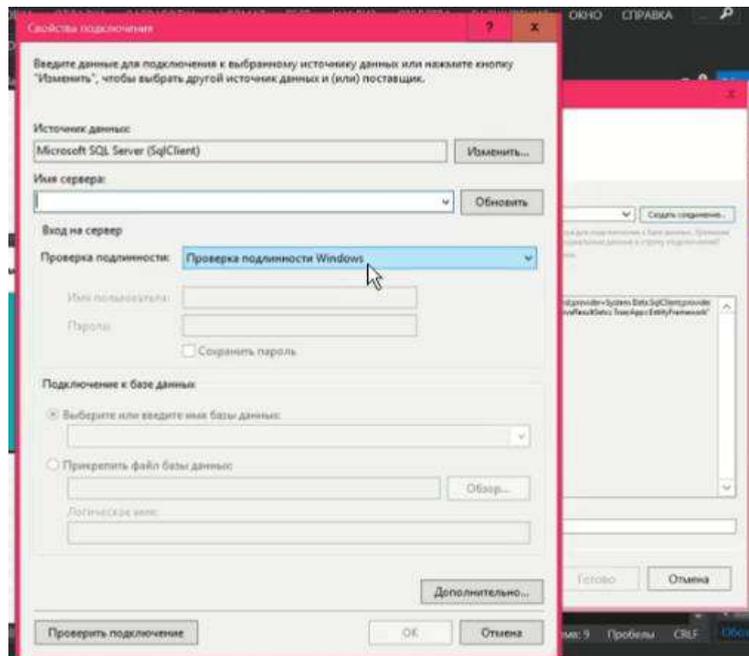
5. Создайте приложение WPF в MS Visual Studio.
6. Добавьте в проект в окне Обозреватель решений папку ApplicationData для размещения в ней модели базы данных.



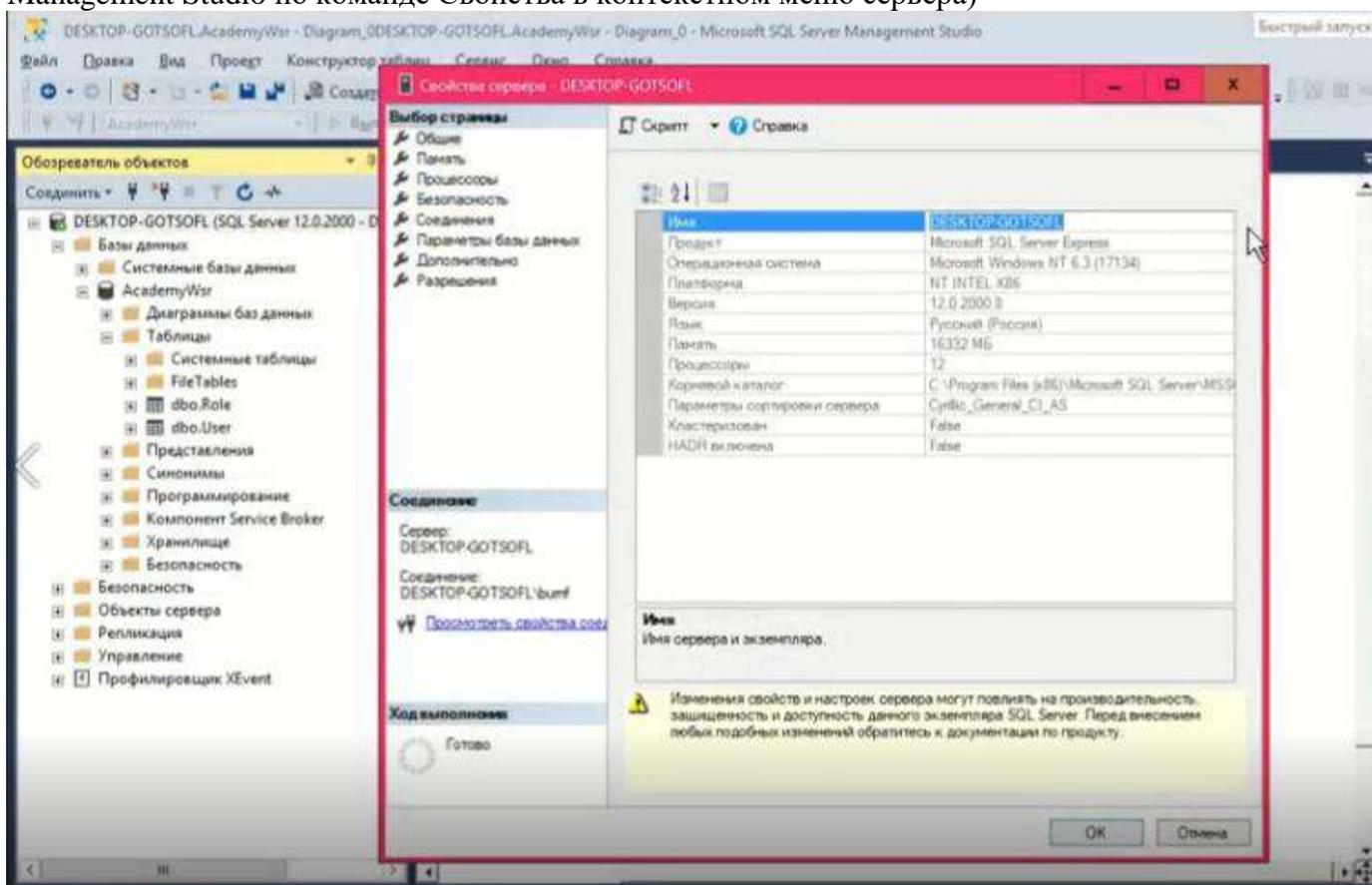
7. В папке ApplicationData создайте элемент Модель ADO.NET.EDM (имя модели можно не менять) Model1. Выберите нужную модель.



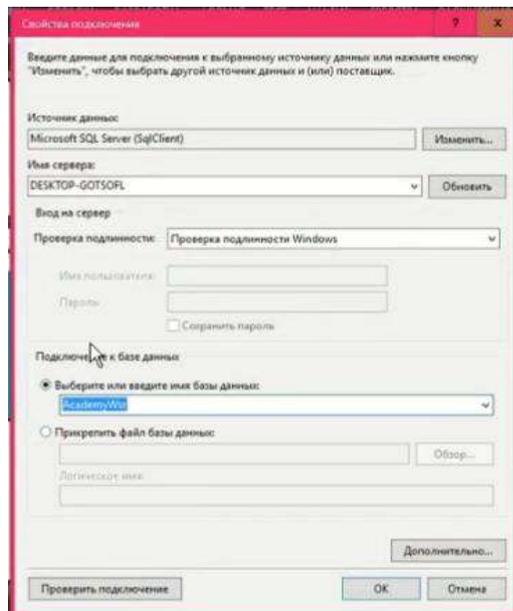
8. Далее Создать соединение:



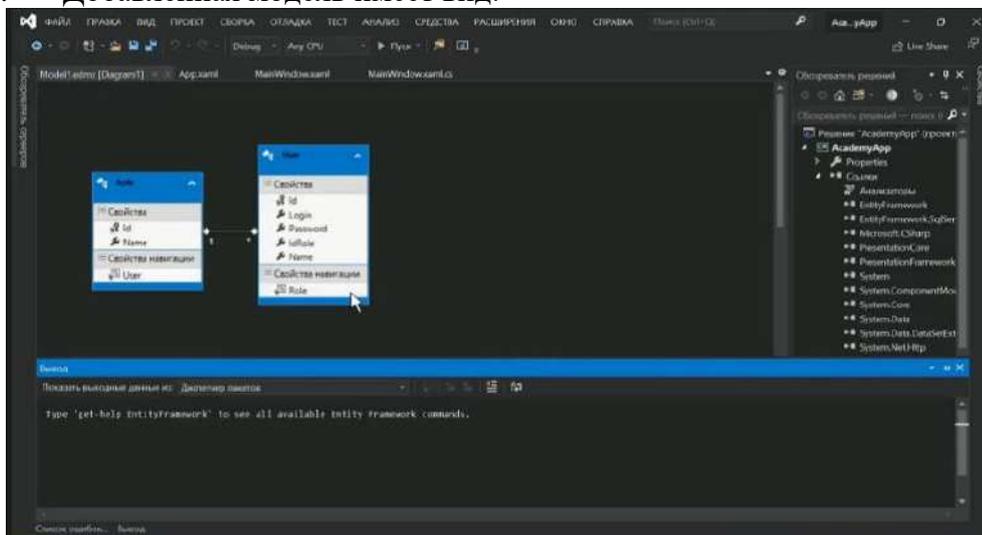
9. написать Имя сервера (имя сервера можно взять в MS SQL Server Management Studio по команде Свойства в контекстном меню сервера)



10. выбрать вариант входа на сервер
11. выбрать имя базы данных



12. Выбрать версию Framework и на последнем шаге выбрать таблицы БД.
13. Добавленная модель имеет вид:



При изменении структуры базы данных в VS необходимо удалить таблицы модели, затем на свободном месте щелкнуть правой кнопкой мыши и выбрать команду Обновить модель из базы данных.

В обозревателе решений проекта отображается файл созданной модели **Model1.edmx**.

### Подключение к БД

В папку **ApplicationData** добавить класс **AppConnect**, написать в нем подключение к нашей БД.



Подключить ту же самую модель как объект к главному окну **MainWindow**.

```

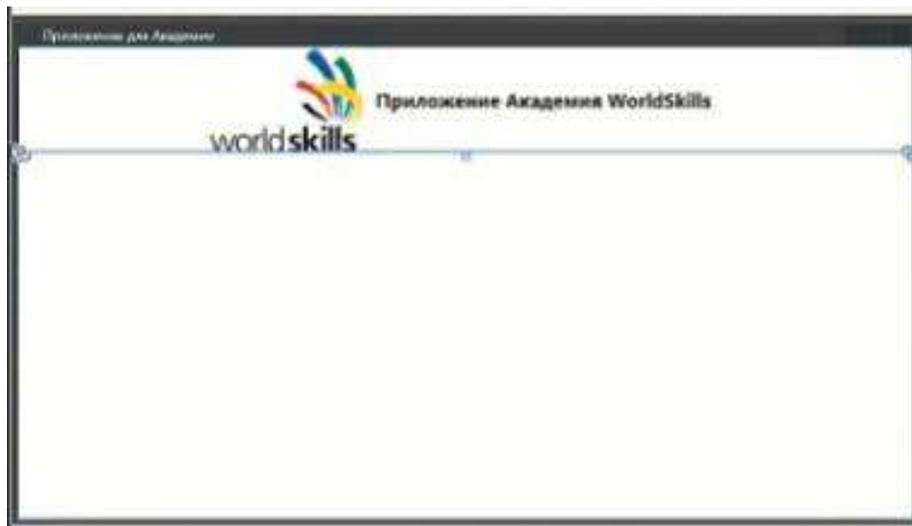
public partial class MainWindow : Window
{
    Ссылка: 0
    public MainWindow()
    {
        InitializeComponent();
        AppConnect.model0db = new AcademyWsrEntities();
    }
}

```

На этом Подключение БД к проекту завершено!

### Интерфейс приложения (главное окно)

1. Организуйте разметку главного окна приложения в соответствии с рисунком:



Ширину окна установите 800, высоту 350, измените текст в строке заголовка. Основной контейнер компоновки в окне - **Grid**, состоящий из двух строк:

```

<Grid.RowDefinitions>
    <RowDefinition Height="100*" />
    <RowDefinition Height="350*" />
</Grid.RowDefinitions>

```

В **верхней строке сетки Grid** размещен контейнер компоновки **StackPanel** с горизонтальным размещением в нем двух элементов: **Image** и **TextBlock**. Изображение логотипа поместить в папку проекта **Resources**. Стиль оформления надписи организовать в ресурсах файла **App.xaml**, например, в таком виде (можно изменить стиль на свое усмотрение):

```

<Style TargetType="TextBlock" x:Key="Title">
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="FontSize" Value="13pt" />
    <Setter Property="FontWeight" Value="Bold" />
    <Setter Property="Margin" Value="15" />
</Style>

```

и применить к элементу **TextBlock**:

```

Style="{StaticResource Title}"

```

В **нижней строке сетки Grid** разместить **Frame**. Дать ему имя **FrmMain**, скрыть в нем панель навигации:

2. Чтобы внутри данного фрейма выполнять действия (хранить и передавать информацию из страницы к странице), создадим специальный класс, и объект этого класса с именем **FrmMain**.

В паку **ApplicationData** добавьте класс **AppFrame**:

```
class AppFrame
{
    public static Frame frameMain;
}
```

Пропишите фрейм в коде главного окна:

```
public MainWindow()
{
    InitializeComponent();
    AppConnect.modelOdb = new AcademyWsrEntities();
    AppFrame.frameMain = FrmMain;
}
```

### Добавление страниц

Для каждой вновь добавляемой страницы приложения будем организовывать в проекте отдельную папку.

### Страница авторизации

1. Создайте в проекте папку **PageMain**. В ней создайте страницу **PageLogin**. Установите высоту 350, ширину 800.

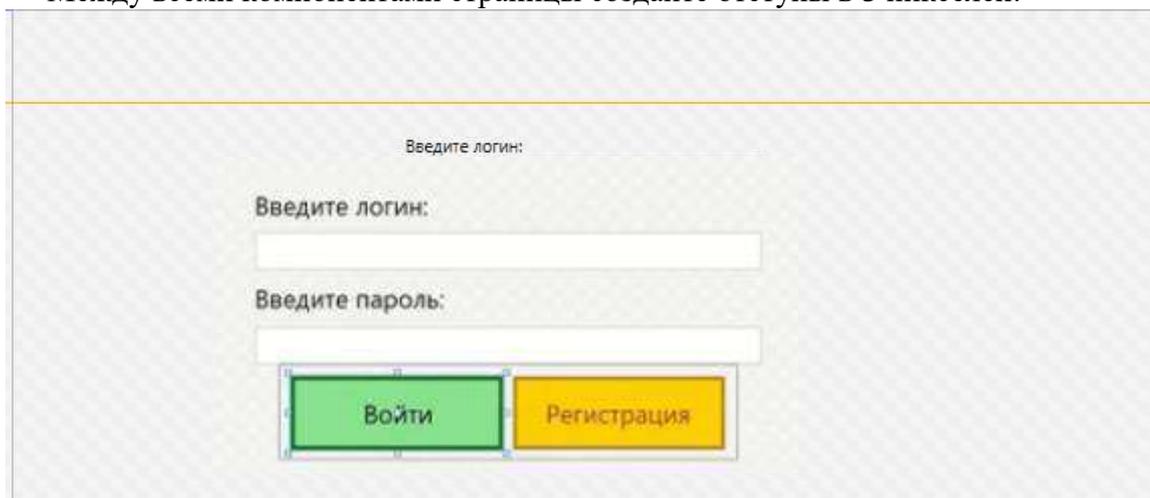
Организируйте разметку страницы в соответствии с рисунком.

Используйте контейнер **StackPanel**, расположите его в центре страницы.

Для ввода пароля используйте компонент **PasswordBox**.

Для оформления кнопок добавьте еще два шаблона в ресурсы файла **APP.xaml**.

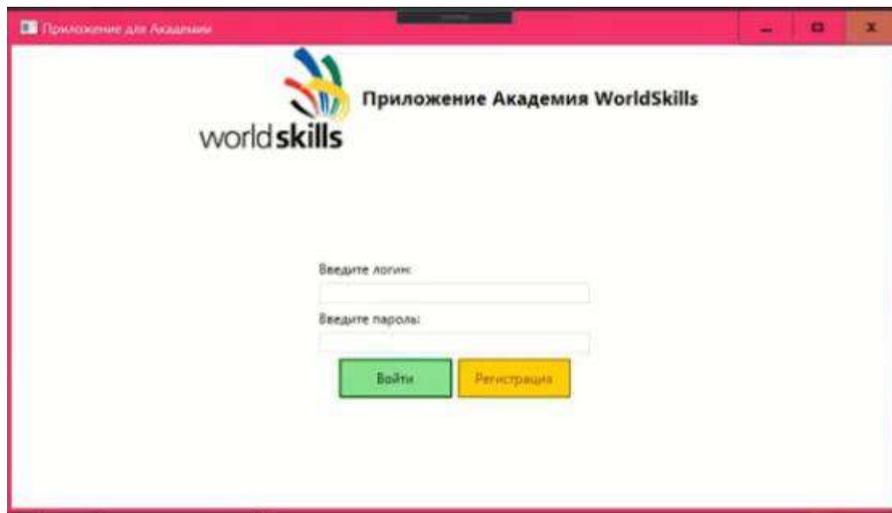
Между всеми компонентами страницы создайте отступы в **5** пикселей:



2. Данная страница авторизации должна загружаться при запуске приложения. Для этого нужно в главном окне внутри фрейма инициализировать нашу страницу:

```
public MainWindow()
{
    InitializeComponent();
    AppConnect.modelOdb = new AcademyWsrEntities();
    AppFrame.frameMain = FrmMain;
    FrmMain.Navigate(new PageLogin());
}
```

Запустите проект, убедитесь, что страница авторизации загружается в окне приложения!



### Функционал авторизации

Создайте обработчик события **Click** для кнопки **Войти**.

```
try
{
    var userObj = AppConnect.model0db.User.FirstOrDefault(x => x.Login == txbLogin.Text && x.Password == psbPassword.Password);
    if (userObj==null)
    {
        MessageBox.Show("Такого пользолвателя нет!", "Ошибка при авторизации!",
            MessageBoxButton.OK, MessageBoxImage.Error);
    }
    else
    {
        switch (userObj.IdRole)
        {
            case 1:MessageBox.Show("Здравствуйете, Администратор " + userObj.Name + "!",
                "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
                break;
            case 2:
                MessageBox.Show("Здравствуйете, Ученик " + userObj.Name + "!",
                    "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
                    break;
            default: MessageBox.Show("Данные не обнаружены!", "Уведомление",MessageBoxButton.OK,MessageBoxImage.Warning);
                break;
        }
    }
}
catch (Exception Ex)
{
    MessageBox.Show("Ошибка " + Ex.Message.ToString() + "Критическая работа приложения!",
        "Уведомление",MessageBoxButton.OK, MessageBoxImage.Warning);
}
}
```

#### Пояснения к коду:

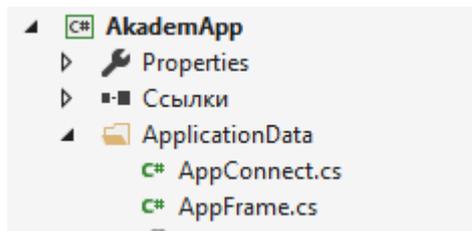
Когда идет работа с данными рекомендуют использовать **try..catch** (обработку исключений).

