

Министерство образования и науки Республики Дагестан
Государственное бюджетное профессиональное образовательное
учреждение
«Дербентский профессионально-педагогический колледж
им.Г.Б.Казиахмедова»

Комплект

**контрольно-оценочных средств по профессиональному
модулю**

ПМ 05. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНФОРМАЦИОННЫХ СИСТЕМ

для специальности 09.02.07 Информационные системы и программирование
квалификация Разработчик веб и мультимедийных приложений

Дербент, 2025

КОС профессионального модуля составлен в соответствии с рабочей программой ПМ.05 Проектирование и разработка информационных систем

Организация-разработчик: Государственное бюджетное профессиональное образовательное учреждение
«Дербентский профессионально-педагогический колледж
им.Г.Б.Казиахмедова»

Разработчики:

Абдулхаликова И.М., преподаватель ГБПОУ ДППК им.Г.Б.Казиахмедова

Мирзоева Д.М., преподаватель ГБПОУ ДППК им.Г.Б.Казиахмедова

Общие положения

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности Проектирование и разработка информационных систем и составляющих его профессиональных компетенций, а также общие компетенции, формирующиеся в процессе освоения ОПОП в целом.

Формой аттестации по профессиональному модулю является экзамен по модулю: форме выполнения комплексного практического задания.

В оценочных процедурах экзамена по модулю используется шкала оценки компетенций, фиксирующая факт достижения или не достижения студентами планируемых образовательных результатов по профессиональному модулю.

Итоговая оценка по профессиональному модулю отражается в сводной ведомости учета освоения профессионального модуля.

Формой аттестации по междисциплинарному курсу является экзамен.

1. Формы контроля и оценивания элементов профессионального модуля

Таблица 1.1

Элемент модуля	Форма контроля и оценивания	
	Промежуточная аттестация	Текущий контроль
МДК 05.01. Проектирование и дизайн информационных систем	5 семестр – дифференцированный зачет в форме выполнения практического задания. 6 семестр – экзамен в форме экзаменационных билетов.	Оценка выполнения практических работ, тестовых заданий
МДК. 05.02 Разработка кода информационных систем	5 семестр – дифференцированный зачет в форме выполнения теста 6 семестр – экзамен в форме экзаменационных билетов	Оценка выполнения практических работ, устных опросов, самостоятельных внеаудиторных работ.
МДК. 05.03 Тестирование информационных систем	7 семестр – другие формы контроля 8 семестр – экзамен в форме выполнения практического задания	Оценка выполнения практических работ, тестовых заданий, самостоятельных внеаудиторных работ.

УП.02.01 Проектирование и разработка информационных систем	5, 7, 8 семестр: дифференцированные зачеты в форме выполнения задания	Наблюдение и оценка выполнения работ при прохождении практики
ПП.02.01 Проектирование и разработка информационных систем	6, 7, 8 семестры: дифференцированные зачеты в форме защиты отчетов по практике	Экспертная оценка работодателем выполнения работ по заданию на практику и дневника практики.
ПМ.02.ЭК Экзамен по модулю Проектирование и разработка информационных систем	8 семестр - Экзамен по модулю	

2. Результаты освоения модуля, подлежащие проверке на экзамене по модулю.

2.1. В результате аттестации по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 2.1

Профессиональные и общие компетенции, которые возможно сгруппировать для проверки	Показатели оценки результата
<p>ПК 5.1 Собирать исходные данные для разработки проектной документации на информационную систему.</p> <p>ОК 1. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p> <p>ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.</p> <p>ОК 8. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.</p>	<ul style="list-style-type: none"> - производить сбор исходных данных для разработки проектной документации на информационную систему с использованием инструментальных средств - обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач. - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач. - демонстрировать грамотность устной и письменной речи. - ясность формулирования и изложения мыслей. - эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.

<p>ПК 5.2 Разрабатывать проектную документацию на разработку информационной системы в соответствии с требованиями заказчика.</p> <p>ОК 3. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<ul style="list-style-type: none"> - использовать инструментальные средства для разработки проектной документации на разработку информационной системы в соответствии с требованиями заказчика. - демонстрация ответственности за принятые решения. - обоснованность самоанализа и коррекция результатов собственной работы.
<p>ПК 5.3. Разрабатывать подсистемы безопасности информационной системы в соответствии с техническим заданием.</p>	<ul style="list-style-type: none"> - производить разработку подсистем безопасности
<p>ПК 5.4. Производить разработку модулей информационной системы в соответствии с техническим заданием.</p>	<ul style="list-style-type: none"> - производить разработку технической документации на эксплуатацию
<p>ПК 5.5. Осуществлять тестирование информационной системы на этапе опытной эксплуатации с фиксацией выявленных ошибок кодирования в разрабатываемых модулях информационной системы.</p>	<ul style="list-style-type: none"> - осуществлять тестирование информационной системы
<p>ПК 5.6 Разрабатывать техническую документацию на эксплуатацию информационной системы.</p> <p>ОК 7. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.</p> <p>ОК 9. Использовать информационные технологии в профессиональной деятельности.</p> <p>ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.</p>	<ul style="list-style-type: none"> - разработка документации по эксплуатации информационной системы. - эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик. - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности. - эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту. - эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.

<p>ПК 5.7 Производить оценку информационной системы для выявления возможности ее модернизации.</p> <p>ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p> <p>ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p> <p>ОК6.Проявлять гражданскопатриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.</p> <p>ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.</p>	<ul style="list-style-type: none"> - проведение оценки качества и экономической эффективности информационной системы и модификации отдельных модулей информационной системы. - использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач. - взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик. - обоснованность анализа работы членов команды (подчиненных). - соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик.
--	--

К дифференцированному зачету по междисциплинарному курсу допускаются обучающиеся, полностью выполнившие все практические работы/задания, и, имеющие положительные оценки по результатам текущего контроля.

К экзамену (квалификационному) по профессиональному модулю допускаются обучающиеся, успешно прошедшие промежуточную аттестацию по междисциплинарным курсам, учебной и производственной практикам в рамках данного профессионального модуля.

3. Оценка освоения теоретического курса профессионального модуля

3.1. Общие положения

Основной целью оценки теоретического курса профессионального модуля является оценка знаний и умений. Оценка теоретического курса профессионального модуля по каждому МДК осуществляется с использованием следующих форм и методов контроля:

1. Текущий контроль: оценка выполнения практических работ, тестовых заданий, самостоятельных внеаудиторных работ.
2. Промежуточная аттестация: экзамены по билетам, дифференцированные зачеты.

Оценка теоретического курса профессионального модуля предусматривает использование накопительной системы оценивания.

3.2. Задания для оценки освоения МДК:

3.2.1. Задания для текущей аттестации по МДК 05.01 Проектирование и дизайн информационных систем

Тема 1. Основы проектирования информационных систем

ПРАКТИЧЕСКАЯ РАБОТА №1

Анализ предметной области методом контент-анализа

Цель: Изучение особенностей контент-анализа и его этапов. Выполнение индивидуального проекта.

Коды формируемых компетенций: ОК 2, ОК5, ПК 5.1., ЛР 4, ЛР 9.

Формируемые умения и навыки: У 1, У 2, У 3; З 1.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задание 1

Найдите архив газеты «Известия», например на портале <https://100.iz.ru/rubric/122>.

Задание 2 Проанализируйте

одну из следующих тем:

- 1) Городской транспорт (1990-е гг, 2000-е гг., 2010-е гг.)
- 2) Животные в городе (1990-е гг, 2000-е гг., 2010 е гг.)
- 3) Городская связь (1990-е гг, 2000-е гг., 2010 е гг.)
- 4) Дети в городе (1990-е гг, 2000-е гг., 2010-е гг.)
- 5) Городское благоустройство (1990-е гг, 2000-е гг., 2010-е гг.)
- 6) Достопримечательности (1990-е гг, 2000-е гг., 2010-е гг.)
- 7) Выборы в прессе (1990-е гг, 2000-е гг., 2010-е гг.)