Внутри оператора **try** нужно написать процесс подключения к базе данных и работы с ней.

```
var userObj = AppConnect.model0db.User.FirstOrDefault(x =>
    x.Login == txbLogin.Text && x.Password == psbPassword.Password);
```

Создадим динамическую переменную **userObj**. Будем внутри нее хранить данные, получаемые из полей для ввода логина и пароля.

Обращаться к БД будем посредством созданного нами ранее класса **AppConnect** и работать непосредственно с ним.



Модель **modelOdb** видит входящие в БД таблицы в качестве атрибутов (**User**).

При наборе строки

```
var userObj = AppConnect.modelOdb.User.
```

после очередной точки перечисляются методы работы с **LINQ**-запросами и **EntityFramework**.

Нам нужен метод **FirstOrDefault** (он возвращает первый элемент последовательности...).

Создаем анонимную переменную **x**, используемую внутри LINQ-запроса (аналогично параметру цикла **for**).

Возвращаем в переменную **userObj** все значения из таблицы **User**, которые совпали по введенным в текстовые поля логину и паролю.

Если в БД пользователя нет, то нужно вывести об этом сообщение (**if...**), иначе – нужно вывести приветствие для соответствующего пользователя.

**Запустите проект, убедитесь, что на экран выводится каждое из организованных в коде приветствий и уведомлений!**

### Страница регистрации

1. В папке **PageMain** создайте страницу **PageCreateAcc**. Установите высоту страницы 350, ширину – 800.

Создадим интерфейс для возможности отправки в БД введенных значений нового пользователя.

Организируйте разметку страницы в соответствии с рисунком.

Используйте контейнер **StackPanel** с вертикальной ориентацией, разместите его в центре страницы.

В нем организуйте еще 4 контейнера **StackPanel** горизонтальной ориентацией.

Для ввода пароля используйте компонент **TextBox**, для подтверждения пароля – **PasswordBox**.



Оформление компонентов организуйте с помощью стилей, шаблоны которых пропишите в файле **App.xaml**.

По нажатию на кнопку **Регистрация** страницы **Авторизация** должна открываться страница для создания аккаунта ученика. При нажатии на кнопку **Назад** страницы **Регистрация** мы должны вернуться на страницу **Авторизация**.

2. Создайте обработчик события **Click** для кнопки **Регистрация**:

```
private void btnRegIn_Click(object sender, RoutedEventArgs e)
{
    AppFrame.frameMain.Navigate(new PageCreateAcc());
}
```

3. Создайте обработчик события **Click** для кнопки **Назад**:

```
private void btnBack_Click(object sender, RoutedEventArgs e)
{
    AppFrame.frameMain.GoBack();
}
```

Пояснения к коду:

Все действия на главном окне у нас выполняются во фрейме.

Этот фрейм хранится в объекте **frameMain**.

Мы загрузили наш фрейм с главной страницы.

**Запустите проект, убедитесь, что осуществляются соответствующие переходы по кнопкам Регистрация и Назад!**

**Функционал прохождения регистрации ученика**

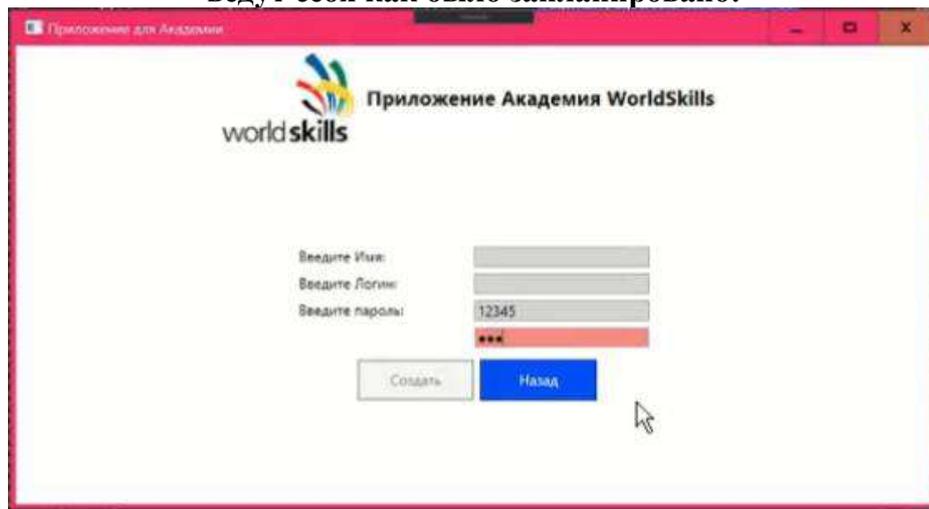
1. Кнопка **Создать** должна быть недоступна, пока в поле подтверждения пароля вводимое значение не совпадет с первоначальным. А поле подтверждения пароля окрашивается красным пока нет совпадения и зеленым в момент совпадения значений в полях.

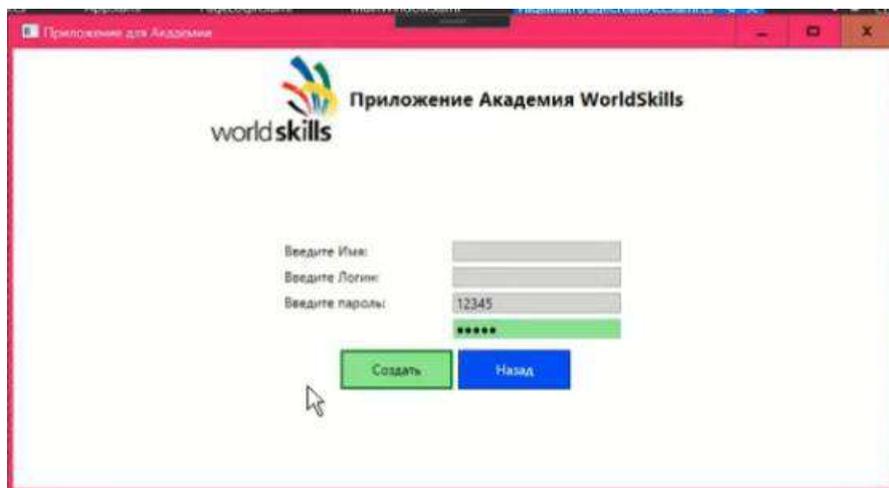
Для этого:

- Кнопку Создать сделать изначально неактивной **IsEnabled="False"**
- Создайте обработчик события **PasswordChanged** компонента **PasswordBox**:

```
private void PasswordBox_PasswordChanged(object sender, RoutedEventArgs e)
{
    if (psbPass.Password!=txbPass.Text)
    {
        btnCreate.IsEnabled = false;
        psbPass.Background = Brushes.LightCoral;
        psbPass.BorderBrush = Brushes.Red;
    }
    else
    {
        btnCreate.IsEnabled = true;
        psbPass.Background = Brushes.LightGreen; ;
        psbPass.BorderBrush = Brushes.Green;
    }
}
```

**Запустите проект. Убедитесь, что кнопка Создать и поле подтверждения пароля ведут себя как было запланировано!**





2. Внутри обработчика события **Click** кнопки **Создать** нужно организовать добавление в базу данных значений, вводимых пользователем в соответствующие текстовые поля. Для этого необходимо обратиться к таблице **User** базы данных, и эта таблица должна получить значения из компонентов страницы **Регистрация** (Имя, Логин и Пароль).

```
private void btnCreate_Click(object sender, RoutedEventArgs e)
{
    if (AppConnect.model0db.User.Count(x => x.Login==txbLogin.Text)>0)
    {
        MessageBox.Show("Пользователь с таким логином есть!",
            "Уведомление",MessageBoxButton.OK,MessageBoxImage.Information);
        return;
    }
    try
    {
        User userObj = new User()
        {
            Login = txbLogin.Text,
            Name = txbName.Text,
            Password = txbPass.Text,
            IdRole=2
        };
        AppConnect.model0db.User.Add(userObj);
        AppConnect.model0db.SaveChanges();
        MessageBox.Show("Данные успешно добавлены!",
            "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
    }
    catch
    {
        MessageBox.Show("Ошибка при добавлении данных!",
            "Уведомление", MessageBoxButton.OK,MessageBoxImage.Error);
    }
}
```

Пояснения к коду:

Сначала нужно проверить, есть ли пользователь с таким логином в системе.

Если значения, возвращаемые из таблицы **User** по полю **Login** имеются (применяем для этой цели метод **Count** LINQ-запроса), то вывести уведомление о том, что пользователь с таким логином существует в БД.

Для добавления данных нового пользователя нужно использовать конструкцию **try..catch**.

Создаем новый объект **UserObj** класса **User**, присваиваем его свойствам значения из компонентов страницы **Регистрация**. Регистрировать можно только учеников, поэтому присваиваем значение 2 полю **idRole**. Записываем созданный объект в таблицу БД **User**

(метод **Add**) и сохраняем изменения (метод **SaveChanges**). По окончании выводим уведомление о проделанной операции.

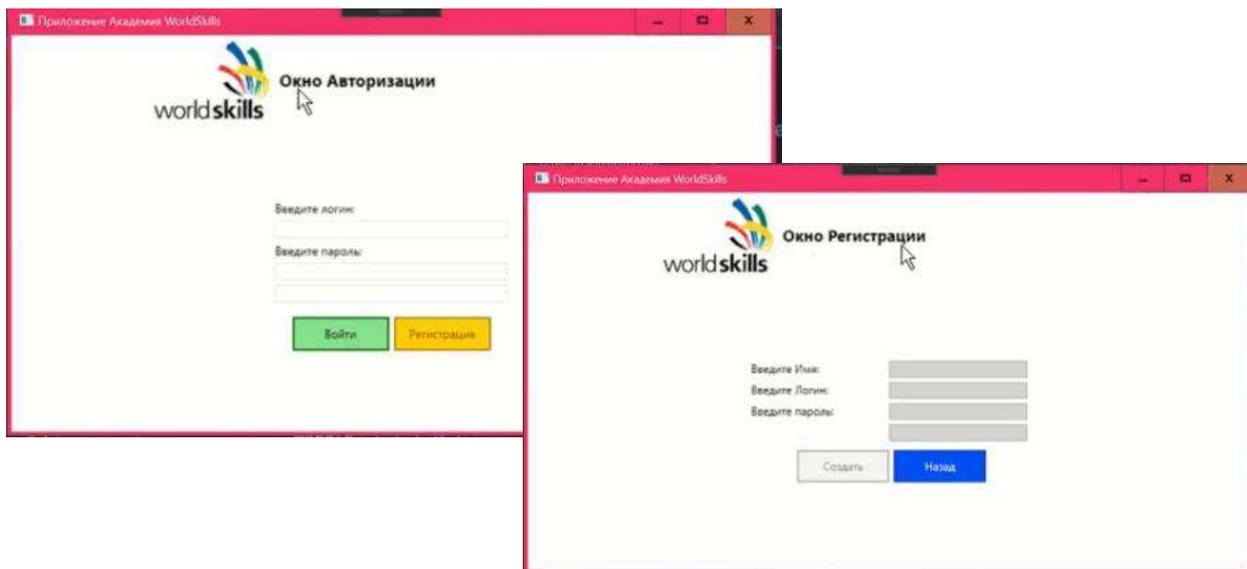
**Запустите проект, зарегистрируйте нового ученика, осуществите вход под его аккаунтом!**

3. Измените интерфейс таким образом, чтобы название проекта отображалось в строке заголовка главного окна, а в верхней части окна каждый раз располагалось название открытой страницы (**Окно авторизации, Окно регистрации**).

Изменения для этого вносим в разметку: добавляем элемент **TextBlock** и в его свойстве **Text** указываем источник текста – **Title** открытой страницы (содержимого фрейма).

```
Text="{Binding ElementName=FrmMain,  
Path=Content.Title}"
```

**Запустите проект, убедитесь в отображении заголовков открытых страниц!**



### Страница администратора

Добавьте в проект папку **PageAdmin** и создайте в ней страницу **PageMenuAdmin** (высота 350).

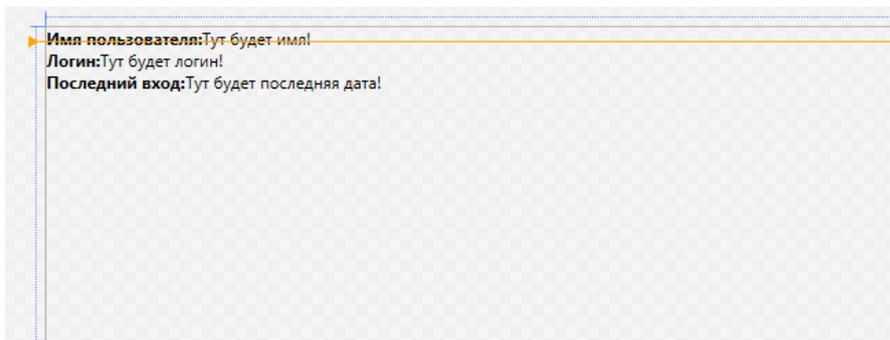
Организируйте разметку страницы в соответствии с рисунком. Используйте контейнер **StackPanel**, расположите его по центру страницы. Организируйте переход к этой странице после авторизации.



### Страница ученика (студента)

Добавьте в проект папку **PageStudent** и создайте в ней страницу **PageAccountStudent** (высота 350).

Организируйте разметку страницы в соответствии с рисунком. Используйте контейнеры **StackPanel**, Организируйте переход к этой странице после авторизации.



**Запустите проект, убедитесь, что переходы к страницам осуществляются и заголовки страниц выводятся в окне!**



**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

**Практическое занятие 23      Введение в RationalRose**

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** изучить принципы работы с Case-пакетом Rational Rose

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Microsoft Visio,

**Практическая часть:**

1. Выберите пункт Tools > Options и откройте вкладку Toolbars.

Чтобы сделать видимой или невидимой стандартную панель инструментов, пометьте (или снимите пометку) контрольный переключатель show standard ToolBar (или show Diagram ToolBar).

2. Увеличьте размер кнопок на панели инструментов:

1. Щелкните правой кнопкой мыши по требуемой панели.

2. Выберите во всплывающем меню пункт Use Large Buttons (Использовать большие кнопки), вернитесь к нормальному размеру кнопок.

3. Настройте панель инструментов:

1. Щелкните правой кнопкой мыши по панели диаграммы Main пакета Use Case View.

2. Выберите пункт Customize (настроить) и добавьте несколько кнопок.

Чтобы добавить или удалить кнопки, выберите соответствующую кнопку и затем щелкните мышью по кнопке Add (добавить) или Remove (удалить).

**Контрольные вопросы:**

1. Для чего нужна программа RationalRose?

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;  
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;

- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 24      Создание      диаграммы      вариантов использования

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Целью** – изучить состав, назначение, функциональные возможности и общие правила работы в CASE-средстве Rational Rose на примере построения диаграмм прецедентов

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Rational Rose

**Практическая часть:**

Задание 1. Построить в среде Rational Rose диаграмму прецедентов проекта регистрации учебных курсов. Порядок выполнения:

1.1. Изучить сценарий проекта, определить действующие лица системы и сценарии их поведения.

*Сценарий проекта регистрации учебных курсов*

Сначала каждый преподаватель университета заполняет специальную форму, в которой указывает, какие учебные курсы он намерен вести в следующем семестре. Данные из формы помещаются в университетский компьютер работником регистратуры.

После этого из полученных данных формируется каталог курсов, который раздается студентам. Студенты выбирают из каталога те курсы, на которых они собираются учиться, и подают заявки на обучение в регистратуру. Все эти данные также попадают в компьютер, где происходит их обработка и формирование списков курсов и студентов. В задачи создаваемой системы входит, в частности, такое комплектование учебных курсов, чтобы каждый курс посещало бы от трех до десяти студентов. Если на какой-то курс не набирается трех студентов, он отменяется.

После формирования курсов преподаватели получают списки студентов, которых им предстоит обучать, а каждый студент получает подтверждение о зачислении на курс и счет на оплату.

1.2. Определить действующие лица системы.

Согласно сценарию действующих лиц в создаваемой системе четыре: преподаватель, студент, регистратор и биллинговая программа – система оплаты. Первые три выбраны действующими лицами, поскольку они активно взаимодействуют с создаваемой системой. Биллинговая же программа чаще всего является отдельным программным продуктом, а в нашем случае она получает информацию для своей работы от создаваемой курсовой системы, поэтому может считаться самостоятельным действующим лицом.

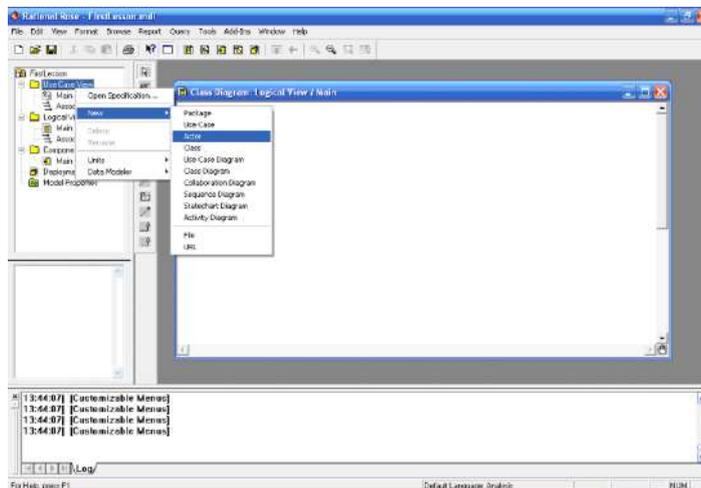
1.3. Определить сценарии поведения действующих лиц. Каждый сценарий описывает некоторое требование к функциям системы:

- Выбор курсов для преподавателя;
- Запрос расписания курсов;
- Регистрация на курсы;
- Создание каталогов ресурсов;
- Хранение информации о курсах;
- Хранение информации о преподавателях;
- Хранение информации о студентах.

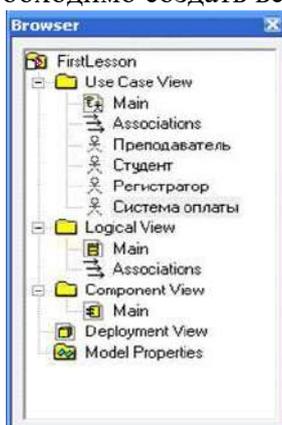
1.4. Запустить программу Rational Rose. Ярлык программы: . При запуске в окне Create New Model нажать кнопку Cancel.

1.5. В программе Rational Rose создать пустой проект. Для этого:

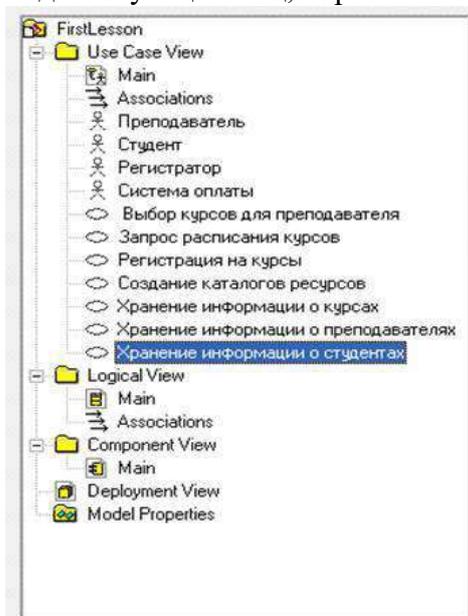
1.5.1. Переключиться на папку Use Case View – для работы с диаграммами прецедентов (Use case) и открыть контекстное меню нажатием правой кнопки мыши. Если теперь выбрать пункт New → Actor (рис. 4.4), то вы получите действующее лицо, которому следует дать имя Преподаватель.



Далее аналогичным образом необходимо создать всех действующих лиц системы



1.5.2. С помощью контекстного меню папки Use Case View и команды New → Use Case создать сценарий поведения действующих лиц, перечисленные в п. 3.



1.6. Построить диаграмму сценариев поведения (прецедентов). Для этого:

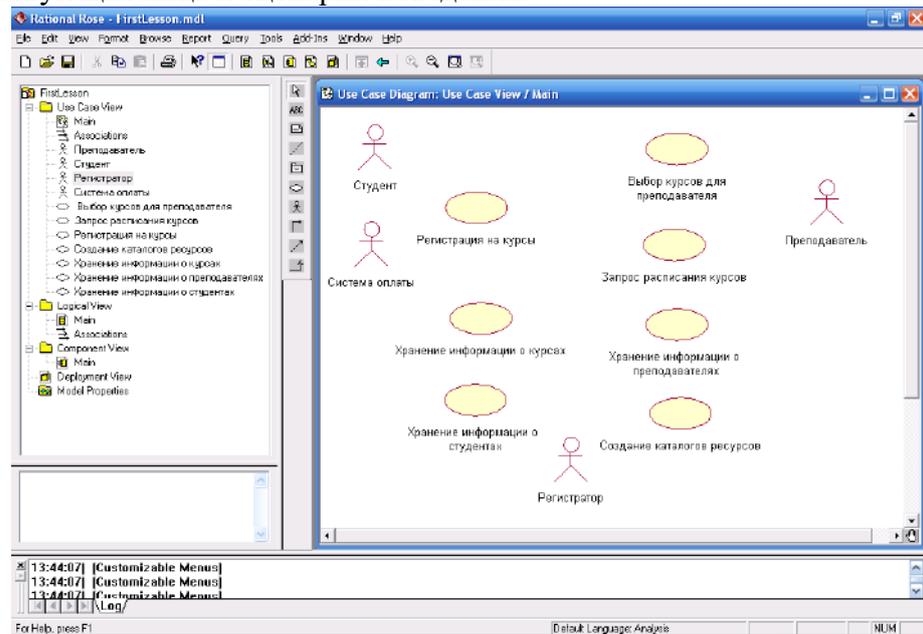
1.6.1. Открыть окно построения диаграммы прецедентов двойным щелчком на пиктограмме Main из папки Use Case View.

1.6.2. Из окна браузера перетащить в окно построения диаграммы прецедентов все созданные действующие лица и сценарии поведения.

1.6. Построить диаграмму сценариев поведения (прецедентов). Для этого:

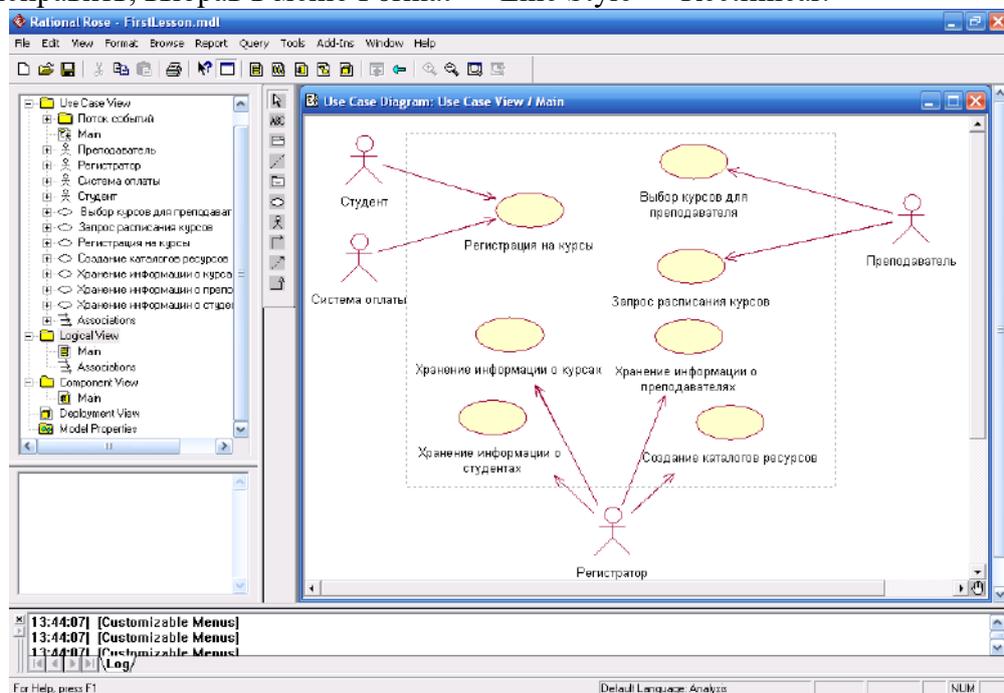
1.6.1. Открыть окно построения диаграммы прецедентов двойным щелчком на пиктограмме Main из папки Use Case View.

1.6.2. Из окна браузера перетащить в окно построения диаграммы прецедентов все созданные действующие лица и сценарии поведения.



1.6.3. Установить взаимосвязи между действующими лицами и сценариями поведения. Для этого на специальной панели инструментов выбрать тип связи Однонаправленная ассоциация (Unidirectional Association), после чего протянуть линию между действующим лицом и сценарием поведения. В результате на диаграмме возникнет стрелка.

1.6.4. Создать границы между актерами и прецедентами, воспользовавшись пунктом меню Tools → Create → Note Anchor или пиктограммой специальной панели инструментов Anchor Note to Item. Выбрав инструмент необходимо щелкнуть один раз на середине одной из стрелок отношения, далее во всех углах интерфейса и завершить границу в месте начала ее рисования. Если прямоугольник получился не очень ровный то это можно исправить, выбрав в меню Format → Line Style → Rectilinear.



### Контрольные вопросы:

1. Что такое UML?
2. Что называют диаграммами UML?
3. Какие типы UML диаграмм Вам известны?
4. Назовите основные элементы окна программы Rational Rose.

5. Какое назначение диаграммы прецедентов?

6. Что такое прецедент в UML?

7. Что представляет собой актер в UML?

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 25      Анализ системы

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Цель** Выполнить архитектурный анализ проектируемой системы

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Rational Rose

**Практическая часть:**

**Упражнение 1. Создание структуры модели и классов анализа в соответствии с требованиями архитектурного анализа**

**Создание пакетов и диаграммы Traceabilities:**

1. Щелкните правой кнопкой мыши по логическому представлению браузера.
2. Выберите пункт New > Package в открывшемся меню.
3. Назовите новый пакет Design Model.
4. Щелкните правой кнопкой мыши по пакету Design Model и создайте аналогичным образом пакеты Use-Case Realizations, Use-Case Realization - Close Registration, Use-Case Realization - Login и Use-Case Realization - Register for Courses.
5. В каждом из пакетов типа Use-Case Realization создайте соответствующие кооперации Close Registration, Login и Register for Courses (каждая кооперация представляет собой вариант использования со стереотипом «use-case realization», который задается в спецификации варианта использования).

Структура логического представления браузера должна иметь следующий вид (рис)

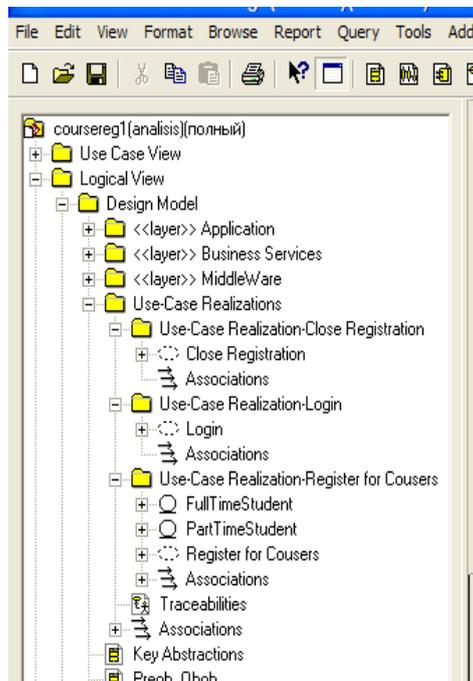


Рис. Структура логического представления браузера

6. Создайте в пакете Use-Case Realization новую диаграмму вариантов использования с названием Traceabilities и постройте ее в соответствии с рис.10

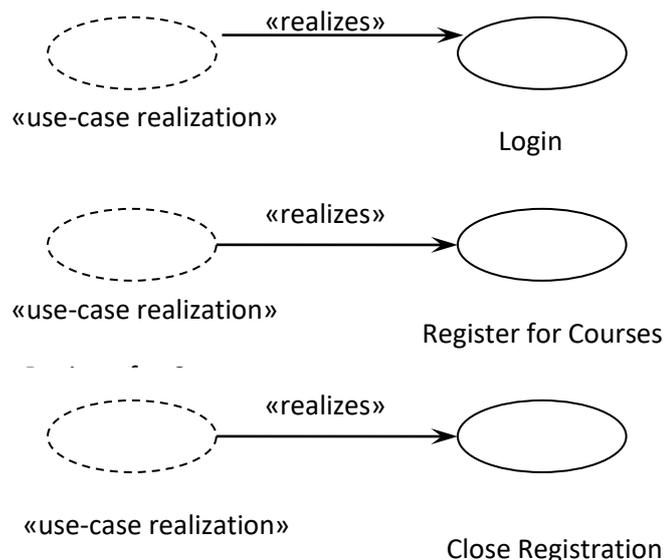


Рис. Диаграмма Traceabilities

Создание классов анализа и соответствующей диаграммы Key Abstractions:

1. Щелкните правой кнопкой мыши по пакету Design Model.
2. Выберите пункт New > Class в открывшемся меню. Новый класс под названием NewClass появится в браузере.
3. Выделите его и введите имя Student.
4. Создайте аналогичным образом классы Professor, Schedule, Course и CourseOffering.
5. Щелкните правой кнопкой мыши по пакету Design Model.
6. Выберите пункт New > Class Diagram в открывшемся меню.
7. Назовите новую диаграмму классов Key Abstractions.
8. Чтобы расположить вновь созданные классы на диаграмме классов, откройте ее и перетащите классы на открытую диаграмму мышью. Диаграмма классов должна выглядеть, как на рис

## **Анализ вариантов использования**

**Идентификация классов, участвующих в реализации потоков события варианта использования.** В потоках событий варианта использования выявляются классы трех типов:

**граничные классы (Boundary)** - служат посредниками при взаимодействии внешних объектов с системой. Как правило, для каждой пары «действующее лицо - вариант использования» определяется один граничный класс. Типы граничных классов: пользовательский интерфейс (обмен информацией с пользователем, без деталей интерфейса - кнопок, списков, окон), системный интерфейс и аппаратный интерфейс (используемые протоколы, без деталей их реализации);

**классы-сущности (Entity)** - представляют собой ключевые абстракции (понятия) разрабатываемой системы. Источники выявления классов-сущностей: ключевые абстракции, созданные в процессе архитектурного анализа, глоссарий, описание потоков события вариантов использования;

**управляющие классы (Control)** - обеспечивают координацию поведения объектов в системе. Могут отсутствовать в некоторых вариантах использования, ограничивающихся простыми манипуляциями с хранимыми данными. Как правило, для каждого варианта использования определяется один управляющий класс. Примеры управляющих классов: менеджер транзакций, координатор ресурсов, обработчик ошибок.

**Упражнение 2. Создание классов, участвующих в реализации варианта использования Register for Courses, и диаграммы классов «View Of Participating Classes» (VOPC)**

1. Щелкните правой кнопкой мыши по пакету Design Model.
2. Выберите пункт New > Class в открывшемся меню. Новый класс под названием NewClass появится в браузере.
3. Выделите его и введите имя RegisterForCoursesForm.
4. Щелкните правой кнопкой мыши по классу RegisterForCoursesForm.
5. Выберите пункт Open Specification в открывшемся меню.
6. В поле стереотипа выберите Boundary и нажмите на кнопку ОК.
7. Создайте аналогичным образом классы CourseCatalogSystem со стереотипом Boundary и RegistrationController со стереотипом Control.
8. Назначьте классам Schedule, CourseOffering и Student стереотип Entity.
9. Щелкните правой кнопкой мыши по кооперации Register for Courses в пакете Use-Case Realization - Register for Courses.
10. Выберите пункт New > Class Diagram в открывшемся меню.
11. Назовите новую диаграмму классов VOPC (classes only).
12. Откройте ее и перетащите классы на открытую диаграмму в соответствии с рис.11.

**Распределение поведения, реализуемого вариантам использования, между классами.** Реализуется с помощью диаграмм взаимодействия (диаграмм последовательности и кооперативных диаграмм). В первую очередь строится диаграмма (одна или более), описывающая основной поток событий и его подчиненные потоки. Для каждого альтернативного потока событий строится отдельная диаграмма. Примеры:

1. обработка ошибок;
2. контроль времени выполнения;
3. обработка неправильных вводимых данных.

Нецелесообразно описывать тривиальные потоки событий (например, в потоке участвует только один объект).

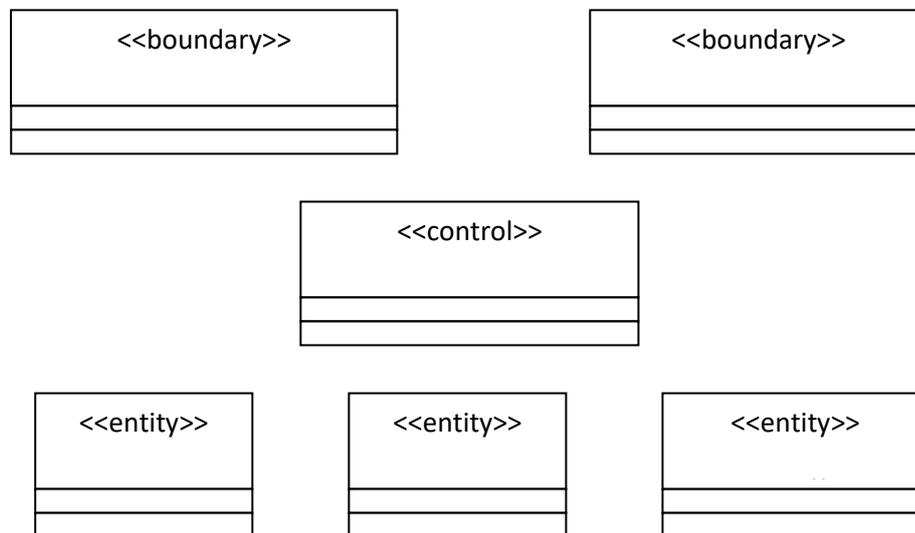


Рис. Диаграмма классов VOPC (classes only)

**Задание для самостоятельной работы:**

Выполните архитектурный анализ для системы управления банкоматом.

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

**Практическое занятие 26      Создание диаграмм взаимодействия**

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Цель** создание диаграмм взаимодействия в RationalRose

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Rational Rose

**Практическая часть:**

Нотация языка UML для объектов, связей и сообщений на диаграмме взаимодействий показана на рис.

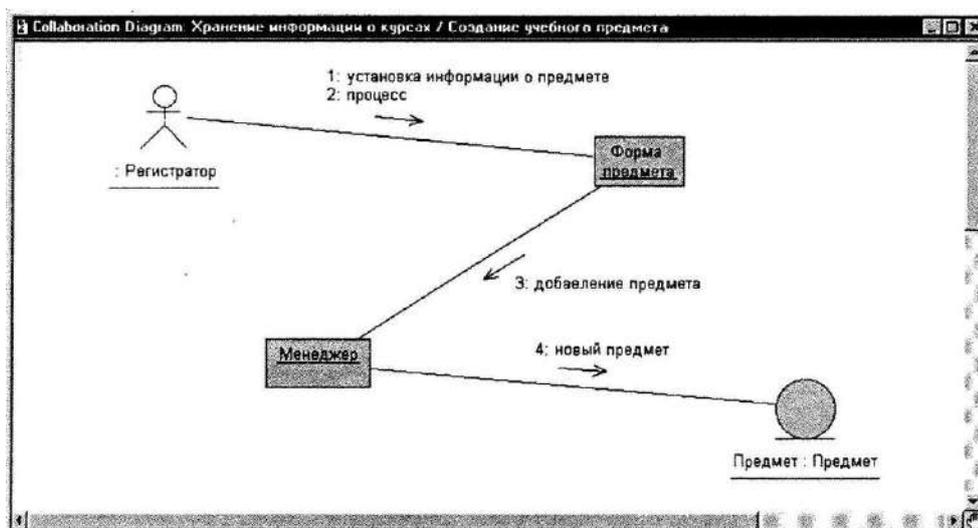
Последовательность создания диаграмм взаимодействий из диаграмм последовательности действий в программе Rational Rose:

1. Дважды щелкните по диаграмме последовательности действий в списке браузера, чтобы открыть диаграмму.
2. Выберите команду меню **Browse => Create collaboration diagram** (Просмотр => Создать диаграмму взаимодействий) или нажмите клавишу **F5**.
3. Расположите объекты и сообщения на диаграмме нужным образом.