- 8) Сеть здравоохранения (1990-е гг, 2000-е гг., 2010-е гг.)
- 9) Обсуждение властей (1990-е гг, 2000-е гг., 2010-е гг.)
- 10) Городской досуг (1990-е гг, 2000-е гг., 2010-е гг.)
- 11) Кинотеатры в городе (1990-е гг, 2000-е гг., 2010-е гг.)
- 12) Образование (1990-е гг, 2000-е гг., 2010-е гг.)
- 13) VIP-персоны (1990-е гг, 2000-е гг., 2010-е гг.)
- 14) Общественное питание (1990-е гг, 2000-е гг., 2010-е гг.)
- 15) Городская «Индустрия красоты» (1990-е гг, 2000-е гг., 2010-е гг.)
- 16) Учебная городская сеть (1990-е гг, 2000-е гг., 2010-е гг.)
- 17) Городской спорт (1990-е гг, 2000-е гг., 2010-е гг.)

Задание 3

Заполнить таблицу

Время	Название статьи	Автор	Количество смысловых единиц	Количество единиц счета	Индикатор

Сделайте вывод по проведенному анализу.

Критерии выставления оценки за практическую работу:

Оценка	Полная, системность, прочность знаний,	Обобщенность
«5» (отлично)	Четкость, правильность и аккуратность в заполнении таблицы. Вывод по работе изложен лаконично, грамотно. Владение профессиональной терминологией.	Ответ должен быть самостоятельным
«4» (хорошо)	Заполнены все поля таблицы. Вывод по работе изложен лаконично, грамотно. Владение терминологией.	Ответ может быть самостоятельным, с подсказками преподавателя
«3» (удовлетворительно)	Нет четкости в изложении вопроса и ответа.	Ответ не самостоятельный, с наводящими вопросами преподавателя
«2» (неудовлетворительно)	Заполнение таблицы неполное, бессистемное. Отсутствие знаний профессиональной терминологии. Не сделан вывод по работе.	Отсутствует порядок в проведении анализа

Разработка модели архитектуры информационной системы

Цель: Разработать модель IDEF0 в системе Ramus Educational.

Коды формируемых компетенций: ОК 1, ОК 2, ОК 9, ПК 5.1., ЛР 4, ЛР 9.

Формируемые умения и навыки: У 4, З 2, З 3.

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

1. Запустите программу Ramus Educational.
2. В окне мастера проекта укажите следующие характеристики:

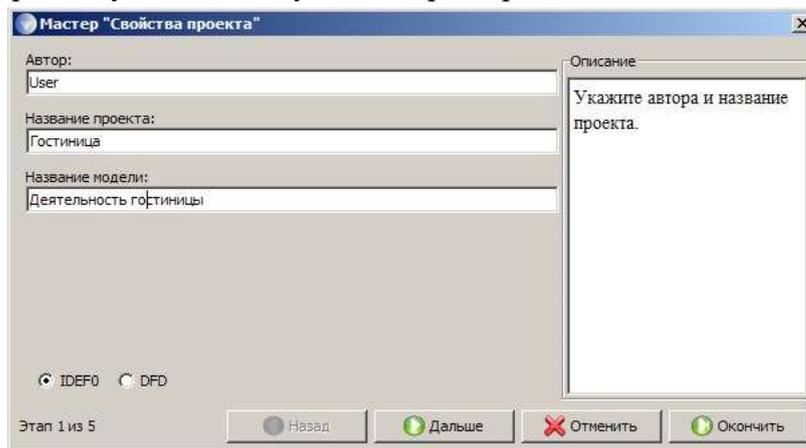


Рис. 1. Окно мастера создания проекта. Шаг 1

- При необходимости можно завершить работу мастера, нажав кнопку «Окончить». 3. Сохраните созданную модель, выбрав опцию меню «Файл» «Сохранить как».

2. Создание контекстной диаграммы

1. На панели инструментов выберите пиктограмму функции и мышью укажите месторасположение на рабочем пространстве.
2. Дайте данному функциональному блоку имя «**Описать деятельность гостиницы**».
3. Используя пиктограмму панели инструментов, создайте стрелки на контекстной диаграмме согласно Таблице 1.

Таблица 1. Контекстная диаграмма

Наименование стрелки	Тип
Заявки на заселение в гостиницу	вход
Материально-техническое и кадровое гостиницы обеспечение работы	вход
Нормативно-правовая база	управление
Гостиничные услуги	выход
Отчетность	выход
Финансовые результаты деятельности	выход
Услуги контрагентов	механизм
Инфраструктура гостиницы	механизм

4. В результате должна получиться контекстная диаграмма, показанная на Рис. 2.
- 2.

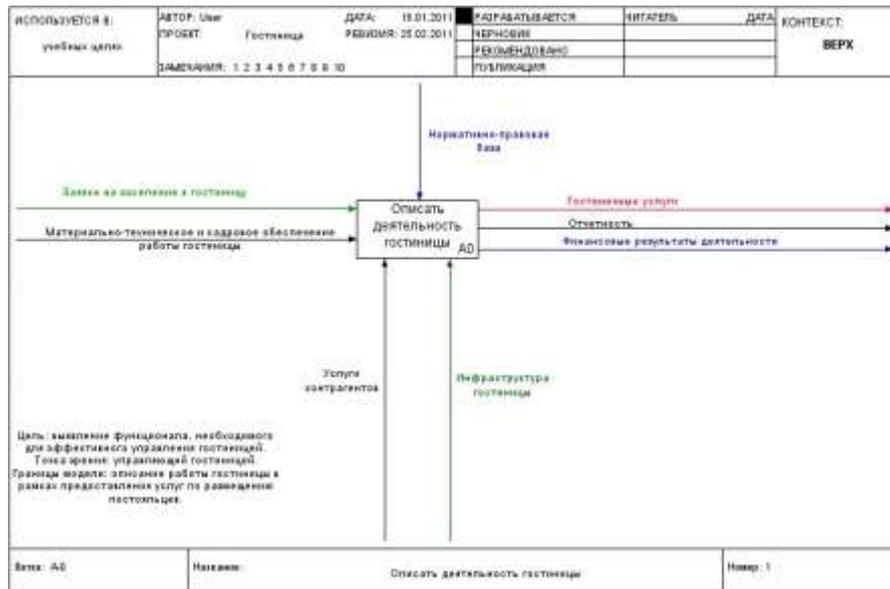


Рис. 2. Контекстная диаграмма предметной области «Гостиница»

5. Измените цвет стрелок согласно Рис. 2.

3. Создание диаграммы декомпозиции

1. Выберите в палитре инструментов кнопку перехода на нижний уровень, в диалоговом окне «Создание новой диаграммы» установите количество функциональных блоков **3**, укажите тип диаграммы (**IDEF0**).
2. Автоматически будет создана диаграмма первого уровня декомпозиции с перенесенными в нее потоками родительской диаграммы. Детализируйте диаграмму согласно Рис. 3.

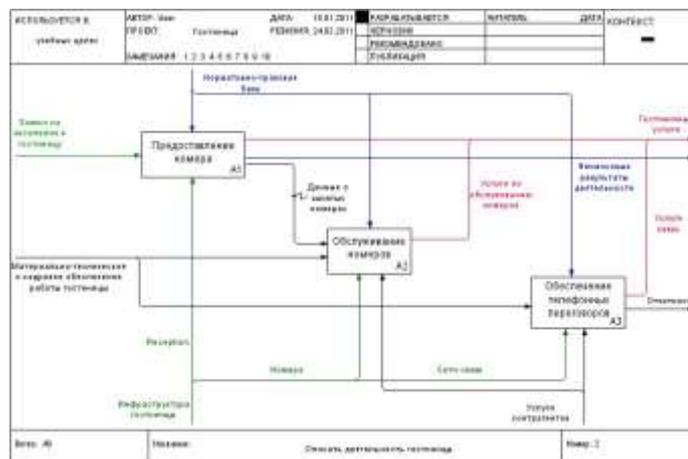


Рис. 3. Детализированная диаграмма первого уровня

3. Осуществите построение детализирующих диаграмм для функциональных блоков согласно представленной иерархии (см. Рис. 4).



Рис. 4. Иерархия функциональных блоков модели

4. В результате должны быть разработаны IDEF0-диаграммы, представленные на Рис. 5 - Рис. 9.