Диаграмма взаимодействий показана на рис.

Можно сначала создать диаграмму последовательности действий. В этом случае диаграмма последовательности действий может быть получена из нее. Для этого необходимо выбрать команду меню **Browse => Create Sequence Diagram** (Просмотр => Создать диаграмму последовательности действий) или нажать клавишу **F5**.



#### Контрольные вопросы:

1. Для чего используются диаграммы взаимодействия?

#### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;

- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 27 Создание диаграмм классов

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

**уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Цель** создание диаграмм классов в RationalRose

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Rational Rose

**Практическая часть:**

Диаграмма *классов* является основным логическим представлением разрабатываемой модели и содержит детальную информацию о внутреннем устройстве объектно-ориентированной программной системы или, используя современную терминологию, об архитектуре программной системы. Активизировать рабочее окно диаграммы *классов* можно несколькими способами:

1. окно диаграммы *классов* появляется по умолчанию в рабочем окне диаграммы после создания нового проекта (Рис.);
2. щелкнуть на кнопке с изображением диаграммы *классов* на стандартной панели инструментов;
3. раскрыть логическое представление (Logical View) в браузере проекта и дважды щелкнуть на пиктограмме **Main** (Главная);
4. выполнить операцию главного меню: **Browse**→**Class Diagram** (Обзор→Диаграмма *классов*).

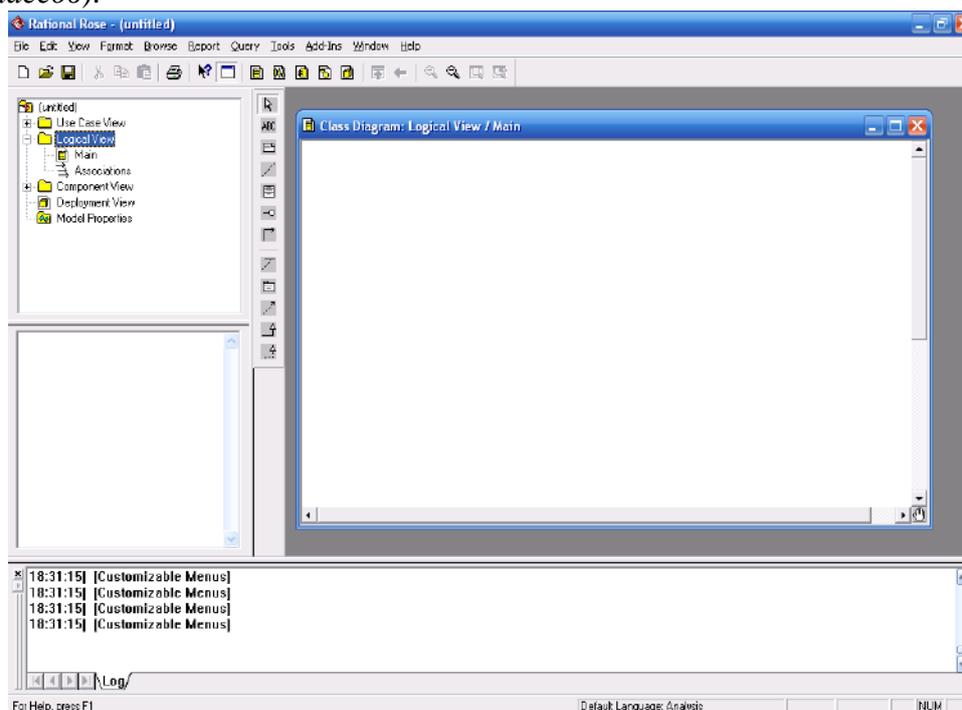


Рис. 1 Рабочее окно диаграммы классов

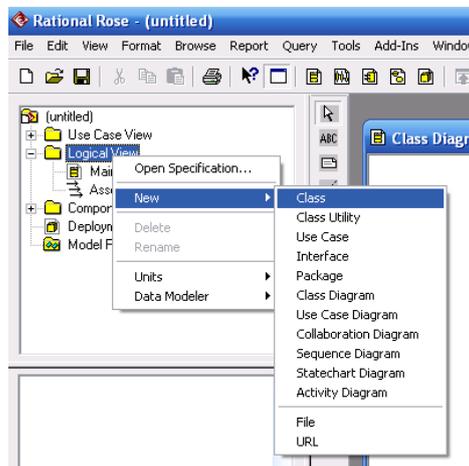
При этом появляется новое окно с чистым рабочим листом диаграммы *классов* и специальная панель инструментов, содержащая кнопки с изображением графических примитивов, необходимых для разработки диаграммы *классов* (Табл. 1.1).

Назначение отдельных кнопок панели можно узнать также из всплывающих подсказок.

Таблица 1.1 Назначение кнопок специальной панели инструментов для диаграммы классов		
Графическое изображение	Всплывающая подсказка	Назначение кнопки
	Selection Tool	Превращает изображение курсора в форму стрелки для последующего выделения элементов на диаграмме
	Text Box	Добавляет на диаграмму текстовую область
	Note	Добавляет на диаграмму примечание
	Anchor Note to Item	Добавляет на диаграмму связь примечания с соответствующим графическим элементом диаграммы
	Class	Добавляет на диаграмму <i>класс</i>
	Interface	Добавляет на диаграмму <i>интерфейс</i>
	Unidirectional Association	Добавляет на диаграмму направленную <i>ассоциацию</i>
	Association Class	Добавляет на диаграмму <i>ассоциацию класс</i>
	Package	Добавляет на диаграмму пакет
	Dependency or Instantiates	Добавляет на диаграмму отношение зависимости
	Generalization	Добавляет на диаграмму отношение обобщения
	Realize	Добавляет на диаграмму отношение реализации

На специальной панели инструментов по умолчанию присутствует только часть пиктограмм элементов, которые могут быть использованы для построения диаграммы *классов*. Добавить кнопки с пиктограммами других графических элементов или удалить ненужные кнопки можно с помощью настройки специальной панели инструментов. Соответствующее диалоговое окно настройки специальной панели инструментов для диаграммы *классов* можно вызвать аналогично другим панелям с помощью операции контекстного меню **Customize** (Настройка) при позиционировании курсора на специальной панели инструментов.

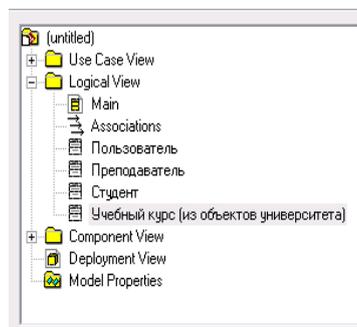
Для добавления *класса* на диаграмму *классов* нужно с помощью левой кнопки мыши нажать кнопку с изображением пиктограммы *класса* на специальной панели инструментов, отпустить левую кнопку мыши и щелкнуть левой кнопкой мыши на свободном месте рабочего листа диаграммы. На диаграмме появится изображение *класса* с маркерами изменения его геометрических размеров и предложенным средой именем по умолчанию NewClass. Или создать класс через пункт всплывающего меню New->Class (Рис.2), а потом “перетащить” созданный класс из окна браузера проекта в область окна диаграммы классов (Рис. 4).



**Рис.2** Создание нового класса

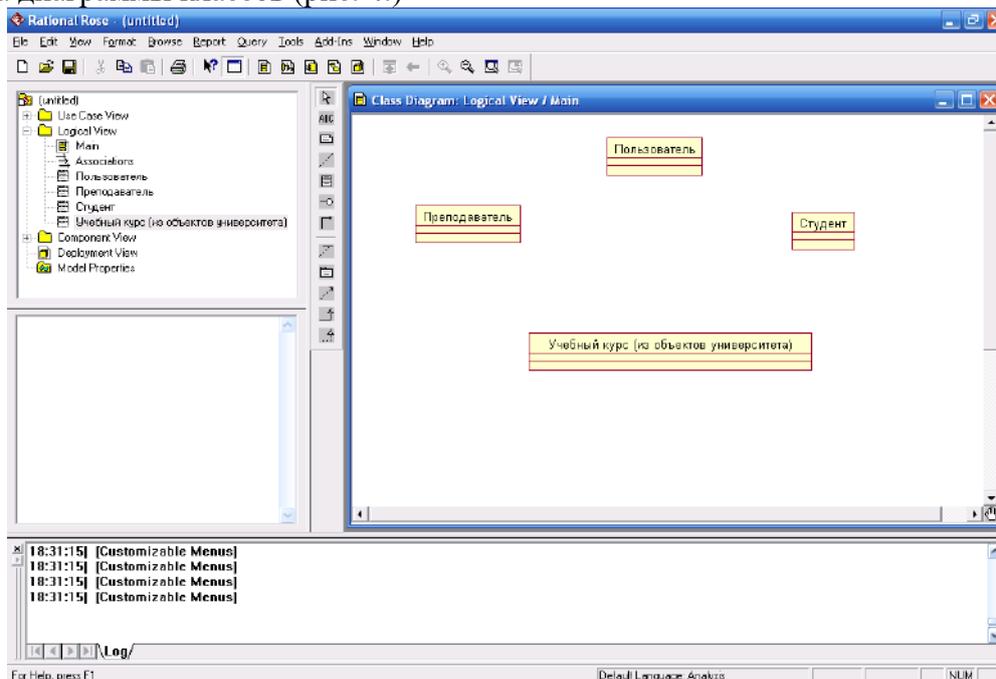
В нашем случае (АИС регистрации учебных курсов) нужно создать 4 класса (Рис.3):

- Пользователь;
- Преподаватель;
- Студент;
- Учебный курс (из объектов университета).



**Рис. 3** Браузер проектов – отображает все заданные классы

“Перетащим” мышкой только что созданные классы из окна браузера проекта в область окна диаграммы классов (рис. 4.)



**Рис. 4** Окно диаграммы классов – отображает все созданные классы

Из всех графических элементов среды IBM Rational Rose класс обладает максимальным набором свойств, главными из которых являются его *атрибуты* и *операции*. Кстати, именно диаграмма классов используется в среде IBM Rational Rose для генерации программного кода, но об этом будет рассказано на другом лабораторном занятии.

На этом лабораторном занятии мы рассмотрим только создание и редактирование атрибутов классов.

Давайте добавим атрибуты для класса “Пользователь”:

- Имя;
- Ид.номер.

Добавить *атрибут* к созданному ранее классу можно одним из следующих способов:

- С помощью *операции* контекстного меню New Attribute (Новый *атрибут*) для класса, выделенного на диаграмме классов. В этом случае активизируется курсор ввода текста в области графического изображения класса на диаграмме (Рис. 5).

- С помощью *операции* контекстного меню: New→Attribute (Новый→*Атрибут*) для класса, выделенного в браузере проекта. В этом случае активизируется курсор ввода текста в области иерархического представления класса в браузере проекта под именем соответствующего класса.

- С помощью *операции* контекстного меню Insert (Вставить), вызванного при позиционировании курсора в области открытой вкладки *атрибутов* в диалоговом окне свойств Class Specification соответствующего класса (Операция контекстного меню OpenSpecification или двойной «щелчок» мышью на изображении класса) (Рис. 6).

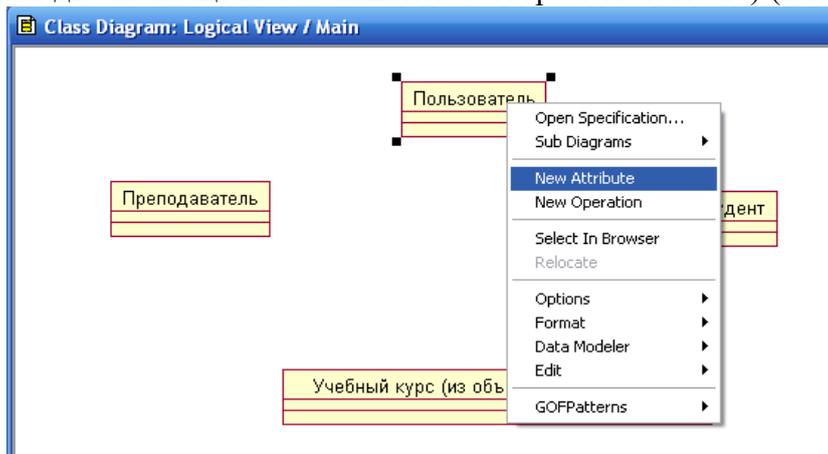


Рис. 5 Окно диаграммы классов – добавляем атрибут к классу

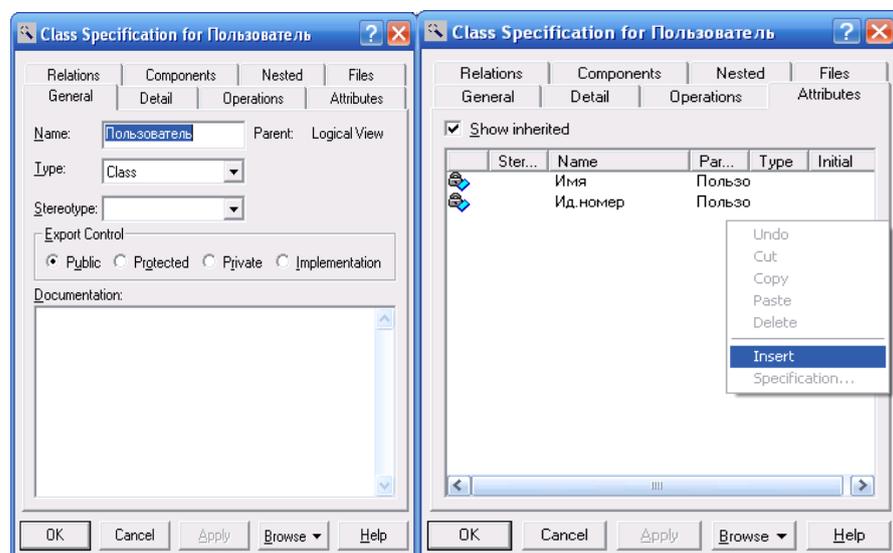
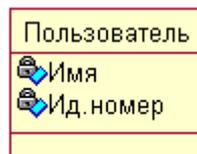


Рис. 6 Диалоговое окно свойств класса – нужно перейти к вкладке Attributes



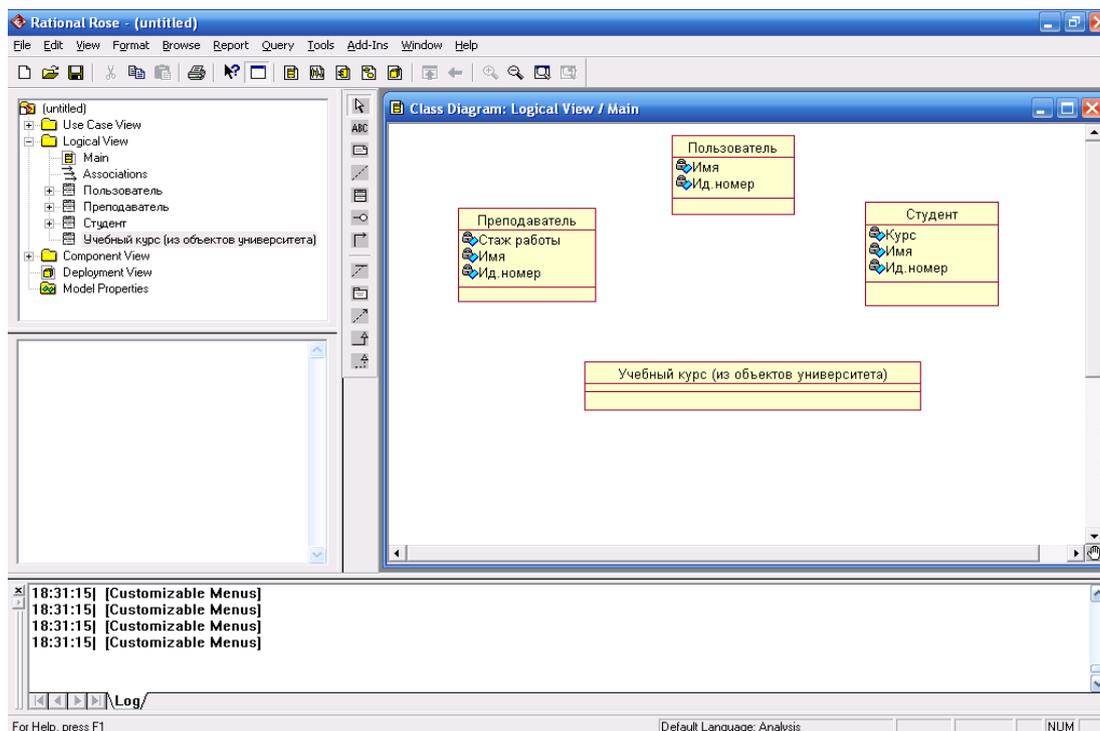
**Рис. 7** Класс “Пользователь”

После добавления *атрибута* к классу по умолчанию ему присваивается имя *name* и *квантор видимости* (Рис. 7). Нужно запомнить, что имена *атрибутов* и *операций* классов должны начинаться со *строчной* буквы.

Видимость *атрибутов* на диаграмме классов изображается в форме специальных пиктограмм или украшений. Используемые пиктограммы видимости изображаются перед именем соответствующего *атрибута* и имеют следующий смысл (Табл.2).

Таблица 2 Пиктограммы видимости атрибутов классов		
Графическое изображение	Текстовый аналог	Назначение пиктограммы
	Public	Общедоступный или открытый. В нотации языка UML такому атрибуту соответствует знак «+»
	Protected	Защищенный. В нотации языка UML такому атрибуту соответствует знак «#»
	Private	Закрытый. В нотации языка UML такому атрибуту соответствует знак «-»
	Implementation	Реализация. В нотации языка UML такому атрибуту соответствует знак «~»

Теперь добавим атрибуты всем классам, изображенным в окне диаграммы классов (Рис.8).

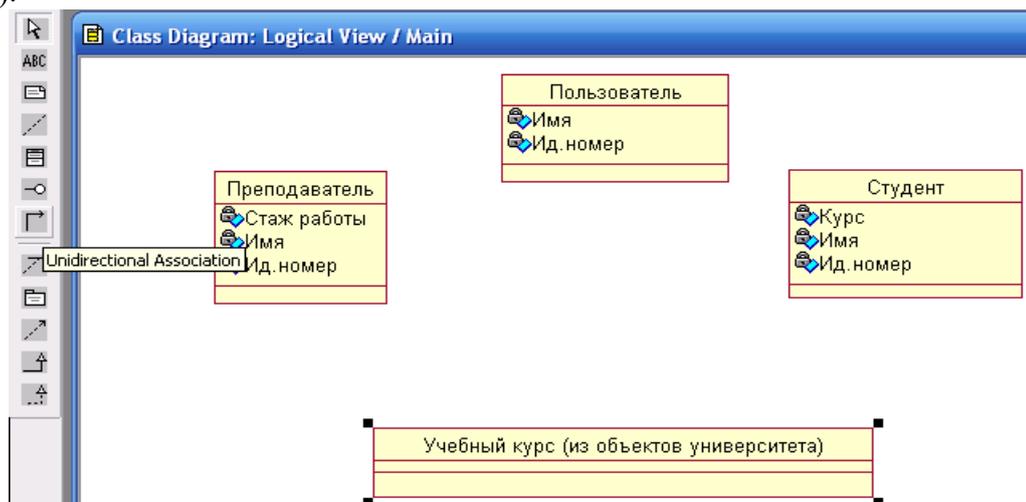


**Рис. 8** Окно диаграммы классов – отображает все созданные классы и их атрибуты  
Следующий этап – это установление взаимосвязей между классами нашей диаграммы.

В этой лабораторной работе мы будем использовать два типа отношений (взаимосвязей):

- ассоциацию;
- обобщение.

Добавление на диаграмму *ассоциации* между двумя классами выполняется с помощью . Построим взаимосвязь между классами “Преподаватель” и “Учебный курс” (Рис.9).



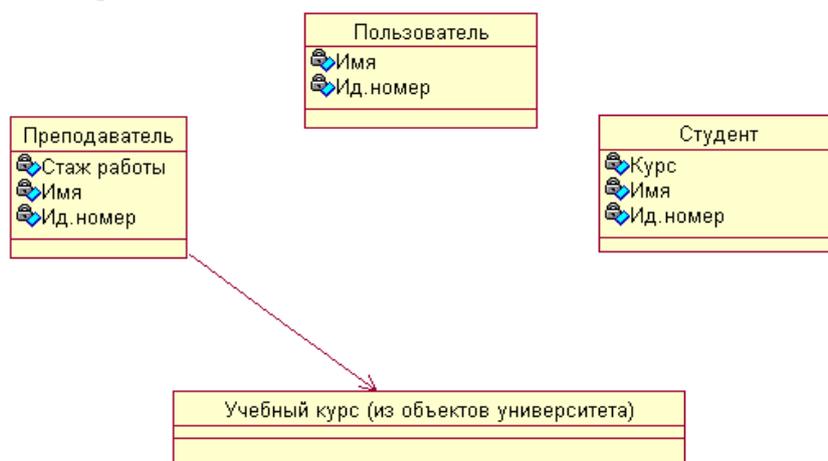
**Рис. 9** Окно диаграммы классов – отображает элемент специальной панели инструментов- “Ассоциация”

Но тогда ассоциация будет однонаправленной (Рис. 9), а нам нужно будет получить двунаправленное отношение “1” к “0..5”, т.е. один преподаватель может вести от 0 до 5 учебных курсов.

**Ассоциация** – это структурное двунаправленное отношение, описывающее совокупность взаимоотношений между объектами. По сути дела ассоциация является сверткой бинарных отношений между объектами. Эту свертку может мысленно выполнить в своем сознании специалист (например, программист), который видит пиктограмму ассоциации на UML диаграмме.

Пометка единица (1) на левом конце линии ассоциации означает, что в двунаправленном отношении, наряду с многими, например, работниками участвует один работодатель. Единица и звездочка на правом конце линии означает "единица или больше" (1..\*). Если один конец линии ассоциации помечен единицей (1), то пометка на другом конце линий называется кратностью ассоциации.

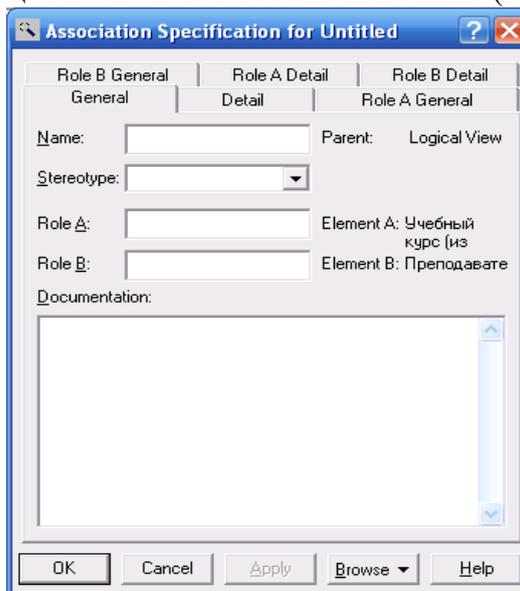
На линии ассоциации можно также задать кратность равную единице (1), можно указать диапазон кратности: ноль или единица (0..1), много (0..\*). Разрешается также указывать кратность определенным числом (например 5). С помощью списков можно задавать и более сложные кратности. Например, список 0..1, 3..4, 6..\* означает "любое число объектов кроме 2 и 5".



**Рис. 10** Окно диаграммы классов – отображает однонаправленную взаимосвязь (отношение) двух классов “Преподаватель” и “Учебный курс”

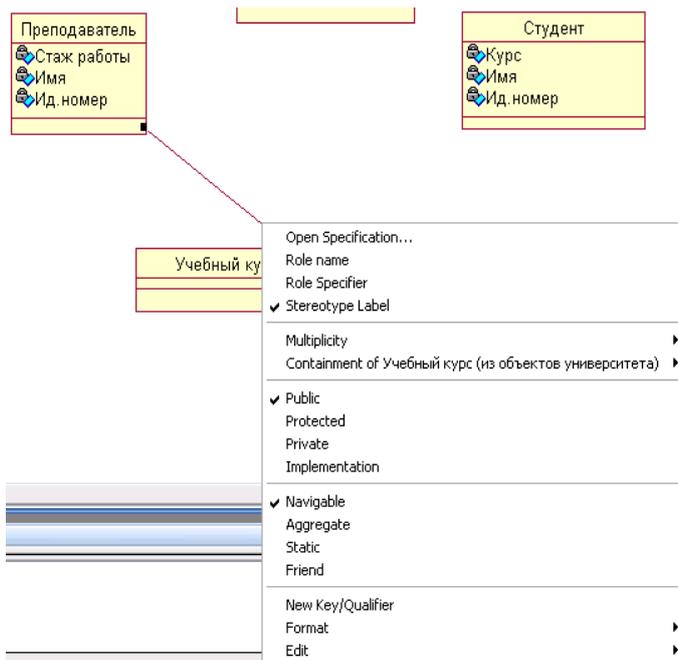
Изменим свойства для данной *ассоциации*, предложенные средой по умолчанию. Это можно выполнить с помощью окна спецификации свойств *ассоциации*. Доступ к

диалоговому окну спецификации свойств *ассоциации* Association Specification можно получить после выделения линии *ассоциации* на диаграмме классов (Рис. 11) или в браузере проекта и двойного щелчка на ней левой кнопки мыши (Рис. 12) .



**Рис. 11** Диалоговое окно спецификации свойств *ассоциации*

Первое, что мы делаем — убираем направленность отношения: во всплывающем меню ассоциации («Клик» правой кнопкой мыши) убираем галочку с пункта «Navigate» (Рис.12).



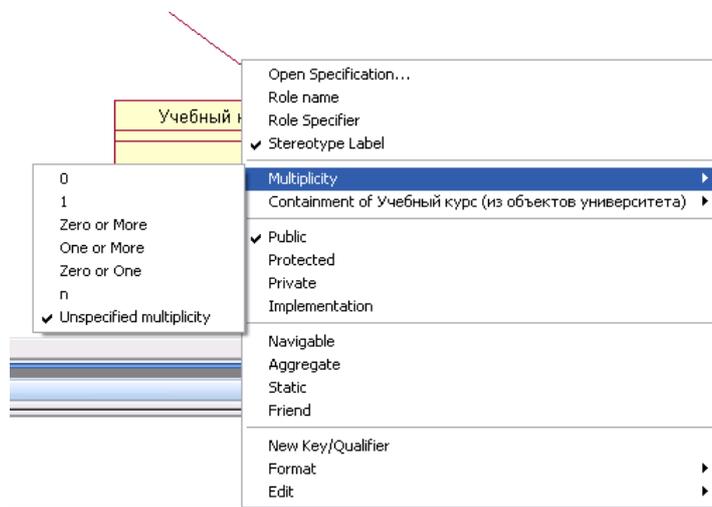
**Рис. 12** Всплывающее меню свойств *ассоциации*

Вторым шагом нужно указать кратности для ассоциации: со стороны “Пользователя” 1, а со стороны “Учебного курса” диапазон кратности “0..5” (от 0 до 5).

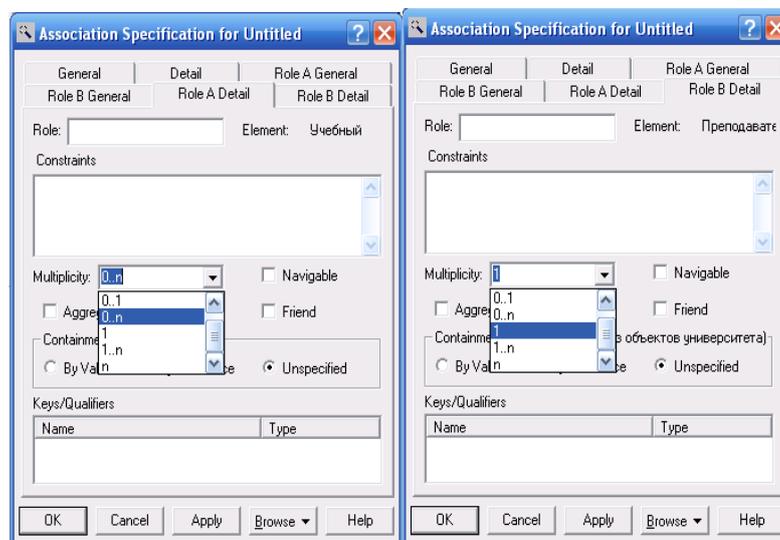
Это можно сделать двумя способами:

- с помощью всплывающего меню ассоциации (см.выше) – выбрать в пункте Multiplicity (дословно “Сложность отношения”) нужную кратность (Рис. 13);
- в диалоговом окне свойств ассоциации (Open Specification - Рис.14) на вкладках RoleA Detail и RoleB Detail выбрать кратности для обеих сторон ассоциации из выпадающего списка пункта Multiplicity.

Воспользуемся вторым способом (Рис. 15). У стороны “Учебного курса” в меню Multiplicity выберем пункт “один ко многим” (1..\* или 1..n) и вместо n поставим “5”.

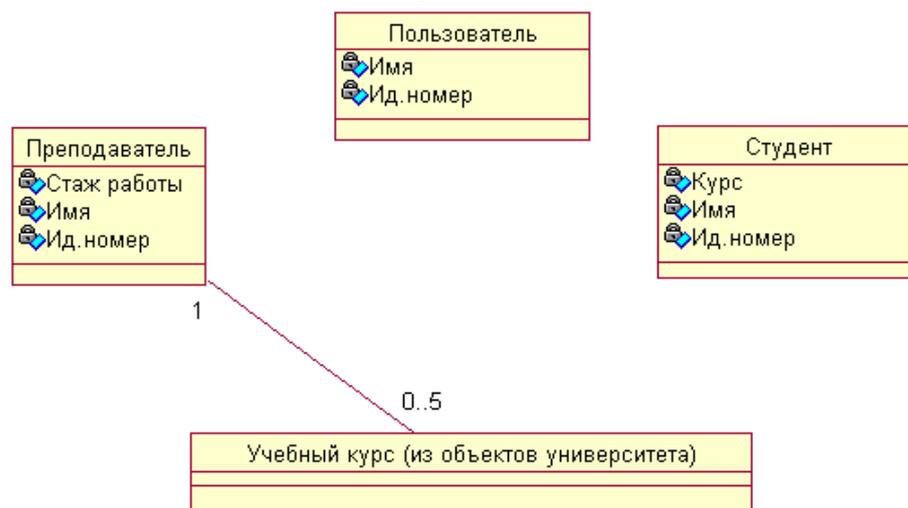


**Рис. 13** Всплывающее меню свойств ассоциации



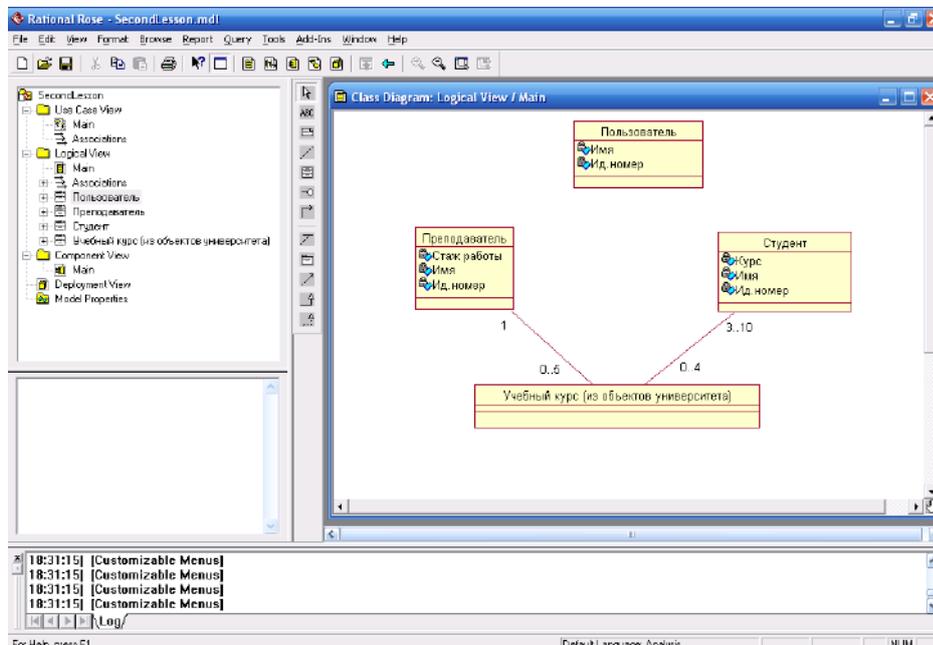
**Рис. 14** Диалоговое окно свойств ассоциации

Полученная взаимосвязь (отношение) представлено на Рис.15.



**Рис. 15** Окно диаграммы классов – отображает взаимосвязь (отношение) двух классов “Преподаватель” и “Учебный курс”

Создадим аналогично взаимосвязь между классами “Студент” и “Учебный курс” (Рис.16)



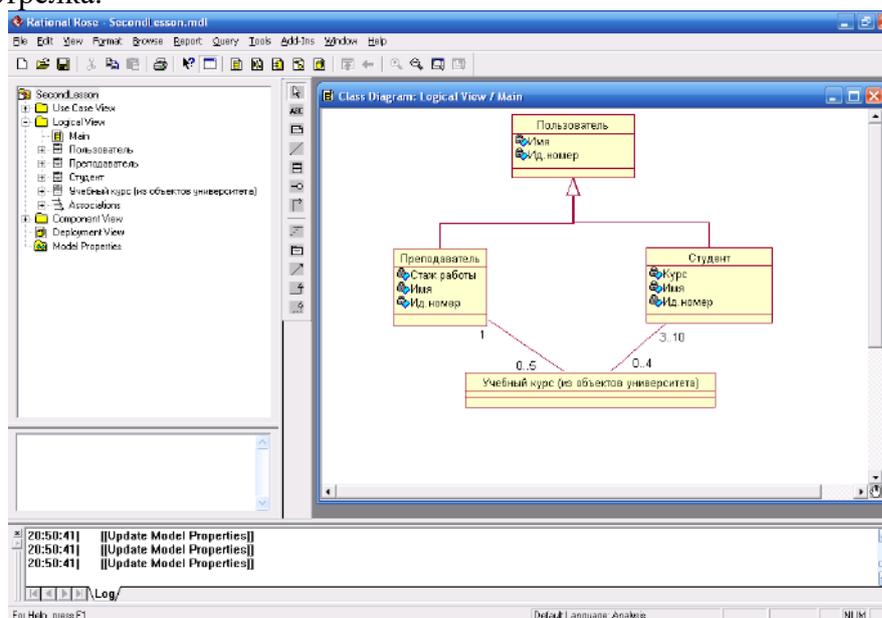
**Рис. 16** Окно диаграммы классов – отображает взаимосвязь (отношение) двух классов “Студент” и “Учебный курс”

Заключительным этапом построения нашей диаграммы классов будет создание взаимосвязи типа Generalization) между классами “Пользователя”, “Преподавателя” и “Студента”.

Обобщение - это однонаправленное отношение, называемое "потомок/прародитель", в котором объект "потомок" может быть подставлен вместо объекта прародителя (родителя или предка). Потомок наследует структуру и поведение своего родителя. Стрелка всегда указывает на родителя.

В данном случае родитель – это “Пользователь”.

Добавление на диаграмму отношения обобщения между двумя классами выполняется следующим образом. На специальной панели инструментов необходимо нажать кнопку с изображением пиктограммы обобщения  и отпустить левую кнопку мыши. Далее на диаграмме классов надо выделить первый элемент обобщения или потомок, от которого исходит стрелка, и, не отпуская нажатую левую кнопку мыши, переместить ее указатель ко второму элементу отношения или предку, к которому направлена стрелка.



**Рис. 17** Окно диаграммы классов – Фрагмент простой концептуальной схемы базы данных компьютерной системы университета “Истерн”

## Контрольные вопросы:

1. Для чего используются диаграммы классов?

### Критерии оценивания:

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## Практическое занятие 28      Проектирование системы

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

### уметь:

- владеть основными методологиями процессов разработки программного обеспечения;

**Цель** проектирование системы в RationalRose

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение: Rational Rose

### Практическая часть:

Спроектировать для выбранной предметной области не менее трех диаграмм в RationalRose: 1. Автострахование.

2. Агентство по сдаче автомобилей в аренду.

3. Аренда коньков, роликов, велосипедов, лыж.

4. Аэропорт – пассажирское расписание и перевозки.

5. Банковская система вкладов (физических и юридических лиц) .

6. Банковская система кредитования (физических и юридических лиц).

7. Биллинг сотовой компании.

8. Ветеринарная лечебница.

9. Клуб обучения танцам.

10. Магазин косметики.

11. Машиностроительное предприятие: система по разработке и модификации изделий (ведение архива, стандартов и пр.).