5. В программе Ramus Educational предусмотрена возможность экспорта разработанных диаграмм в виде рисунков формата *.png, *.bmp или *.jpeg.

Для этого в главном меню необходимо выбрать команду «Диаграммы» «Экспортировать как рисунки». В появившемся окне указывается список экспортируемых рисунков, выбирается их формат и размер, а также путь для сохранения (см. Рис. 10).

6. Покажите преподавателю разработанные диаграммы

Критерии оценки выполнения практической работы:

«Отлично» - самостоятельное выполнение практической работы; работа соответствует заданию и выполнена в полном объеме; оформлена в соответствии с рекомендациями, отсутствуют грамматические ошибки.

«Хорошо» - самостоятельное выполнение практической; работа соответствует заданию и выполнена в полном объеме; присутствуют недочеты в оформлении и грамматические ошибки.

«Удовлетворительно» - выполнение практической работы не соответствует заданию и выполнена не в полном объеме; присутствуют недочеты в оформлении и грамматические ошибки. Отсутствие экспорта диаграмм.

«Неудовлетворительно» - выполнение практической работы не соответствует заданию, присутствуют недочеты в оформлении и грамматические ошибки. Отсутствие экспорта диаграмм.

Тема 2. Система обеспечения качества информационных систем

Компьютерное тестирование осуществляется на платформе ИЦ «Академия»

Не зашифровано | pkgn/do.php?kurs=Mdk0501R

Панель инструментов

МДК 05.01 Проектирование и дизайн информационных систем (Белова Любовь Викторовна)

Учебники

1. Информационные системы (доступен с 2020-09-01 до 2020-12-30)
2. Проектирование информационных систем (доступен с 2020-09-01 до 2020-12-30)
3. Гресул. Проектирование информационных систем (доступен с 2020-09-01 до 2020-12-30)
4. Основы построения автоматизированных информационных систем (доступен с 2020-09-01 до 2020-12-30)

Тема 1. Основы проектирования информационных систем

5. Автоматизированные системы: основные понятия (доступен с 2020-09-01 до 2020-12-30)
6. Жизненный цикл ИС (доступен с 2020-09-03 до 2020-12-30)
7. Модели жизненного цикла ПО ИС (доступен с 2020-09-03 до 2020-12-30)
8. Жизненный цикл ИС. Модели ЖЦ ИС (презентация) (доступен с 2020-09-04 до 2020-12-30)
9. Методы сбора информации для анализа предметной области (доступен с 2020-09-08 до 2020-12-30)
10. Методы анализа предметных областей (доступен с 2020-09-08 до 2020-12-30)
11. Практическая работа №1 (доступен с 2020-09-10 до 2020-12-30)
12. Системный анализ (доступен с 2020-09-09 до 2020-12-30)
13. Системный анализ (презентация 1) (доступен с 2020-09-09 до 2020-12-30)
14. Системный анализ (презентация 2) (доступен с 2020-09-09 до 2020-12-30)
15. Системный анализ (презентация 3) (доступен с 2020-09-09 до 2020-12-30)
16. Пример структурной модели предметной области (доступен с 2020-09-11 до 2020-12-30)
17. Пример структурной модели предметной области (Презентация) (доступен с 2020-09-11 до 2020-12-30)
18. Практическая работа №2 (доступен с 2020-09-16 до 2020-12-30)
19. Постановка задачи обработки информации (доступен с 2020-09-17 до 2020-12-30)
20. Практическая работа №3 (доступен с 2020-09-18 до 2020-12-30)
21. Модели деятельности организации. ТЗ (доступен с 2020-09-21 до 2020-12-30)
22. Связь между процессами ЖЦ ПО. Организация коллективной разработки ПО (доступен с 2020-09-21 до 2020-12-30)
23. Методологии моделирования предметной области (доступен с 2020-09-21 до 2020-12-30)
24. Лабораторная работа №1 (доступен с 2020-09-23 до 2020-12-30)
25. Структурный подход к проектированию и разработке ПО (доступен с 2020-09-29 до 2020-12-30)

Профессиональный колледж г.Новокузнецка

Разделы

- Дисциплины
- Дисциплины ДО
- Журнал
- Расписание занятий
- Почта
- ЭЛЕКТРОННЫЙ КАБИНЕТ
- АКАДЕМИЯ-МЕДИА**

Администрирование

- Авторизация
- Пользователи и группы
- Редактор тестов
- Портфолио
- Файловый архив
- Заявка техникуму
- ВПК

Авторизация

Текущий пользователь: Love

Выход

ИИЦ Кемеровской области | ГОУ НПО Профессиональный колледж г. Новокузнецка

Личный кабинет | Белова Любовь Викторовна

Электронное обучение

Система электронного обучения для очного образования

Коммуникация

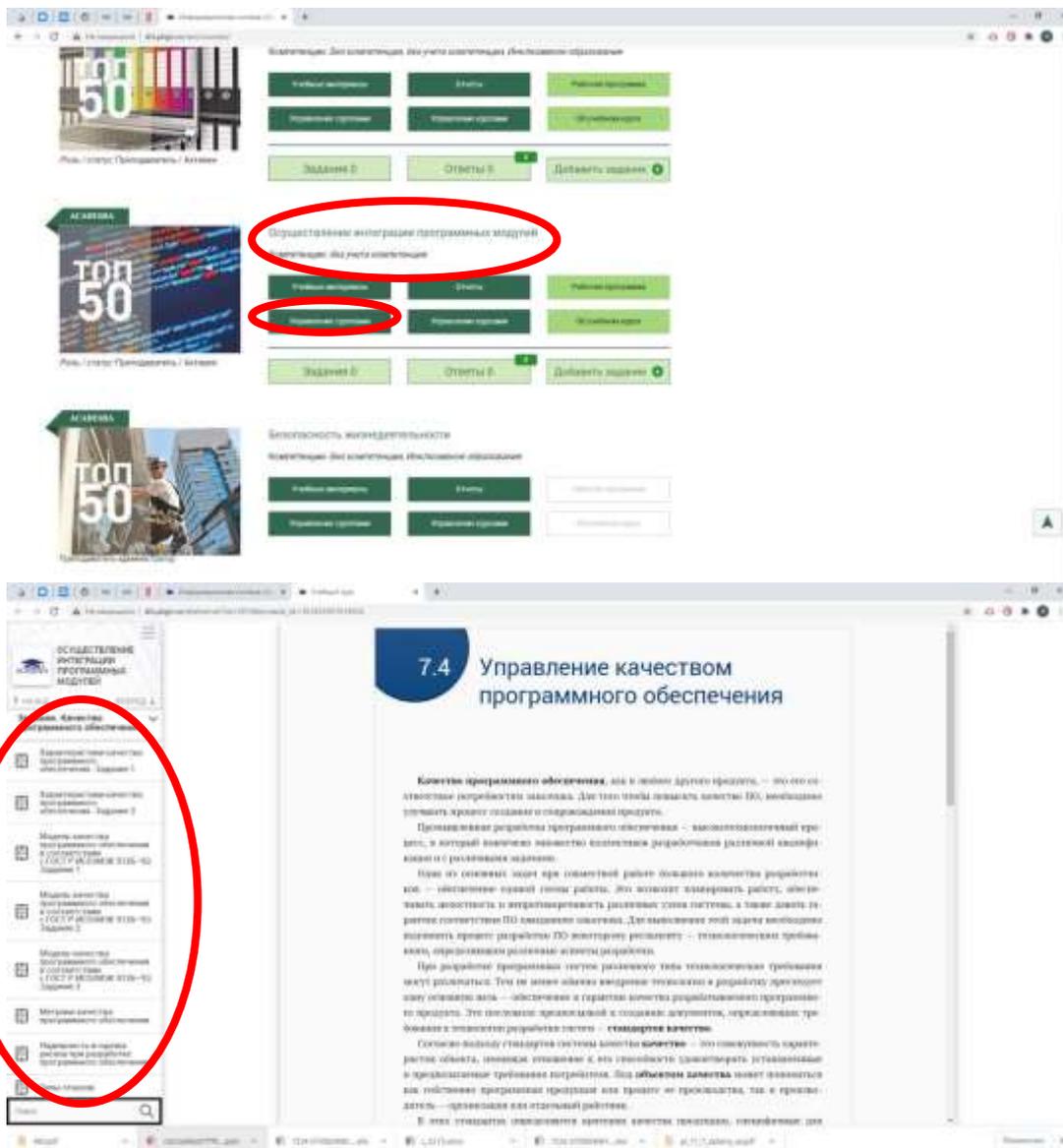
Надзорные ресурсы в области информатизации

Результаты

Дипломы

Электронный архив

Технологии электронного обучения и информатизации



Задания. Качество программного обеспечения. Пройти все тесты

Критерии оценки выполнения тестовых заданий:

Открыть отчет о результатах работы с материалами курса ИЦ «Академия».