12. Нефтеперерабатывающая компания.

13. Парикмахерская.

14. Поставка вин.

15. Приемная комиссия ВУЗа.

16. Производство мебели (прием индивидуальных и типовых заказов и изготовление).

17. Рекламное агентство.

18. Риэлторская компания: аренда; продажа первичного и вторичного жилья.
19. Санаторий.
20. Система управления проектом для IT-компании.
21. Складская логистика.
22. Спа-салон (услуги, обслуживающий персонал и пр.).
23. Страховая компания.
24. Такси.
25. Транспортная логистика.
26. Туристическое агентство (путешествия за рубеж).
27. Туристическое агентство (путешествия по России).
28. Учет оборудования на крупном промышленном предприятии.
29. Филармония.
30. Электронный проездной.

#### **Контрольные вопросы:**

1. Какое назначение программного средства Rational Rose?
2. Что такое CASE-средство?
3. Что такое UML? 4.
- Что называют диаграммами UML?
5. Какие типы UML диаграмм Вам известны?
6. Какое назначение диаграммы прецедентов?
7. Что такое прецедент в UML?
8. Что представляет собой актер в UML?
9. Какое назначение диаграммы классов? 1

#### **Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## **Практическое занятие 29      Работа с Kanban**

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся должен:

#### **уметь:**

- владеть основными методологиями процессов разработки программного обеспечения;

**Цель** получить навыки работы с Kanban

**Оборудование:** ноутбук Dell Inspiron 15 5000 Series

**Программное обеспечение:** сервис Trello

## Практическая часть:

В Trello можно работать достаточно гибко и быстро, чтобы управлять проектом любого масштаба от начала и до конца. Узнайте, как создать свою первую доску и приступить к работе.

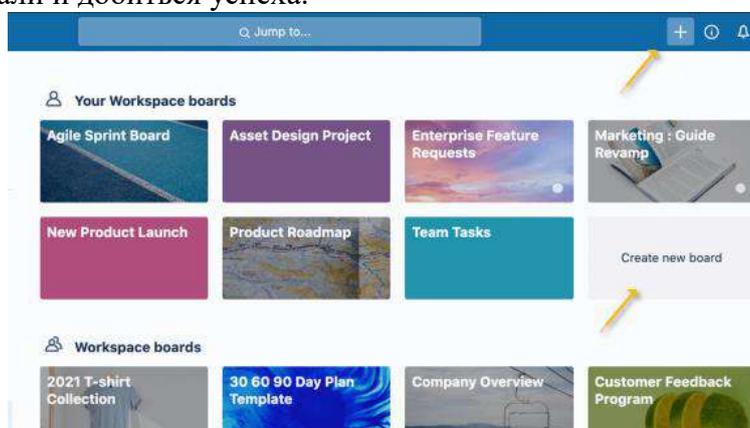
Создайте свой первый проект в Trello

Как и в любом другом проекте, вы будете управлять большим объемом информации, отслеживать статусы заданий и привлекать нужные заинтересованные стороны на каждом этапе работы. Ваша задача — продвигаться вперед в устойчивом темпе, зная, что результат будет поставлен в срок и в рамках бюджета, и сохранять улыбку на лице.

С Trello у вас будет поддержка на каждом этапе этого пути. Теперь, когда вы узнали основное о настройке доски Trello, самое время приступить к работе над первым проектом.

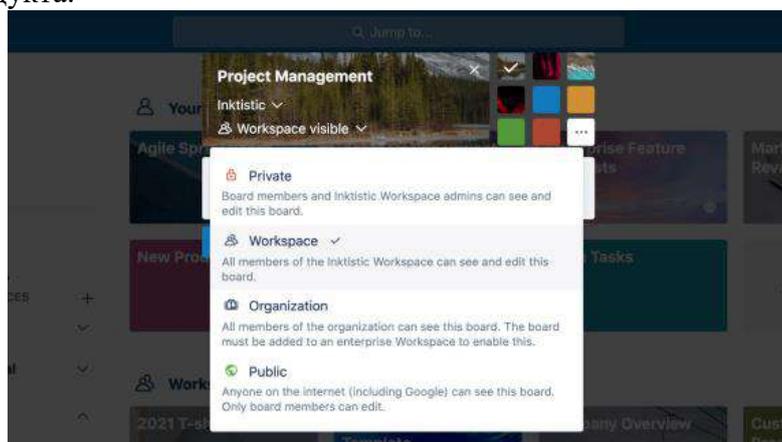
Создание доски

Учитывая количество участников, идей, заданий и дат выполнения, входящих в проект, не так-то просто разработать оптимальную структуру для совместной работы. К счастью, вы можете начать с чистого листа и использовать доску Trello. Она поможет вам организовать все детали и добиться успеха.



На вкладке Workspace (Рабочее пространство) вашей команды нажмите Create new board (Создать новую доску) либо нажмите значок плюса (+) на верхней панели Trello слева от вашего имени. Затем выберите Create Board (Создать доску).

Дайте доске название, чтобы пояснить, над чем вы работаете, от организации мероприятия или управления блогом до отслеживания потенциальных клиентов или запуска нового продукта.



Данные проекта и связанная с ним информация — ваши самые ценные ресурсы. При создании новой доски Trello у вас будет возможность настроить уровень конфиденциальности.

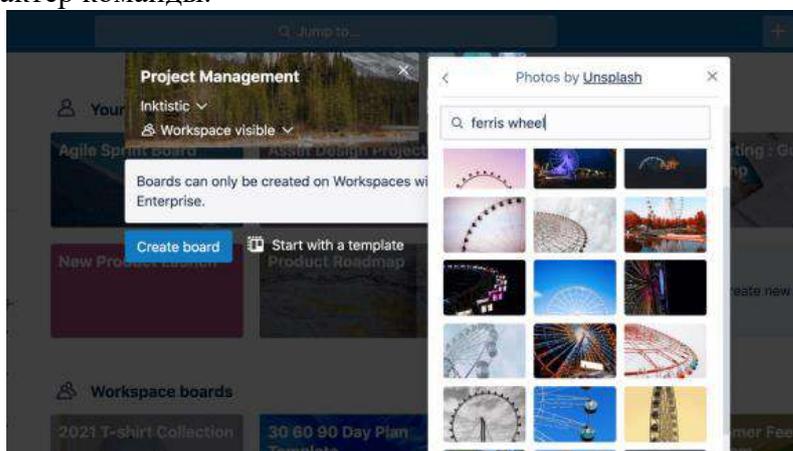
При создании доски используется настройка по умолчанию Workspace Visible (Для рабочего пространства). Это означает, что просматривать, редактировать карточки и присоединяться к ним может любой участник данного рабочего пространства.

Если изменить значение этой настройки на Private (Приватная), доска будет видна только ее непосредственным участникам. Как правило, рекомендуется устанавливать значение Private (Приватная) в качестве значения по умолчанию, если вы работаете над закрытым проектом, доступ к которому нужно ограничить для других участников команды. Этот параметр отлично подходит для личных досок, досок с персональными заданиями или досок, используемых совместно с клиентами.

Public (Публичные) доски могут просматривать все, у кого есть соответствующая ссылка. Публичные доски можно искать с помощью поисковых систем, например Google, а потому любая информация с такой доски доступна всем пользователям в Интернете. Эту настройку можно использовать для открытого проекта или дорожной карты продукта, в отношении которого нужно изучить общественное мнение. Как и другие параметры доски, вы можете точно задать, кто может оставлять комментарии, добавлять или удалять участников, а также редактировать доску и присоединиться к ней.

#### Настройка доски

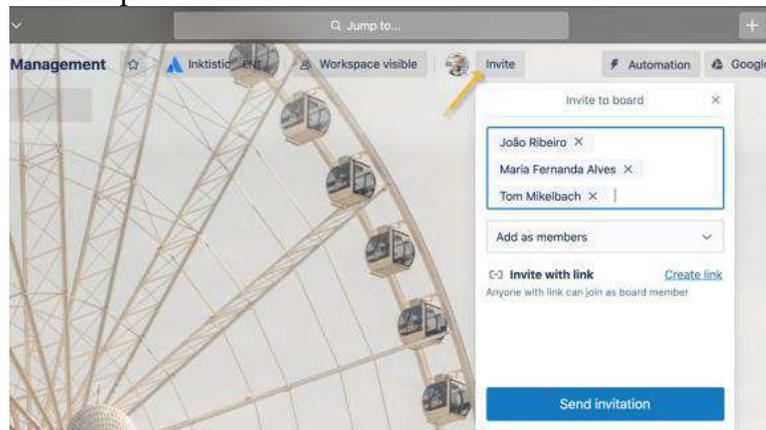
Не бывает двух одинаковых рабочих процессов — так почему бы не придать каждой доске свой уникальный внешний вид? Мы за гибкий и увлекательный формат работы, поэтому в Trello есть возможность настроить фон доски, чтобы отразить уникальный характер команды.



## 1. Начало совместной работы

Совместная работа над проектом идет лучше, если все участники команды остаются в курсе происходящего — и неважно, руководит участник проектом или просто заглядывает в него периодически, чтобы проверить статус выполнения.

Пригласите участников на доску, чтобы назначить им задания и предоставить возможность совместной работы.



2. В верхней части доски нажмите Invite (Пригласить), затем выберите участника команды, которого хотите добавить на доску. Вы также можете найти и пригласить участника по адресу электронной почты или имени.

3. В нижней части меню Invite (Пригласить) доступна ссылка на доску, которой можно без труда поделиться со всеми нужными участниками. Опубликуйте

ссылку в канале Slack или отправьте по электронной почте внешним заинтересованным сторонам, чтобы начать совместную работу. Кроме того, можно пригласить участников, загрузив сканируемый QR-код.

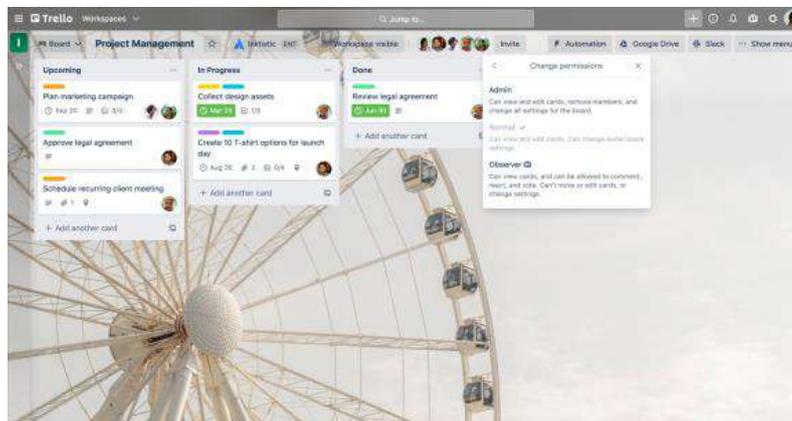
#### 4. Настройка прав доступа участников

Настройте права доступа участников, чтобы четко определить их роли. Права доступа позволяют определить точный уровень доступа для конкретных сотрудников. Ознакомьтесь с приведенными ниже настройками пользователей, чтобы нужные сотрудники могли контролировать нужный объем информации.

**Обычный пользователь.** Может просматривать и редактировать карточки, а также в зависимости от настроек изменять некоторые права доступа к доскам. Отлично подходит для участников рабочего пространства, активно задействованных в проекте.

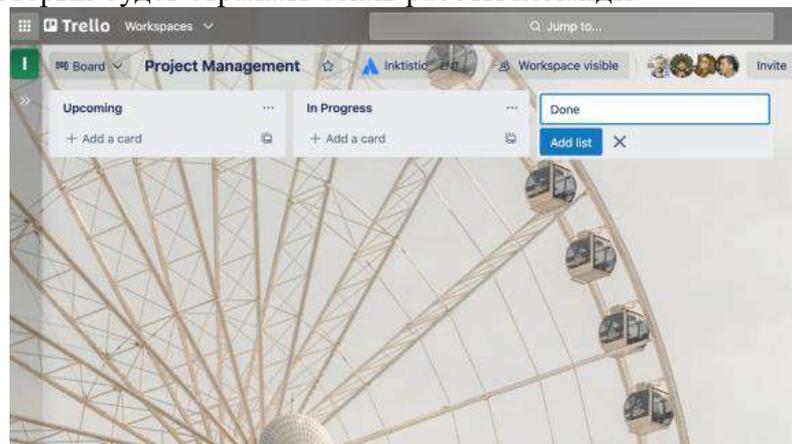
**Администратор.** Может просматривать и редактировать карточки, исключать участников и изменять все настройки доски. Отлично подходит для руководителей проектов и менеджеров, которые полностью контролируют проект.

**Наблюдатель\*.** Может просматривать содержимое, отображаемое на доске, но не может редактировать карточки. Наблюдателям можно разрешить оставлять комментарии, отвечать с помощью эмодзи и выполнять другие действия в зависимости от настроек доски. Отлично подходит для досок с информацией для клиентов, менеджеров, третьих сторон и т. п.



### 5. Создание рабочего процесса

Колонки Trello позволяют упорядочить карточки по статусу. В колонке можно объединить ряд элементов, например идей или заданий, в рамках более крупного проекта. Но самая главная задача — будь то стандартная доска Kanban, конвейер продаж, маркетинговый календарь или сложная система управления проектами — сформировать рабочий процесс, который будет отражать стиль работы команды.



1. Нажмите Add a list (Добавить колонку), чтобы добавить на доску свою первую колонку. В названии отразите ее содержание или этап работы. Названия колонок могут быть простыми, как для этапов рабочего процесса («Нужно сделать», «В процессе» или «Готово»), либо подробными, если это нужно для работы команды. Помните: Trello

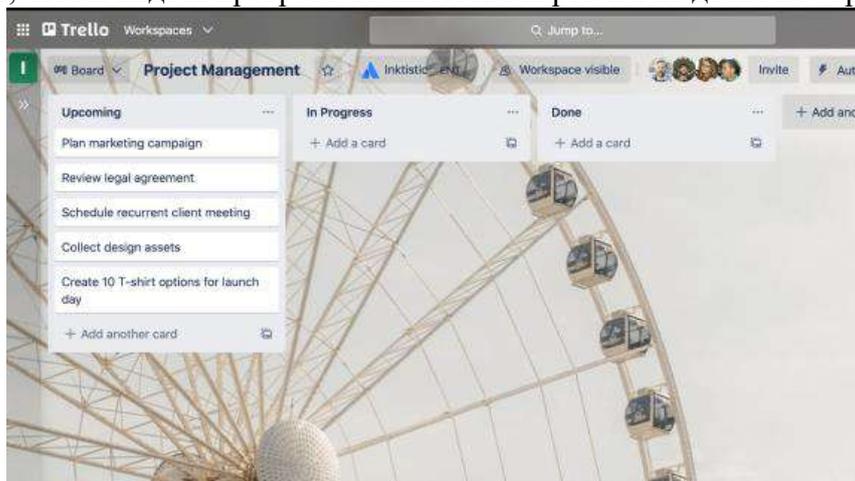
очень просто настроить под уникальные особенности вашего рабочего процесса, а колонкам можно давать любые названия.

2. Добавьте на доску столько колонок, сколько нужно для организации рабочего процесса.

## 6. Добавление заданий и задач

Основной элемент доски, карточка, представляет собой задание или идею. Карточка описывает определенную задачу (например, задача по поддержке клиентов, разработка нового дизайна или оформление записи в блоге) либо содержит информацию, связанную с проектом, которая должна оставаться на виду.

Карточки можно настроить так, чтобы хранить самую разную полезную информацию, которую можно просмотреть одним нажатием. Перетаскивайте карточки между колонками, чтобы видеть прогресс. Количество карточек на доске не ограничено.



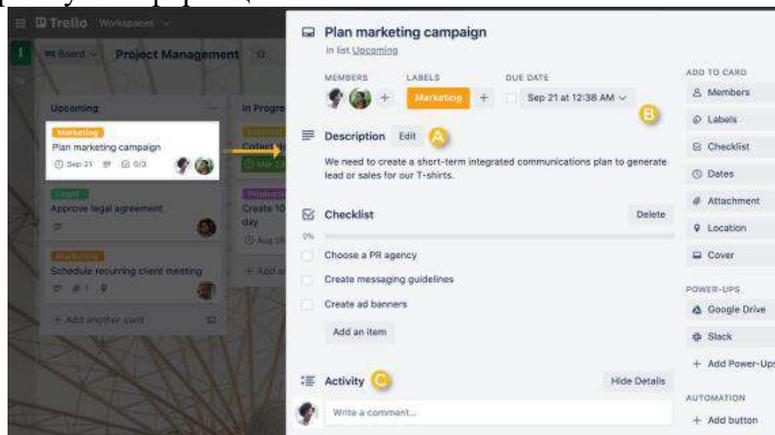
1. Добавляйте карточки для каждого задания, которое нужно выполнить, нажимая Add A Card (Добавить карточку) в первой колонке. Старайтесь давать карточкам короткие названия, чтобы их было проще найти на доске и соотнести со статусом.

2. Чтобы каждый сотрудник имел четкое представление о том, что нужно сделать, нажмите карточку и добавьте в нее дополнительную информацию, например следующую.

- Описание
- Даты выполнения
- Участники
- Списки задач
- Вложения
- Комментарии

7. Добавьте подробности на обратной стороне карточки

Как упоминалось ранее, нажав карточку, можно развернуть ее для просмотра или добавления сведений. Открывшееся окно называется обратной стороной карточки и содержит более подробную информацию.



Здесь находятся три основных раздела, с которыми стоит ознакомиться.

**А. Описание карточки.** При ежедневном общении в команде важно пополнять контекст и представлять его в понятном виде. При наличии актуальной информации можно подготовиться к любому взаимодействию и избежать недопонимания в любых вопросах касательно проекта. Это особенно ценно для тех участников команды, которые работают с доской не ежедневно, а лишь периодически.

В поле описания можно добавить более подробную информацию о карточке, например ссылки на веб-сайты или пошаговые инструкции. Чтобы добавить сведения в карточку, нажмите Edit the description (Редактировать описание) в верхней части обратной стороны карточки. Здесь даже можно использовать разметку для форматирования текста.

**В. Раздел добавления.** В этом разделе обратной стороны карточки собраны дополнительные инструменты.

- Добавляйте участников в карточки, чтобы назначать им задания и без труда отслеживать, чем занят каждый и что еще требуется сделать.