Группы	Статьи	Качество выполнения задания 1	Качество выполнения задания 2	Модель качества программного обеспечения в соответствии с требованиями ГОСТ Р ИСО/МЭК 9126-01, Баллы/количество заданий 1	Модель качества программного обеспечения в соответствии с требованиями ГОСТ Р ИСО/МЭК 9126-01, Баллы/количество заданий 2	Модель качества программного обеспечения в соответствии с требованиями ГОСТ Р ИСО/МЭК 9126-01, Баллы/количество заданий 3	Метрики качества программного обеспечения	Надежность и оценка деталей программы и программного обеспечения	Тема статьи
Группа 1	Введение Демонстр. Скриншот	100%	100%	100%	100%	100%	100%	100%	100%
Группа 2	Введение Выводы Скриншот	80%	100%	100%	100%	100%	100%	100%	100%
Группа 3	Введение Демонстр. Скриншот	100%	100%	100%	100%	100%	100%	100%	100%

Рассчитать для каждого студента средний балл за выполненные 8 тестов.

Критерии оценки выполнения тестовых заданий:

- «Отлично» - 90-100% ср. балл;
- «Хорошо» - 80-90% ср. балл;
- «Удовлетворительно» - 50-80% ср. балл;
- «Неудовлетворительно» - менее 50% ср. балл.

Тема 3 Разработка документации информационных систем

ПРАКТИЧЕСКАЯ РАБОТА №14

Разработка руководства пользователя программного средства по индивидуальному заданию

Цель: разработать руководство пользователя, пользуясь специализированным программным обеспечением по индивидуальному заданию.

Коды формируемых компетенций: ОК 1, ОК 2, ОК 9, ПК 5.6., ЛР 4, ЛР 9.
Формируемые умения: У 1, У 3.

Порядок выполнения работы

1. Выберите тему, по которой будет создано руководство.
2. Создайте не менее 9 файлов в формате .htm с использованием редактора по выбранной теме.
3. Запустите программу HTML Help Workshop и с помощью мастера создайте новый проект.
4. Выберите и добавьте в проект созданные htm файлы.
5. Создайте контекстный файл проекта и добавьте статьи в проект.
6. Настройте иерархию для подразделов и статей.
7. Создайте указатель справочного файла.

8. Скомпилируйте проект.
9. По результатам практической работы составьте отчет.
10. Отправьте получившиеся отчет и файл справки преподавателю. **Требования к содержанию и оформлению отчета**

Отчет должен быть продемонстрирован в электронной форме с приложением файла справки, полученного в ходе выполнения лабораторной работы. Руководство должно содержать как минимум одну основную папку и 3 вложенные в нее папки с файлами. Также руководство должно содержать указатель, в котором с помощью ключевых слов можно найти искомые статьи.

В отчете должна присутствовать иерархическая структура справочного файла.

Отчет должен также содержать вывод относительно проделанной работы. В выводе не пересказывают этапы проведенной работы, а кратко излагают результаты. Например:

- какие файлы получены;
- есть ли ошибки или недостатки в полученных файлах.

Титульный лист отчета должен содержать всю информацию, необходимую для однозначной идентификации авторов и работы.

Для этого на титульном листе указывают название МДК, тему работы, номер группы, фамилия и инициалы студента.

Критерии оценки работы

Практическая работа считается выполненной на 5 (отлично) в том случае, если студент:

1. Выполнил все указанные задания, следуя порядку выполнения работы, освоил методику выполнения типовых заданий и способен продемонстрировать работу программы;
2. По результатам выполнения работы составил отчет, содержащий всю необходимую информацию;
3. Отправил отчет и получившийся файл руководства преподавателю.

Практическая работа считается выполненной на 4 (хорошо) в том случае, если студент:

1. Выполнил все указанные задания, следуя порядку выполнения работы, освоил методику выполнения типовых заданий и способен продемонстрировать работу программы;
2. По результатам выполнения работы составил отчет, содержащий не полную (не верную) информацию;
3. Отправил отчет и получившийся файл руководства преподавателю.

Практическая работа считается выполненной на 3 (удовлетворительно) в том случае, если студент:

1. Выполнил не все указанные задания, следуя порядку выполнения работы, не в полной мере освоил методику выполнения типовых заданий, но способен продемонстрировать работу программы;
2. По результатам выполнения работы составил отчет, содержащий ошибки и недочеты;
3. Создал отчет, но не смог отправить его и получившийся файл руководства преподавателю.

3.2.2 Задания для текущей аттестации по МДК 05.02 Проектирование и разработка информационных систем

Тема 1. Основные инструменты для создания, исполнения и управления информационной системой

1. Структура CASE-средства. Структура среды разработки. Основные возможности.
2. Основные инструменты среды для создания, исполнения и управления информационной системой. Выбор средств обработки информации
3. Организация работы в команде разработчиков. Система контроля версий: совместимость, установка, настройка
4. Обеспечение кроссплатформенности информационной системы
5. Сервисно - ориентированные архитектуры.
6. Интегрированные среды разработки для создания независимых программ.
7. Особенности объектно-ориентированных и структурных языков программирования.
8. Разработка сценариев с помощью специализированных языков

Тема 2. Разработка и модификация информационных систем

1. Обоснование и осуществление выбора модели построения или модификации информационной системы.
2. Обоснование и осуществление выбора средства построения информационной системы и программных средств.
3. Построение архитектуры проекта. Шаблон проекта
4. Определение конфигурации информационной системы. Выбор технических средств.
5. Формирование репозитория проекта, определение уровня доступа в системе контроля версий. Распределение ролей
6. Настройки среды разработки
7. Мониторинг разработки проекта. Сохранение версий проекта
8. Требования к интерфейсу пользователя. Принципы создания графического пользовательского интерфейса (GUI).
9. Понятие спецификации языка программирования. Синтаксис языка программирования. Стиль программирования
10. Основные конструкции выбранного языка программирования. Описание переменных, организация ввода-вывода, реализация типовых алгоритмов
11. Создание сетевого сервера и сетевого клиента.
12. Разработка графического интерфейса пользователя.
13. Отладка приложений. Организация обработки исключений.
14. Виды, цели и уровни интеграции программных модулей.

15. Выбор источников и приемников данных, сопоставление объектов данных.
16. Транспортные протоколы. Стандарты форматирования сообщений.
17. Организация файлового ввода-вывода.
18. Процесс отладки. Отладочные классы. 19. Спецификация настроек типовой ИС.

Критерии оценки устного ответа:

«5» (отлично) Обучающийся в полном объеме ответил на все вопросы и дополнительные вопросы, поставленные преподавателем, умеет работать со всеми видами источников, проявив самостоятельность и знания межпредметного характера, применять принципы учебной дисциплины в жизни.

«4» (хорошо) Обучающийся раскрыл содержание вопросов, но в его ответе содержатся недочеты или одна не грубая ошибка; при ответе на поставленные вопросы имеются незначительные замечания и поправки со стороны преподавателя. Обучающийся может самостоятельно добывать знания, пользуясь различными источниками, имеет развитые практические умения, но необязательно их применять.

«3» (удовлетворительно) Обучающийся раскрыл более, чем на 50% содержание вопросов, но его ответ содержит недочеты или 2-3 негрубые ошибки, при ответе на поставленные вопросы преподаватель оказывал ему значительную помощь в виде наводящих вопросов.

Обучающийся знает только основные принципы, умеет добывать знания лишь из основных источников, частично сформированы знания и умения.

«2» (неудовлетворительно) Обучающийся раскрыл менее, чем на 50% содержание вопросов, его ответ содержит более двух грубых ошибок, при ответе на поставленные вопросы преподаватель оказывал ему постоянную помощь.

Обучающийся не умеет самостоятельно работать с источниками, не знает принципов учебной дисциплины, у него не сформированы знания и умения.

Примерный вариант тестового задания для текущего контроля:

1. Какой тип тестов отвечает за испытание системы на производительность?
 - a) Функциональные тесты
 - b) Регрессионные тесты
 - c) Нагрузочные тесты
 - d) Unit-тесты
 - e) Анализ исходного кода
 - f) Анализ утечек памяти
2. Поддержка технологий инструментарием тестирования – это
 - a) Возможность записи и воспроизведения тестовых скриптов
 - b) Тестовые вызовы элементов программного кода, выполненных в стандартах той или иной технологии

с) Поддержка выбора из набора графического интерфейса

3. Установите соответствие

1	Среда разработки уровня	а	Visual Source Safe, CVS
2	Инструменты планирования и управления процессом разработки	б	Borland StarTeam, Rational ClearQuest
3	Инструменты планирования документооборота и управления ошибками, конфигурациями	в	Visual Studio, Delphi
4	Средства централизованного хранения и изменения данных	г	Microsoft Project Manager, DevPartner, Rational Unified Process

4. Пакет сопроводительной информации является

- а) Документальной поддержкой
- б) Технической поддержкой

5. Обработка исключительных ситуаций - это

- а) Обработка исключений
- б) Обработка сбоев внутри виртуальной машины
- с) Обработка ошибок ввода/вывода
- д) Обработка сбоев внутри компьютера

6. Укажите способы выявления и устранения программных сбоев

- а) Анализ журнала событий
- б) Исправление системного реестра
- с) Замена драйвера
- д) Переустановка программы
- е) Апгрейд программы
- ф) Запуск в режиме совместимости
- г) Изменение прав доступа
- х) Перезапуск программы
- и) Замена браузера

7. Сведение результатов автоматических и ручных тестов в системе управления качеством позволяет всем участникам проекта

- а) Видеть уровень качества очередного релиза
- б) Повысить производительность тест-дизайнеров и тестировщиков
- с) Контролировать изменение релиза
- д) Опирается на данную информацию при планировании своей работы

8. Тестовые сценарии используются для проведения

- а) Функционального тестирования
- б) Приемочного тестирования
- с) Нагрузочного тестирования
- д) Стресс-тестирования
- е) Исследовательского тестирования

f) Smoke-тестирования

9. Стандартные и альтернативные сценарии работы с приложением входят в

- a) Тестовую документацию
- b) Тестовый сценарий
- c) Контроль качества
- d) Тест-план

10. Что не входит в тест-план?

- a) Тестирование очередной версии программного продукта
- b) Тестирование безопасности
- c) Конфигурационное тестирование
- d) Тестирование комфортного использования
- e) Тестирование развертывания

11. Автоматизация тестирования позволяет

- a) Отказаться от ручного тестирования
- b) Высвободить ресурсы для проведения более сложных видов тестирования
- c) Снизить количество дефектов, доходящих до стадии контроля качества
- d) Ускорить выпуск релизов

12. Моделирование бизнес-процессов осуществляется средствами

- a) CRM системы
- b) ERP системы
- c) ARIS
- d) DFD

13. Что скрывается за данным сочетанием V&V?

- a) Проверка (верификация)
- b) Аттестация

c) Оба процесса 14. Какие процессы обеспечивают нахождение дефектов?

- a) SQL
- b) SQM
- c) SQMT
- d) SQA

15. Процессы подтверждения качества, которые исследуют и оценивают программный продукт на предмет: трассируемости, согласованности, полноты/завершенности, корректности, выполнения требований, - называются

Критерии оценивания:

Оценка	Критерии, %
«Отлично» - 5	90-100
«Хорошо» - 4	80-89
«Удовлетворительно» - 3	71-79
«Неудовлетворительно» - 2	0-70

3.2.2. Рубежный контроль

Контрольно-оценочные материалы для оценки освоения умений и знаний при текущем контроле

(приводятся типовые задания, перечисленные в таблице выше)

Выполнение практических работ при проведении практических занятий направлено на проверку умений и сформированности компетенций (элемента компетенций). В текущем контроле оценивается правильность выполнения заданий по теме и степень самостоятельности обучающегося при выполнении заданий. При проведении практических занятий может быть проведена деловая или ролевая игра. Деловая и/или ролевая игра – совместная деятельность группы обучающихся и преподавателя под управлением преподавателя с целью решения учебных и 12 профессионально-ориентированных задач путем игрового моделирования реальной проблемной ситуации. Позволяет оценивать умение анализировать и решать типичные профессиональные задачи, а также уровень сформированности компетенций (элемента компетенций)

Лабораторные задания для рубежного контроля по МДК. 05.02 Разработка кода информационных систем

Тема 1. Основные инструменты для создания, исполнения и управления информационной системой

ЛАБОРАТОРНАЯ РАБОТА №1. ПОСТРОЕНИЕ ДИАГРАММЫ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ И ДИАГРАММЫ ПОСЛЕДОВАТЕЛЬНОСТИ И ГЕНЕРАЦИЯ КОДА

Цель: ознакомиться с методологией моделирования информационных систем на основе языка UML.

Теоретические вопросы

Универсальный язык моделирования UML.

Понятие диаграммы.

Виды диаграмм.

Основные элементы диаграммы вариантов использования. Основные элементы диаграммы последовательности.

Задание № 1. Ознакомиться с методологией построения диаграммы вариантов использования основе языка UML.

Задание № 2. Проанализируйте пример построения диаграммы вариантов использования.

Пример. Магазин видеопродукции

Магазин продает видеокассеты, DVD-диски, аудиокассеты, CD-диски и т.д., а также предлагает широкой публике прокат видеокассет и DVD-дисков.

Товары поставляются несколькими поставщиками. Каждая партия товара предварительно заказывается магазином у некоторого поставщика и доставляется после оплаты счета. Вновь поступивший товар маркируется, заносится в базу данных и затем распределяется в торговый зал или прокат.

Видеоносители выдаются в прокат на срок от 1 до 7 дней. При прокате с клиента взимается залоговая стоимость видеоносителя. При возврате видеоносителя возвращается залоговая стоимость минус сумма за прокат. Если возврат задержан менее чем на 2 дня, взимается штраф в размере суммы за прокат за 1 день* кол-во дней задержки. При задержке возврата более чем на 2 дня – залоговая сумма не возвращается. Клиент может взять одновременно до 4 видеоносителей (прокат-заказ). На каждый видеоноситель оформляется квитанция.

Клиенты могут стать членами видео-клуба и получить пластиковые карточки. С членов клуба не берется залог (за исключением случая описанного ниже), устанавливается скидка на ставку проката и покупку товаров. Члены клуба могут делать предварительные заказы на подбор видеоматериалов для проката или покупки.

Каждый член клуба имеет некоторый статус. Первоначально – "новичок". При возврате всрок 5 прокат-заказов, статус меняется на "надежный". При задержке хотя бы одного видеоносителя более чем на 2 дня, статус "новичок" или "надежный" меняется на "ненадежный" и клиенту высылается предупреждение. При повторном нарушении правил статус меняется на "нарушитель". Члены клуба со статусом "надежный" могут брать до 8 видеоносителей одновременно, все остальные – 4. С членов клуба со статусом "нарушитель" берется залоговая сумма.

Клиент – покупает(прецедент "Продажа видео"), берет на прокат и возвращает видеоносители (прецеденты "Прокат видео" и "Возврат видео"), вступает в клуб (прецедент "Сопровождение клиентов"), задает вопросы (прецедент "Запрос сведений").

Последние два субъекта Поставщик и Клиент не будут иметь непосредственного доступа к разрабатываемой системе (второстепенные субъекты), однако именно они являются основным источником событий, инициализирующих прецеденты, и получателями результата работы прецедентов

От прецедента "Прокат видео" к прецеденту "Предупреждения" установлено отношение включения на том основании, что каждый выданный видеоноситель должен быть проверен на своевременный возврат и, в случае необходимости, выдано предупреждение клиенту.