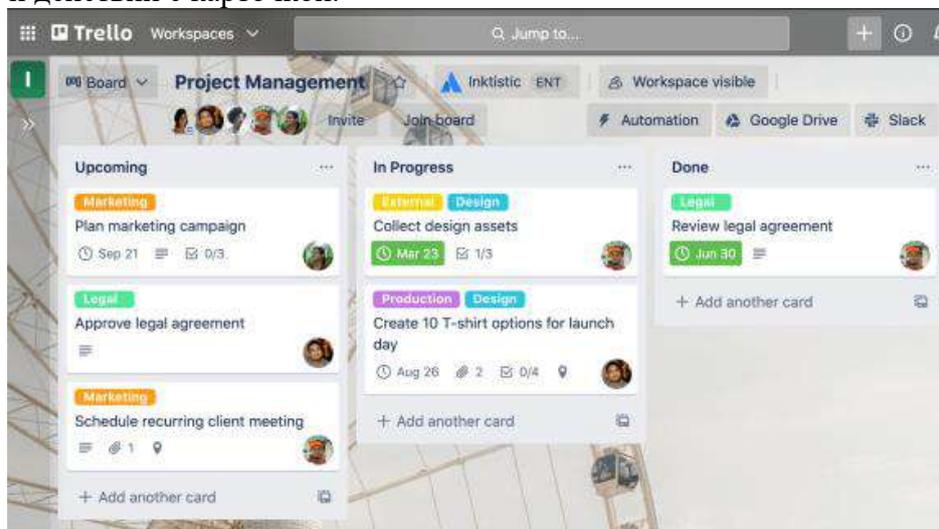
- Добавляйте в карточки списки задач, которые содержат подзадачи или этапы выполнения. Так вы не упустите ни одной мелочи. Вы даже можете копировать списки задач из других карточек на доске. Кроме того, [клиентам с подпиской Business Class](#) доступны усовершенствованные списки задач, для элементов которых можно назначать участников и даты выполнения, что позволит внести дополнительную ясность и подотчетность.

- Добавляйте в карточки даты, чтобы отслеживать сроки: все участники карточки получают уведомление за 24 часа до наступления срока. В меню дат можно выбрать дату начала и дату выполнения, чтобы отслеживать начало и окончание проектов и заданий. Как только задания будут завершены, для соответствующих дат можно установить отметку о выполнении.

- Добавляйте вложения с компьютера или из облачного сервиса (например, Dropbox, Google Диск, Box или OneDrive).

- Обложки помогают дополнительно упорядочить доски и сделать их наглядными. Добавляя в карточки обложки и настраивая цвета, вы создаете визуальный контекст и можете использовать дополнительные обозначения и сведения. Вложенные изображения карточки можно показывать на переднем плане в качестве полноразмерных обложек. Параллельно с этим есть возможность менять цвета карточек.

**С. Комментарии и события.** В ходе совместной работы и общения в команде можно добавлять комментарии к карточкам, в том числе обратную связь и обновления. Используйте в комментариях @упоминания участников доски или команды, чтобы отправить им уведомление в Trello. В ленте событий показана хронология всех комментариев и действий с карточкой.



В Trello вся информация хранится в одном месте, чтобы все были в курсе происходящего — убедитесь в этом сами. Все готово для успешной работы вашей команды!

**Задание для самостоятельного выполнения:**

Подготовить план разработки приложения демоэкзамена (разработка ег-диаграммы, создание базы данных, импорт данных и т.д.) в онлайн-сервисе Trello.

**Критерии оценивания:**

*оценка «5» ставится, если:*

- обучающийся самостоятельно выполнил все этапы решения задач на компьютере;
- работа выполнена полностью и получен верный ответ или иное требуемое представление результата работы;

*оценка «4» ставится, если:*

- работа выполнена полностью, но при выполнении обнаружилось недостаточное владение навыками работы с программным обеспечением в рамках поставленной задачи;
- правильно выполнена большая часть работы (свыше 85 %), допущено не более трех ошибок;
- работа выполнена полностью, но использованы наименее оптимальные подходы к решению поставленной задачи.

*оценка «3» ставится, если:*

- работа выполнена не полностью, допущено более трех ошибок, но обучающийся владеет основными навыками работы со средой разработки, требуемыми для решения поставленной задачи.

*оценка «2» ставится, если:*

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными знаниями, умениями и навыками работы с программным обеспечением и средой разработки или значительная часть работы выполнена не самостоятельно.

## 2.2 Теоретические вопросы для проведения устного опроса

**– принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения;**

1. Что представляют собой CASE-средства разработки ИС?
2. Перечислите требования, предъявляемые к методикам и программным инструментальным средствам разработки ИС.

3. В чем заключаются задачи календарного планирования

**– инструментальные средства поддержки тестирования**

1. Назовите известные вам программные средства поддержки тестирования.
2. Средства тестирования для разработчиков.
3. Дополнительные возможности средств тестирования для разработчиков

**КРИТЕРИИ ОЦЕНКИ:**

Оценка «отлично» выставляется при 3 правильных ответах.

Оценка «хорошо» выставляется при 2 правильных ответах.

Оценка «удовлетворительно» выставляется при 1 правильном ответе.

Если студент не ответил ни на один вопрос, то выставляется оценка «неудовлетворительно»

2.3. Тесты для проведения тестирования

**– методы организации работы в коллективах разработчиков программного обеспечения;**

Задание №1

Вопрос:

Что такое модель жизненного цикла программного обеспечения

Выберите один из 5 вариантов ответа:

1) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

2) модель содержащая процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

3) действия содержащие процессы действия и задачи, которые осуществляются в ходе разработки, использования и сопровождения программного продукта.

4) структура, содержащая процессы задачи, которые осуществляются в ходе использования и сопровождения программного продукта.

5) структура, содержащая процессы действия и задачи, которые осуществляются в ходе разработки.

Задание №2

Вопрос:

Дана модель:

1-Постановка задачи

2-Выполнение

3-Проверка результата

4-При необходимости переход к первому пункту

Выберите название данной модели

Выберите один из 5 вариантов ответа:

- 1) Каскадная модель
- 2) Модель кодирования и устранения ошибок
- 3) Каскадная модель с промежуточным контролем
- 4) V модель
- 5) Спиральная модель

Задание №3

Вопрос:

Какая это модель жизненного цикла программного обеспечения?

(Нажмите на изображение)

Изображение:



Выберите один из 5 вариантов ответа:

- 1) Каскадная модель (водопад)
- 2) Каскадная модель с промежуточным контролем
- 3) V модель
- 4) Модель кодирования и устранения ошибок
- 5) Спиральная модель жизненного цикла ПО

Задание №4

Вопрос:

Количество стадий, разработки программного обеспечения

Выберите один из 5 вариантов ответа:

- 1) 7
- 2) 6
- 3) 3
- 4) 5
- 5) 4

Задание №5

Вопрос:

Согласно ГОСТ 19.102-77, в этап: Разработка программы входит

Выберите один из 5 вариантов ответа:

- 1) Программирование и отладка программы.
- 2) Разработка плана мероприятий по разработке и внедрению программ.
- 3) Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.
- 4) Разработка, согласование и утверждение порядка и методики испытаний.
- 5) Корректировка программы и программной документации по результатам испытаний.

Задание №6

Вопрос:

Согласно ГОСТ 19.102-77, на стадии 1.Техническое задание выполняется

Выберите один из 5 вариантов ответа:

- 1) Постановка задачи. Сбор исходных материалов
- 2) Уточнение структуры входных и выходных данных. Постановка задачи.
- 3) Сбор исходных материалов, Разработка алгоритма решения задачи
- 4) Разработка структуры, Постановка задачи, Сбр исходных материалаов

5) Постановка задачи, Сбор исходных материалов, Обоснование необходимости проведения научно-исследовательских работ

Задание №7

Вопрос:

ГОСТ 19.001-77 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) Виды программ
- 3) Стадии разработки
- 4) Общие требования
- 5) Основные надписи

Задание №8

Вопрос:

ГОСТ 19.101-77 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Виды программ
- 2) Виды программ и программной документации
- 3) Виды программной документации и общие положения
- 4) Виды программ и общие положения
- 5) Виды программ и программной обозначений

Задание №9

Вопрос:

Что такое ЕСПД?

Выберите один из 5 вариантов ответа:

- 1) Единая система программной документации
- 2) Единая система проектной документации
- 3) Единый стандарт проектной документации
- 4) Единственный стандарт программной документации

5) Нет ответа

Задание №10

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “0” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации изготовления
- 5) Правила обобщения программной документации

Задание №11

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “1” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации изготовления
- 5) Правила обобщения программной документации

Задание №12

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “2” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты

- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации изготовления
- 5) Правила обобщения программной документации

Задание №13

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “3” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации изготовления
- 5) Правила обобщения программной документации

Задание №14

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “4” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации сопровождения
- 5) Правила обобщения программной документации

Задание №15

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “5” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации эксплуатации документации
- 5) Правила обобщения программной документации

Задание №16

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “6” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Общие положения
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации эксплуатации документации
- 5) Правила обобщения программной документации

Задание №17

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “7/8” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Резервные группы
- 2) основополагающие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации эксплуатации документации
- 5) Правила обобщения программной документации

Задание №18

Вопрос:

Классификация и обозначения стандартов ЕСПД. За что отвечает группа “9” в стандартах ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Резервные группы
- 2) Прочие стандарты
- 3) Правила выполнения документации разработки
- 4) Правила выполнения документации эксплуатации документации
- 5) Правила обобщения программной документации

Задание №19

Вопрос:

2-ух значное число после точки в стандартах ЕСПД означает

Выберите один из 5 вариантов ответа:

- 1) Порядковый номер стандарта в группе
- 2) Год регистрации стандарта
- 3) Классификацию группы стандарта
- 4) Общие положения
- 5) Порядковый номер стандарта в регистрационной базе

Задание №20

Вопрос:

2-ух значное число до тире в стандартах ЕСПД означает

Выберите один из 5 вариантов ответа:

- 1) Порядковый номер стандарта в группе
- 2) Год регистрации стандарта
- 3) Классификацию группы
- 4) Общие положения
- 5) Порядковый номер стандарта в регистрационной базе

Задание №21

Вопрос:

ГОСТ 19.102-77 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Стадии разработки
- 2) Виды программ
- 3) Стадии разработки
- 4) Общие требования
- 5) Основные надписи

Задание №22

Вопрос:

ГОСТ 19.103-77 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Стадии разработки
- 2) Обозначения программ и программных документов
- 3) Стадии разработки
- 4) Общие требования
- 5) Основные надписи

Задание №23

Вопрос:

ГОСТ 19.104-78 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Стадии разработки
- 2) Основные надписи
- 3) Стадии разработки
- 4) Общие требования
- 5) Основные надписи

Задание №24

Вопрос:

ГОСТ 19.402-78 отвечает за

Выберите один из 5 вариантов ответа:

- 1) Стадии разработки
- 2) Описание программы
- 3) Стадии разработки
- 4) Общие требования
- 5) Основные надписи

Задание №26

Вопрос:

Укажите правильную запись стандарта ЕСПД

Выберите один из 5 вариантов ответа:

- 1) ГОСТ 19-001-787
- 2) ГОСТ 19-002.487
- 3) ГОСТ Е19.00-1997
- 4) ГОСТ 19.103-77г.
- 5) ГОСТ 19.103-77

Задание №27

Вопрос:

Что такое стиль программирования?

Выберите один из 5 вариантов ответа:

- 1) Набор приемов и методов программирования, которые необходимы соблюдать при написании программы
- 2) Программирование, которое стилизуется при написании программы
- 3) Хороший стиль программирования
- 4) Набор элементов, которые образуют надежность, дружелюбность, отличный интерфейс
- 5) Использование отступов

Задание №28

Вопрос:

Что такое надежность?

Выберите один из 5 вариантов ответа:

- 1) Программа контролирует исходные данные, проверяет результат выполнения операции
- 2) Программа контролирует выходные данные, проверяет результат выполнения
- 3) Предполагает хорошо спроектированные диалоговые окна
- 4) Хороший интерфейс
- 5) Качественный код

Задание №29

Вопрос:

Что такое дружелюбность?

Выберите один из 5 вариантов ответа:

- 1) Программа контролирует исходные данные, проверяет результат выполнения операции
- 2) Программа контролирует выходные данные, проверяет результат выполнения
- 3) Предполагает хорошо спроектированные диалоговые окна
- 4) Хороший интерфейс
- 5) Качественный код

Задание №30

Вопрос:

Что такое транслятор?

Выберите один из 5 вариантов ответа:

- 1) Программа для переводы с языка программирования на машинные коды
- 2) Программа для изменения кода
- 3) Программа для создания изменений исходных программ

4) Программа для перевода из машинного кода в язык программирования

5) Переводит исходный текст в программный код

Задание №31

Вопрос:

Что такое компилятор?

Выберите один из 5 вариантов ответа:

1) Переводит исходный текст программы в язык программирования низкого уровня

2) Переводит исходный текст программы в язык программирования высокого уровня

3) Переводит исходный текст программы в процедуру

4) Переводит программу в функцию

5) Нет правильного ответа

Задание №32

Вопрос:

Что выполняет интерпретатор?

Выберите один из 5 вариантов ответа:

1) Выполняет покомандную обработку текста программы

2) Выполняет перевод

3) Выполняет покомандную переводку текста программы

4) Выполняет текст программы

5) Выполняет функцию обработки текста программы

Задание №33

Вопрос:

Что такое структурное программирование?

Выберите один из 5 вариантов ответа:

- 1) Методология разработки ПО, в основе которых лежит представление программ в виде иерархической структуре блоков
- 2) Методология разработки ПО, в основе которых лежит представление программ в виде блоков
- 3) Разработки из частей
- 4) Программирование по частям одной процедуры
- 5) Методология разработки ПО, в основе которых лежит представление программ в виде линейной задачи

#### Задание №34

Вопрос:

Цель структурного программирования:

Выберите один из 5 вариантов ответа:

- 1) Снижение памяти всего процесса создания ПО
- 2) Снижение затрат всего процесса создания ПО
- 3) Снижение трудоемкости всего процесса создания ПО
- 4) Снижение работоспособности всего процесса создания ПО
- 5) Снижение трудоемкости описания создания ПО

#### Задание №35

Вопрос:

Какого вида ошибок не существует?

Выберите один из 5 вариантов ответа:

- 1) Синтаксические
- 2) Алгоритмические
- 3) Нет правильного ответа
- 4) Ошибки времени выполнения
- 5) Структурированные

#### Задание №36

Вопрос:

Синтаксические ошибки, это ошибки

Выберите один из 5 вариантов ответа:

- 1) ошибки времени компиляции
- 2) ошибки при выполнении
- 3) ошибки при подсчете
- 4) ошибки во время не выполнения
- 5) ошибки некорректного ввода

Задание №37

Вопрос:

Алгоритмические ошибки, это ошибки

Выберите один из 5 вариантов ответа:

- 1) ошибки времени компиляции
- 2) ошибки при выполнении
- 3) ошибки при подсчете из-за не правильного алгоритма
- 4) ошибки во время не выполнения
- 5) ошибки некорректного ввода

Задание №38

Вопрос:

Угроза это

Выберите один из 5 вариантов ответа:

- 1) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению хороших данных
- 2) возможное событие, действие, процесс или явление, которое может привести к сохранности данных
- 3) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению изменению данных
- 4) потенциальное возможное событие, действие, процесс или явление, которое может привести к нанесению ущерба.
- 5) нет правильного ответа

Задание №39

Вопрос:

Область распространения ЕСПД

Выберите один из 5 вариантов ответа:

- 1) Распространяются на документы
- 2) Распространяются на компьютерное оборудование
- 3) Распространяется на программы
- 4) Распространяется на чертежи
- 5) Распространяется на

Задание №40

Вопрос:

Из каких 2 составляющих состоит надежность программ

Выберите один из 5 вариантов ответа:

- 1) Корректность и Контроль
- 2) Корректность и Обеспечение
- 3) Корректность и Устойчивость
- 4) Устойчивость и Контроль
- 5) Обеспечение и Контроль

– *инструментальные средства поддержки процесса документирования*

1. Зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать:

- 1) Сообщение;
- 2) Документ;
- 3) Текст.

2. Что не относится к свойствам, которыми должно обладать сообщение, чтобы стать документом?

- 1) Легитимность;
- 2) Доступность;

- 3) Качество.
3. Что не относится к внутренним признакам документа?
  - 1) Стиль;
  - 2) Управляющая сила;
  - 3) Размер.
4. К каким признакам документа относятся: авторство, наименование вида документа, датирование, удостоверение, форма и размер, способ документирования и используемые носители?
  - 1) Внешние;
  - 2) Документообразующие;
  - 3) Внутренние.
5. По какому признаку классифицированы следующие документы: протоколы, приказы, отчеты?
  - 1) Вид документа;
  - 2) Качество документа;
  - 3) Разновидность документа.
6. По какому признаку классифицирована информация на общедоступную и конфиденциальную?
  - 1) Порядок представления;
  - 2) Порядок распространения;
  - 3) Категория доступа.
7. Что не относится к юридически значимым элементам оформления документа?
  - 1) Телефон организации, являющейся автором документа;
  - 2) Подпись документа;
  - 3) Регистрационный номер документа.
8. Частное понятие, которое детализирует, уточняет характер деятельности, документируемой тем или иным видом:
  - 1) Вид документа;
  - 2) Разновидность документа;
  - 3) Признак документа.

9. Совокупность присущих документу существенных признаков и особенностей, позволяющих выделить его из среды других предметов:

- 1) Качество документа;
- 2) Признак документа;
- 3) Вид документа.

10. К чему относится изменение во времени яркости элементов изображения в телевизионных системах?

- 1) Документ;
- 2) Информация;
- 3) Сообщение.

11. Информация, содержащая сведения об источниках, где могут находиться необходимые данные (факты):

- 1) Постоянная;
- 2) Фактографическая;
- 3) Документальная.

12. Задача документоведения:

- 1) Практическое обоснование процессов документационного обеспечения аппарата управления обществом;
- 2) Теоретическое обоснование процессов документационного обеспечения аппарата управления обществом;

#### КРИТЕРИИ ОЦЕНКИ:

При выполнении заданий ставится отметка:

«отлично» - за правильное выполнение более 85% заданий.

«хорошо» - за 70-85% правильно выполненных заданий,

«удовлетворительно» - за 50-70% правильно выполненных заданий,

При выполнении менее 50% ставится оценка «неудовлетворительно»

#### 2.4 Задания для проведения дифференцированного зачета

Комплект материалов для оценки сформированных общих и профессиональных компетенций по виду профессиональной деятельности курса МДК.03.02. с использованием стека технологий WorldSkills:

1.Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

2.Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

3.Разработать приложение Windows «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.