Дальнейшее развитие модели поведения системы предполагает спецификацию прецедентов. Для этого традиционно используют два способа. Первый – описание с помощью текстового документа. Такой документ описывает, что должна делать система, когда субъект инициировал прецедент. Типичное описание содержит следующие разделы:

- краткое описание;
- участвующие субъекты;
- предусловия, необходимые для инициирования прецедента;
- поток событий (основной и, возможно, подпотоки, альтернативный);
- постусловия, определяющие состояние системы, по достижении которого прецедент завершается.

Описательная спецификация прецедента "Прокат видео"

Раздел	Описание
Краткое описание	Клиент желает взять на прокат видеокассету или диск, которые снимаются с полки магазина или были предварительно зарезервированы клиентом. При условии, что у клиента нет невозвращенных в срок видеоносителей, сразу после внесения платы фильм выдается напрокат. Если невозвращенные в срок видеоносители есть, клиенту выдается напоминание о просроченном возврате
Субъекты	Продавец, Клиент

Предусловия	В наличие имеются видеокассеты или диски, которые можно взять напрокат. У клиентов есть клубные карточки. Устройство сканирования работает правильно. Работники за прилавком знают, как обращаться с системой
Основной поток	<p>Клиент может назвать номер заказа или взять видеоноситель с полки. Видеоноситель и членская карточка сканируются, и продавцу не сообщается никаких сведений о задержках, так, что он не задает клиенту соответствующих вопросов. Если клиент имеет статус <надежный>, он может взять до 8 видеоносителей, во всех остальных случаях – до 4-х. Если статус клиента определен как <нарушитель>, его просят внести задаток. Клиент расплачивается наличными или кредитной картой. После получения суммы, информация о наличии фильмов обновляется и видеоносители передаются клиенту вместе с квитанциями на прокат. О прокате каждого видеоносителя делается отдельная запись с указанием идентификационного номера клиента, даты проката, даты возврата, идентификационного номера продавца, полученной суммы.</p> <p>Прецедент генерирует предупреждения о просроченном возврате клиенту, если видеофильм не был возвращен в течение двух дней по истечении даты возврата и изменяет статус клиента на <ненадежный> (первое нарушение) или <нарушитель> (повторное нарушение)</p>
Альтернативный поток	<p>У клиента нет членской карточки. В этом случае прецедент <Сопровождение клиента> может быть активизирован для выдачи новой карточки.</p> <p>Видеофильмы не выдаются, поскольку у клиента есть невозвращенные в срок видеоносители.</p> <p>Попытка взять напрокат слишком много видеоносителей. Видеоноситель или кредитная карта не могут быть отсканированы из-за их повреждения</p> <p>У клиента не хватило наличных или платеж по кредитной карте отклонен</p>
Постусловия	Видеофильмы сданы напрокат, и база данных соответствующим образом обновлена

Задание № 3. Постройте диаграмму вариантов использования для выбранной информационной системы (практическая работа № 11).

Задание № 4. Ознакомиться с методологией построения диаграммы последовательности основе языка UML.

Задание № 5. Проанализируйте пример построения диаграммы последовательности (рисунок 2).

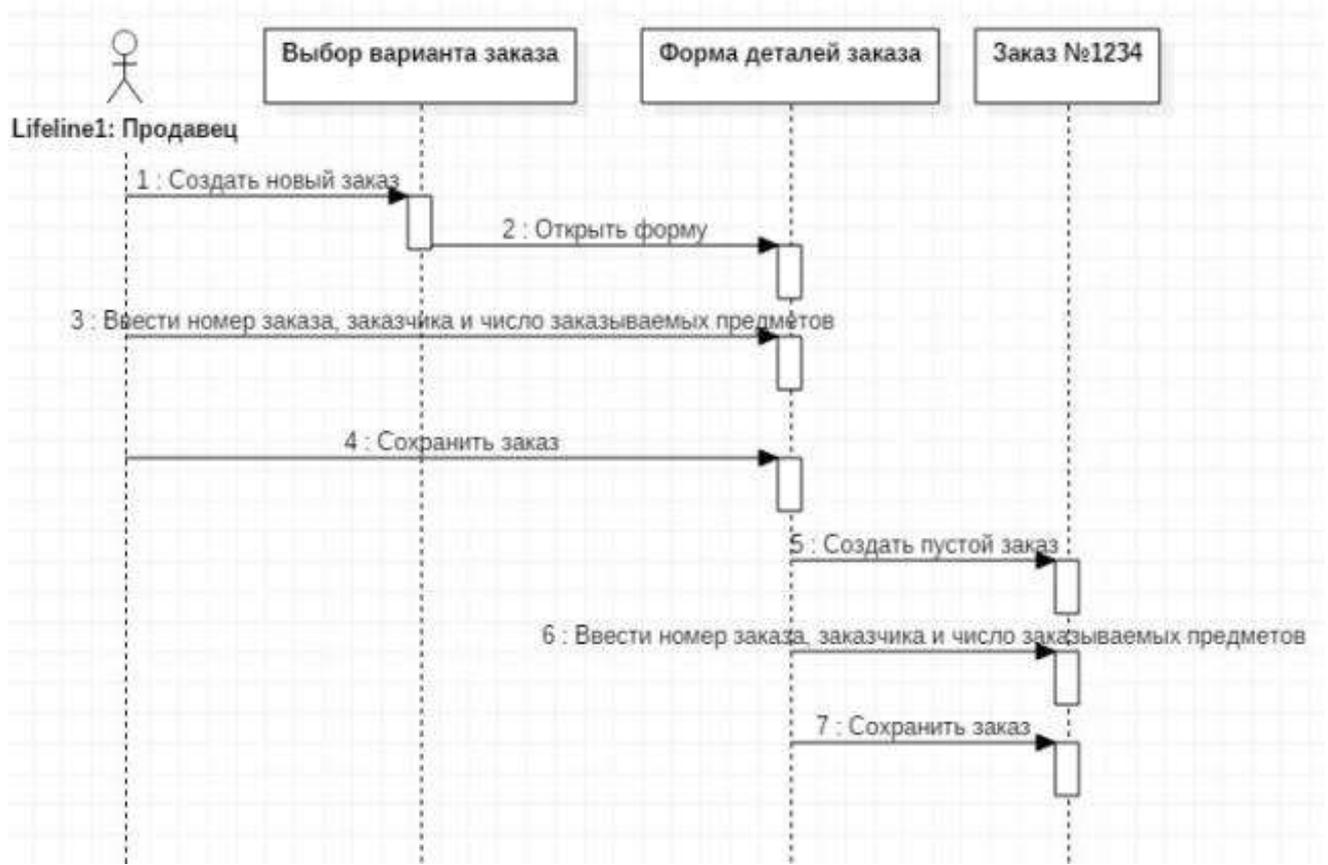


Рисунок 2

Пример Ввод
заказа.

Действующее лицо «Продавец».

Сообщения:

- создать новый заказ;
- открыть форму;
- ввести номер заказа, заказчика и число заказываемых предметов;
- сохранить заказ;
- создать пустой заказ;
- ввести номер заказа, заказчика и число заказываемых предметов; – сохранить заказ.

Теперь нужно позаботиться об управляющих объектах и о взаимодействии с базой данных. Как видно из диаграммы, объект Форма Деталей Заказа имеет множество ответственностей, с которыми лучше всего мог бы справиться управляющий объект. Кроме того, новый заказ должен сохранять себя в базе данных сам. Вероятно, эту обязанность лучше было бы переложить на другой объект.

Окончательный вид диаграммы последовательности представлен на рисунке 3.

Задание № 6. Постройте диаграмму последовательности для выбранной информационной системы (практическая работа № 11). Задание № 7. Оформите отчет.

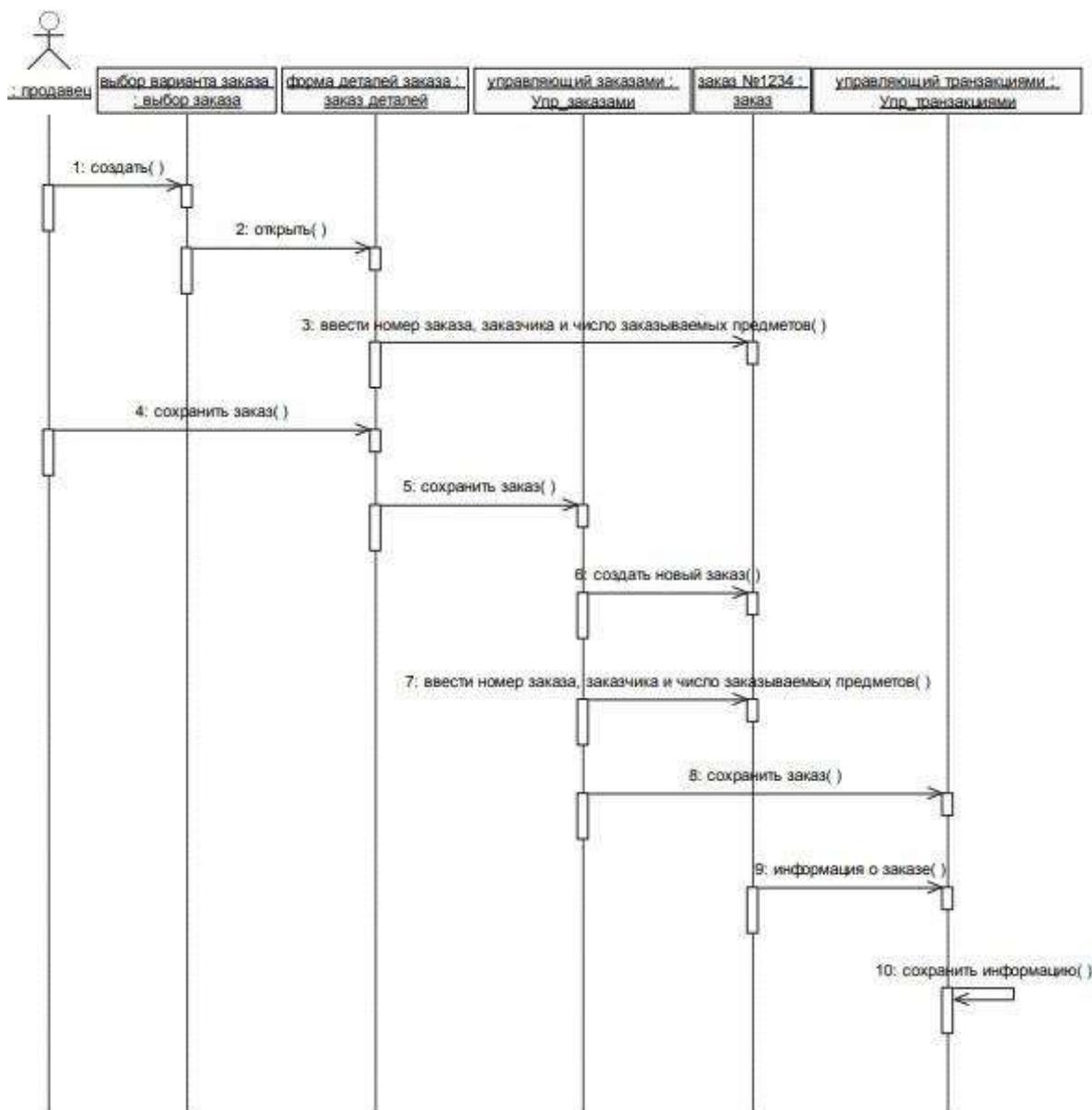


Рисунок 3

ЛАБОРАТОРНАЯ РАБОТА № 2. ПОСТРОЕНИЕ ДИАГРАММЫ КООПЕРАЦИИ И ДИАГРАММЫ РАЗВЕРТЫВАНИЯ И ГЕНЕРАЦИЯ КОДА

Цель: ознакомиться с методологией моделирования информационных систем на основе языка UML.

Теоретические вопросы

Универсальный язык моделирования UML.

Понятие диаграммы.

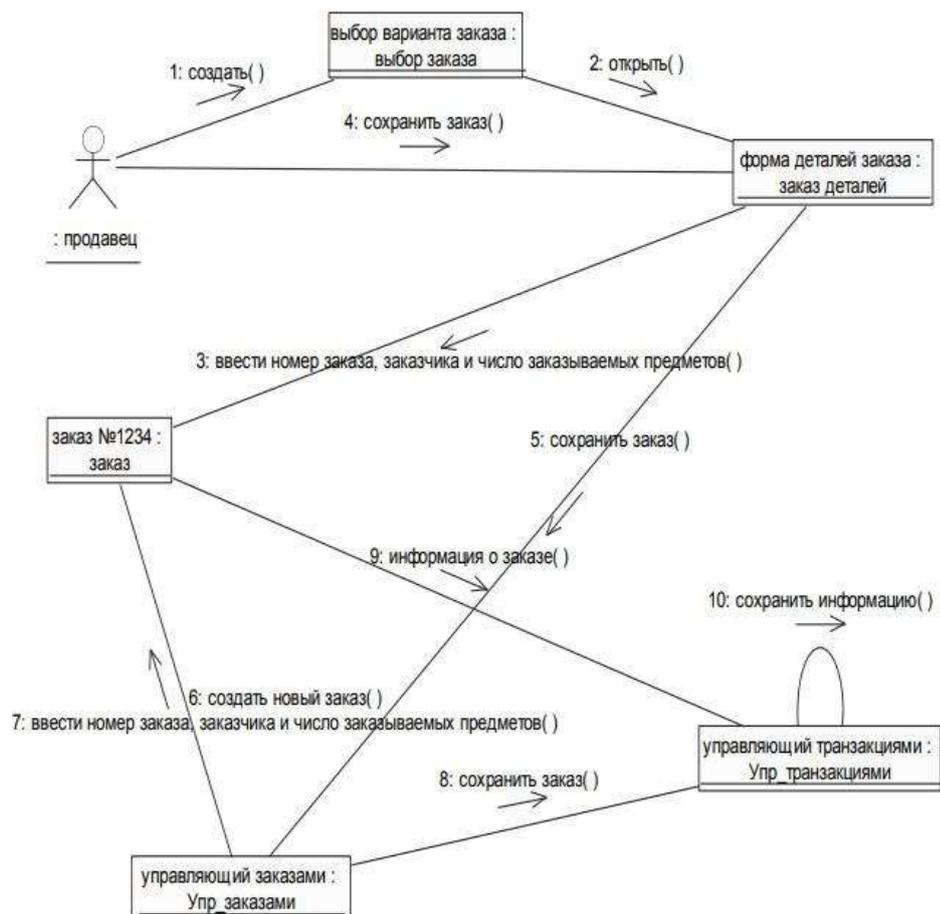
Виды диаграмм.

Основные элементы диаграммы кооперации.

Основные элементы диаграммы развертывания.

Задание № 1. Ознакомиться с методологией построения диаграммы кооперации основе языка UML.

Задание № 2. Проанализируйте пример построения диаграммы кооперации (рисунок 4).



Задание № 3. Постройте диаграмму кооперации для выбранной информационной системы (практическая работа № 11).

Задание № 4. Ознакомьтесь с методологией построения диаграммы развертывания основе языка UML.

Задание № 5. Проанализируйте пример построения диаграммы развертывания.

Примеры построения диаграмм развертывания

Фрагмент диаграммы развертывания с соединениями между узлами показан на рисунке 5.

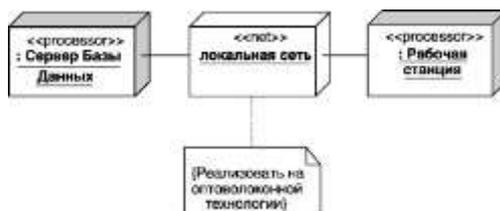


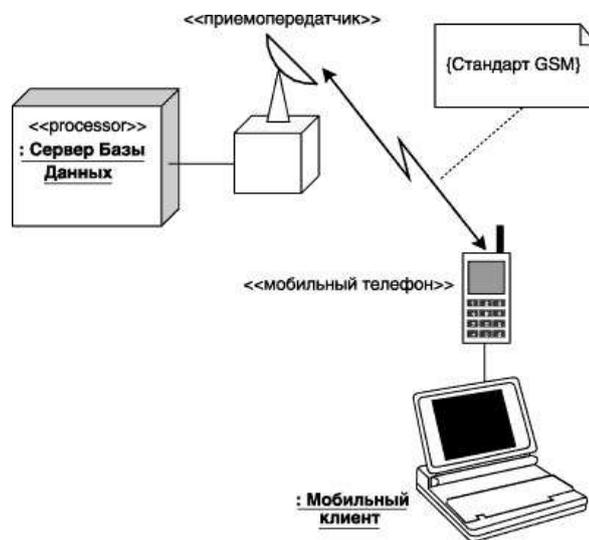
Рисунок 5

Диаграмма развертывания с отношением зависимости между узлом и развернутыми на нем компонентами приведена на рисунке 6.