4.Разработать приложение Windows «Калькулятор». Приложение предназначено для любых пользователей и должно содержать все арифметические операции (с соблюдением приоритетов) и желательно (но не обязательно) несколько математических функций.

5.Разработать программный модуль «Кафедра», содержащий сведения о сотрудниках кафедры (ФИО, должность, ученая степень, дисциплины, нагрузка, общественная работа, совместительство и др.). Модуль предназначен для использования сотрудниками отдела кадров и деканата.

6.Разработать программный модуль «Лаборатория», содержащий сведения о сотрудниках лаборатории (ФИО, пол, возраст, семейное положение, наличие детей, должность, ученая степень). Модуль предназначен для использования сотрудниками профкома и отдела кадров.

7. Разработать программный модуль «Автосервис». При записи на обслуживание заполняется заявка, в которой указывают ФИО владельца, марка автомобиля, вид работы, дата приема заказа и стоимость ремонта. После выполнения работ распечатывается квитанция.

8. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.

9. Разработать программный модуль «Картотека агент недвижимости», предназначенный для использования работниками агентства. В базе содержатся сведения о квартирах (количество комнат, этаж, метраж и др.). При поступлении заявки обмен (куплю, продажу) производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.

10. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и повременной). Считается, что повременная оплата местных телефонных разговоров уже введена.

11. Разработать программный модуль «Авиакасса», содержащий сведения о наличии свободных мест на авиамаршруты. В базе должны содержаться сведения о номере рейса, экипаже, типе самолета, дате и времени вылета, а также стоимости авиабилетов (разного класса). При поступлении заявки на билеты программа производит поиск подходящего рейса.

12. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.

13. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.

14. Разработать программный модуль «Кадровое агентство», содержащий сведения о вакансиях и резюме. Программный модуль предназначен как для поиска сотрудника, отвечающего требованиям руководителей фирмы, так и для поиска подходящей работы.

15. Создать ER-диаграмму базы данных Музей

Экспонаты : таблица		
Имя поля	Тип данных	
Код экспоната	Числовой	
Наименование	Текстовый	
Код зала	Числовой	
Дата поступления	Дата/время	
Автор	Текстовый	
Материал	Текстовый	
Техника	Текстовый	

Залы : таблица		
Имя поля	Тип данных	
Код зала	Числовой	
Наименование	Текстовый	
Код ответственного	Числовой	

Экскурсии : таблица		
Имя поля	Тип данных	
Код экскурсии	Числовой	
Время проведения	Дата/время	
График	Текстовый	
Срок действия	Дата/время	
Код ответственного	Числовой	
Стоимость	Денежный	

Залы : таблица		
Имя поля	Тип данных	
Код зала	Числовой	
Наименование	Текстовый	
Код ответственного	Числовой	

16. Создать ER-диаграмму базы данных Спортивный клуб

Клиенты				
Код_клиента	Фамилия	Имя	Телефон	Код_тренера
1	Тараблина	Елена	89053214571	112
2	Зайцев	Вячеслав	89057841232	112
3	Горбачев	Сергей	89031255478	114
4	Светлаков	Леонид	89631522354	119
5	Быков	Илья	89634178578	118
6	Быкова	Светлана	89631445286	115
7	Бирюков	Роман	89104500877	113
8	Васильева	Наталья	89196603122	115
9	Лужков	Николай	89512255673	116
10	Титов	Сергей	89654122157	117

Абонементы			
Код_абонемена	Описание	Цена	Код_зала
30101	2 раза в неделю	350,00р.	1001
30102	2 раза в неделю	300,00р.	1002
30103	2 раза в неделю	250,00р.	1003
30104	3 раза в неделю	450,00р.	1001
30105	3 раза в неделю	400,00р.	1002
30106	3 раза в неделю	320,00р.	1004
30107	3 раза в неделю	320,00р.	1005
30108	3 раза в неделю	420,00р.	1006
30109	3 раза в неделю	420,00р.	1008
30110	1 раз в неделю	800,00р.	1007
30111	2 раза в неделю	1 400,00р.	1007
30112	3 раза в неделю	2 000,00р.	1007

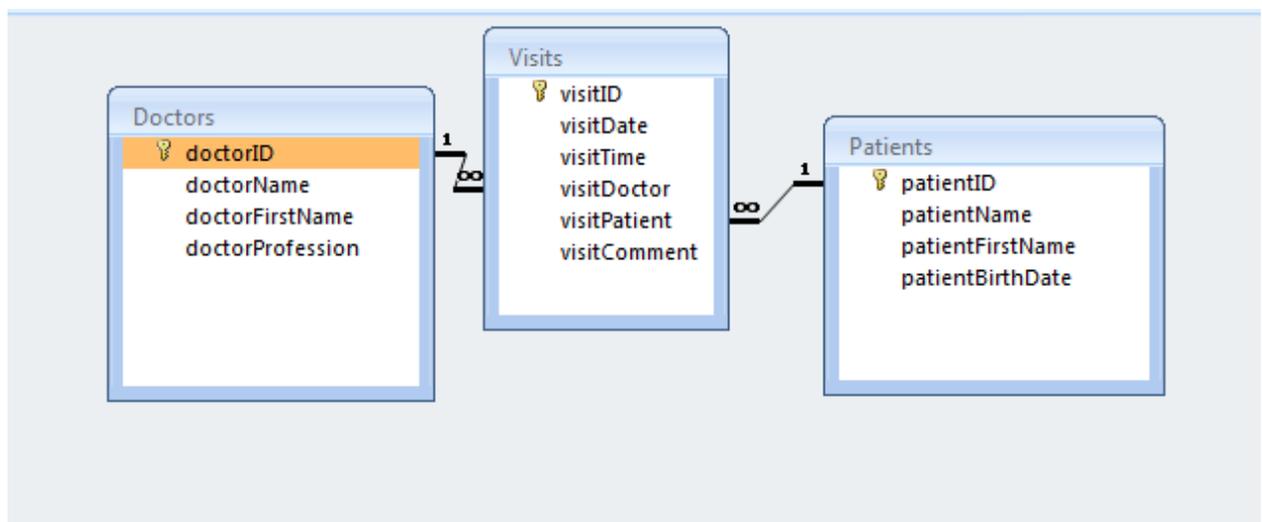
Залы	
Код_зала	Наименование
1001	Бассейн 25м
1002	Бассейн 15м
1003	Бассейн детский
1004	Тренажерный1
1005	Тренажерный2
1006	Фитнес1
1007	Спортивный
1008	Фитнес2

Тренеры		
Код_тренера	Фамилия_имя_тренера	Оклад
111	Иванов Алексей	12 300,00р.
112	Рыбкина Валентина	11 500,00р.
113	Королев Иван	15 000,00р.
114	Пронин Александр	7 500,00р.
115	Малкина Ольга	7 500,00р.
116	Часовских Михаил	11 500,00р.
117	Лебедев Анатолий	12 000,00р.
118	Карнаух Александр	10 500,00р.
119	Баранов Дмитрий	7 500,00р.
120	Попова Инна	9 300,00р.

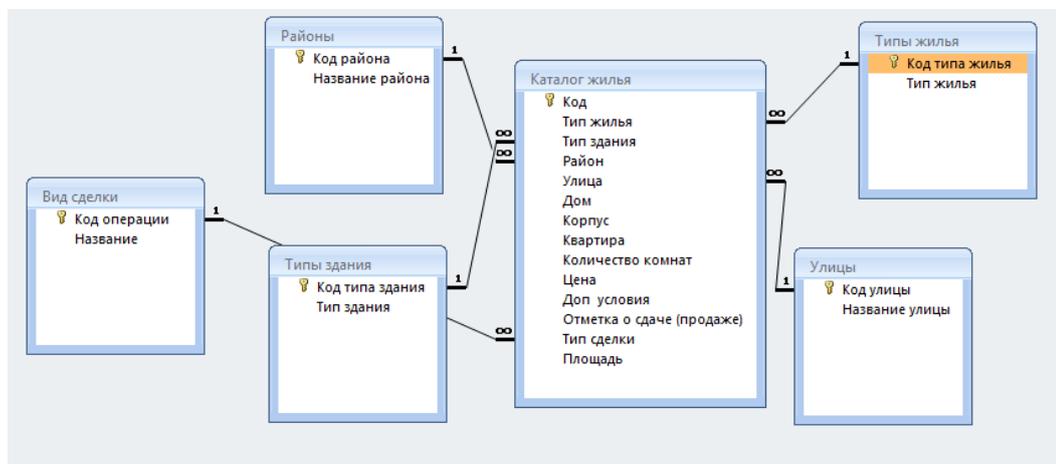
17. Создать ER-диаграмму базы данных Сведения о клиентах и заказах



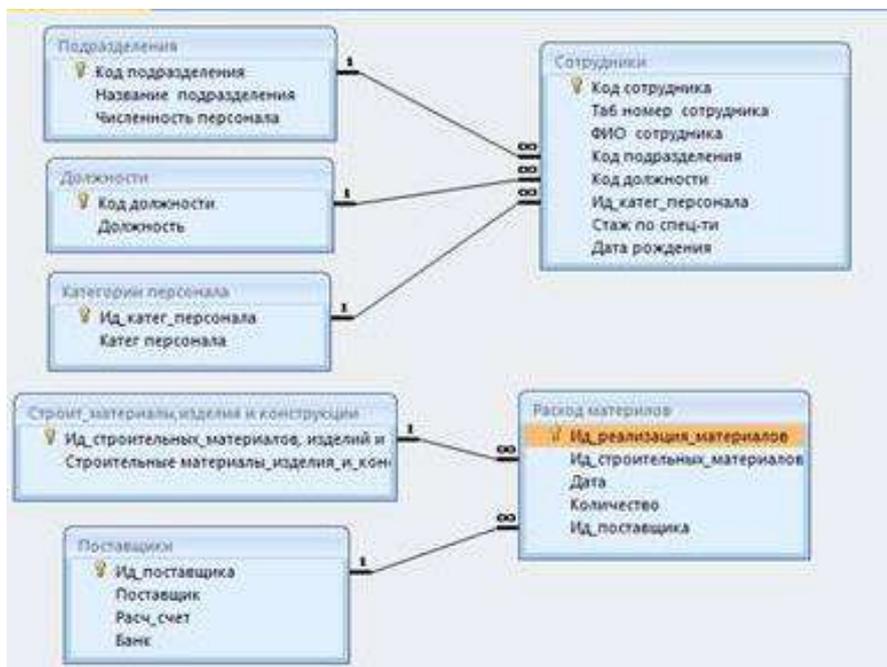
18. Создать ER-диаграмму базы данных Поликлиника



19. Создать ER-диаграмму базы данных Агентство недвижимости



20. Создать ER-диаграмму базы данных Строительная компания



Количество вариантов: 20 (1 задание на выбор из вариантов 1-15 и 1 задание на выбор из вариантов 15-20)

Время выполнения каждого задания: (1 задание -60 мин, 2 задание – 30 мин)

Оборудование:

Оборудование рабочих мест обучающихся: ноутбук Dell Inspiron 15 5000 Series

Программное обеспечение:

- Microsoft Visio 2019
- Microsoft Access 2010
- MS SQL Server+Manadgment Studio
- MySQL + Workbench
- Embarcadero RAD Studio XE2

Литература для обучающегося:

Основные источники:

1. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие/ под ред. Л.Г. Гагариной. -М.: ИД «ФОРУМ»: ИНФРА-М, 2015.
2. Рудаков А.В. Технология разработки программных продуктов: Учеб. Пособие для сред. проф. образования – М.: Издательский центр «Академия», 2012

3. Рудаков А.В., Федорова Г.Н. Технология разработки программных продуктов: Практикум. Учеб. Пособие для сред. проф. образования – М.: Издательский центр «Академия», 2012
4. Сетевая энциклопедия Википедия
5. Методы и средства инженерии программного обеспечения, НОУ ИНТУИТ, электронное учебное пособие

Дополнительные источники:

1. Соммерфилл, Иан. Инженерия программного обеспечения, 6-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002
2. С.Л. Котов, Б.В. Палюх, С.Л.Федченко, Разработка, стандартизация и сертификация программных средств и информационных технологий и систем. Тверь.: ТТУ, 2006. -104с.
3. Брауде Э. Технология разработки программного обеспечения. — СПб.: Питер, 2004.
4. Орлов С. А. Технологии разработки программного обеспечения. — СПб.: Питер, 2010.
5. Басс, Л. Архитектура программного обеспечения на практике /Л. Басс, П. Клементс, П. Кацман. – 2-е изд. – СПб. : Питер, 2006. – 575 с.
6. Камаев, В. А. Технологии программирования / В. А. Камаев, В. В. Костерин. – М. : Высш. шк., 2005. – 359 с.
7. Якунин, Ю. Ю. Технологии разработки программного обеспечения: учеб. пособие / Ю. Ю. Якунин. – Красноярск: ИПК СФУ, 2008. – 232 с. – (Технологии разработки программного обеспечения: У МКД № 1 83-2007 /рук. творч. коллектива Ю. Ю. Якунин).
8. Крупский А.Ю. Разработка и стандартизация программных средств: Учебное пособие/А.Ю. Крупский, Л. А. Феоктистова. – М.: Издательская торговая корпорация «Дашков и К», 2009. – 100 с.
9. Благодатских В.А. Стандартизация разработки программных средств: Учебное пособие/ В.А. Благодатских, В.А. Волнин, К.Ф. Посакалов; Под ред. О.С. Разумова. – М.: Финансы и статистика, 2005. – 288 с.
10. С.Л. Котов, Б.В. Палюх, С.Л.Федченко, Разработка, стандартизация и сертификация программных средств и информационных технологий и систем. Тверь.: ТТУ, 2006. -104 с
11. Благодатских В. А. др. Стандартизация разработки программных средств: Учеб. пособие / Под ред. О. С. Разумова — М.: Финансы и статистика, 2005
12. Крупский А. Ю. Разработка и стандартизация программных средств: Учебное пособие — М.: Дашков и К°, 2009
13. С. В. Сеницын, Н. Ю. Налютин Верификация программного обеспечения: Курс лекций – М.: МИФИ, 2006
14. Богданов Д. В. и др. Стандартизация процессов обеспечения качества программного обеспечения – Апатиты: КФ ПетрГУ, 1997
15. Богданов Д. В., Фильчаков В. В. Стандартизация жизненного цикла и качества программных средств: Учеб. пособие – СПбГУАП. СПб, 2000
16. Гудов А. М. Метрология и качество программного обеспечения: Конспект лекций – Кемерово: Кемеровский гос. ун-т, 2009
17. Метрология программного обеспечения. Экономические и правовые основы разработки программного обеспечения (Текст лекций) – Энгельс: Саратовский Гос Технич. Ун-т., 2003
18. Ковалевская Е.В. Материалы к курсу «Метрология, качество и сертификация программного обеспечения» – М.: Моск. гос. ун-т экон-ки, стат-ки и инф-ки, 2002
19. Андон Ф.И. и др. Основы инженерии качества программных систем – К: Академперіодика, 2002

20. Эванс, Эрик Предметно-ориентированное проектирование (DDD): структуризация сложных программных систем.: Пер.с англ. – М.: ООО «И.Д. Вильямс», 2011. – 448 с.: ил. – Парал.тит.англ.
21. Гелмерс Скотт А. Microsoft Visio 2010. Русская версия. Серия «Шаг за шагом»/ Гелмерс С.А.; пер.с англ. – М.: ЭКОМ Паблшерз, 2011. – 576 с.: ил.
22. Эдуардо Коста Visual Prolog 7.1 для начинающих. Перевод с английского. Перевод: И. Алексеев, Е.А. Ефимова, 2008.
23. ISO 12182:1998. (ГОСТ Р– 2002) – ИТ. Классификация программных средств.
24. ISO 9126:1991. (ГОСТ – 1993) – ИТ. Оценка программного продукта. Характеристики качества и руководство по их применению.
25. ISO 9126:1-4:2002-2004 ТО – Качество программных средств:
26. ISO 14598:1-6:1998-2000 – Оценивание программного продукта.
27. ISO 25000:2005 ТО. – Руководство для применения новой серии стандартов по качеству программных средств
28. IEC 61508:1-6:1998-2000 – Функциональная безопасность электрических / электронных и программируемых электронных систем.
29. ISO 15408:1-3:1999 (ГОСТ Р – 2002) – Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий.
30. IEC 60880:1996-2000 – Ч. 1.1986. Программное обеспечение компьютеров в системах безопасности атомных электростанций.
31. ISO 6592:2000 (ГОСТ) – Руководство по документации для вычислительных систем.
32. ISO 9294:1990 (ГОСТ?1993 г). ТО. ИТ. Руководство по управлению документированием программного обеспечения.
33. ISO 15910:1999 (ГОСТ Р – 2002) ИТ. Пользовательская документация программных средств.
34. ISO 18019:2004 ИТ. Руководство по разработке пользовательской документации на прикладные программные средства
35. Введение в распределенные алгоритмы: Ж. Тель — Санкт-Петербург, МЦНМО, 2009 г.- 616 с.
36. Верификация программного обеспечения: С. В. Сеницын, Н. Ю. Налютин — Санкт-Петербург, Бином. Лаборатория знаний, Интернет-университет информационных т, 2008 г.- 368 с.

Периодическая литература

## КРИТЕРИИ ОЦЕНКИ

Оценка «отлично» выставляется если задание выполнено полностью.

Оценка «хорошо» выставляется при выполнении задания с небольшими неточностями.

Оценка «удовлетворительно» выставляется при выполнении задания с неточностями

Оценка «неудовлетворительно» выставляется, если задание выполнено не точно.

ПРИЛОЖЕНИЕ А  
(обязательное)

Лист ознакомления обучающихся

**ЛИСТ ОЗНАКОМЛЕНИЯ ОБУЧАЮЩИХСЯ**  
с формами, процедурой текущего, рубежного контроля знаний, промежуточной аттестации по дисциплинам, профессиональным модулям, содержанием комплекта оценочных средств

Дисциплина МДК.03.02.Инструментальные средства разработки программного обеспечения

*код и наименование*

Группа 4-П-1, 4-П-2

Специальность 09.02.03 Программирование в компьютерных системах

*код и наименование*

Преподаватель Барилова С.В., Крайнева А.А.

№	ФИО обучающихся	Подпись	Примечание
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			

Преподаватель \_\_\_\_\_ С.В. Барилова

Председатель УМО \_\_\_\_\_ О.А. Афиногенова