Рисунок 6

Диаграмма развертывания для системы мобильного доступа к корпоративной базе данных изображена на рисунке 7.



Задание № 6. Постройте диаграмму развертывания для выбранной информационной системы (практическая работа №11). *Задание № 7.* Оформите отчет.

ЛАБОРАТОРНАЯ РАБОТА № 3. ПОСТРОЕНИЕ ДИАГРАММЫ КОМПОНЕНТОВ И ГЕНЕРАЦИЯ КОДА

Цель: ознакомиться с методологией моделирования информационных систем на основе языка UML.

Теоретические вопросы

Универсальный язык моделирования UML.

Понятие диаграммы.

Виды диаграмм.

Основные элементы диаграммы компонентов.

Основные элементы диаграммы развертывания.

Задание № 1. Ознакомьтесь с методологией построения диаграммы компонентов основе языка UML.

Задание № 2. Проанализируйте пример построения диаграммы компонентов.

Выделяем компоненты, отображаем зависимости между ними.

Фрагмент диаграммы компонентов с отношениями зависимости и реализации показан на рисунке 8.

Графическое изображение отношения зависимости между компонентами приведено на рисунке 9.

На рисунке 10 показано графическое изображение зависимости между компонентом и классами.

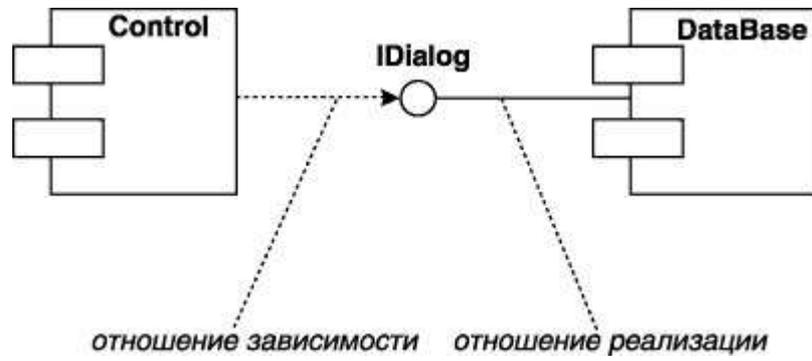


Рисунок 8

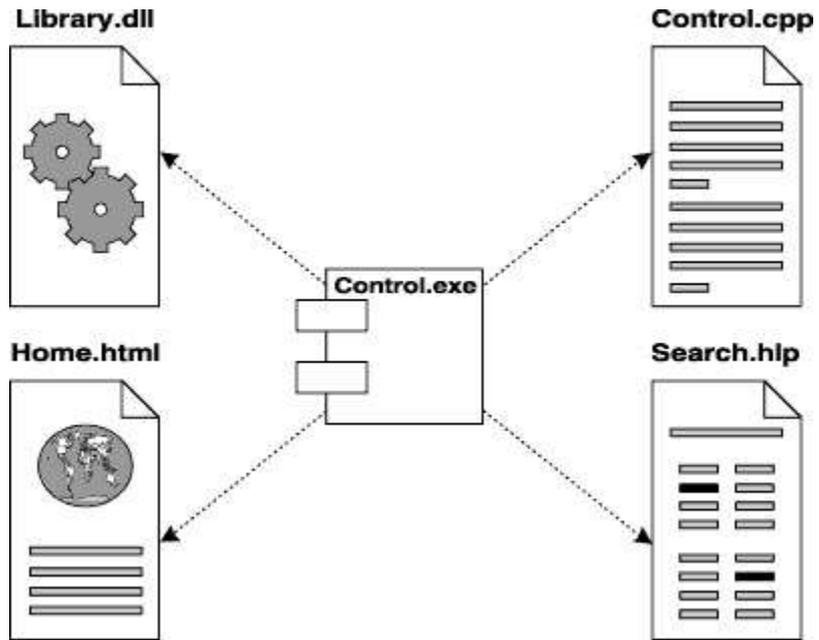


Рисунок 9

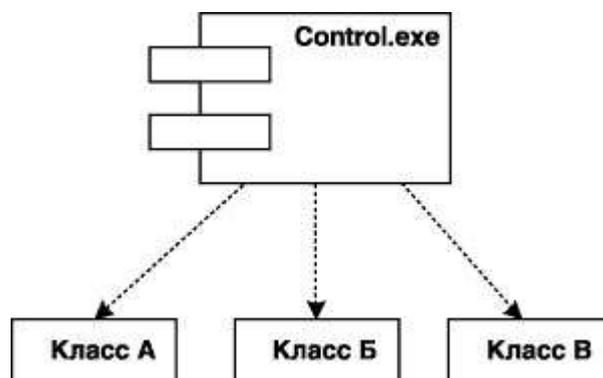


Рисунок 10

Задание № 3. Постройте диаграмму компонентов для выбранной информационной системы (практическая работа № 11).

Задание № 4. Оформите отчет.

ЛАБОРАТОРНАЯ РАБОТА № 4. ПОСТРОЕНИЕ ДИАГРАММ ПОТОКОВ ДАННЫХ И ГЕНЕРАЦИЯ КОДА

Цель: получение навыков построения диаграмм потоков данных.

Теоретические вопросы

Понятие диаграммы потоков данных. Элементы диаграммы потоков данных.

Хранилища данных.

Потоки управления.

Задание № 1. Ознакомиться с методологией построения диаграмм потоков данных.

Диаграммы потоков данных (Data Flow Diagrams – DFD) используются для описания движения документов и обработки информации как дополнение к IDEF0. В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, стрелки в DFD показывают лишь то, как объекты (включая данные) движутся от одной работы к другой. DFD отражает функциональные зависимости значений, вычисляемых в системе, включая входные значения, выходные значения и внутренние хранилища данных. DFD – это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах.

DFD содержит процессы, которые преобразуют данные, потоки данных, которые переносят данные, активные объекты, которые производят и потребляют данные, и хранилища данных, которые пассивно хранят данные. Диаграмма потоков данных содержит:

- процессы, которые преобразуют данные;
- потоки данных, переносящие данные;
- активные объекты, которые производят и потребляют данные;
- хранилища данных, которые пассивно хранят данные.

Процесс DFD преобразует значения данных и изображается в виде эллипса, внутри которого помещается имя процесса (рисунок 11).



Рисунок 11

Поток данных соединяет выход объекта (или процесса) с входом другого объекта (или процесса) и представляет собой промежуточные данные вычислений. Поток данных изображается в виде стрелки между производителем и потребителем данных, помеченной именами соответствующих данных. Дуги могут разветвляться или сливаться, что означает соответственно разделение потока данных на части либо слияние объектов.

Активным объектом является объект, который обеспечивает движение данных, поставляя или потребляя их. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа (рисунок 12).

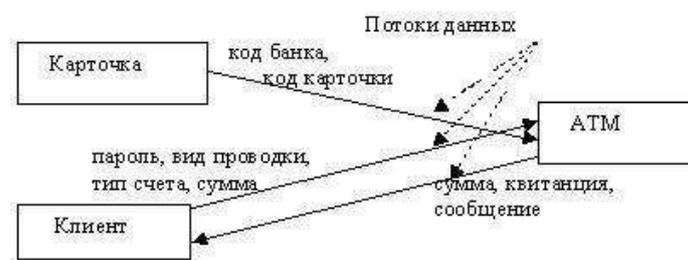


Рисунок 12

Хранилища данных. Хранилище данных – это пассивный объект в составе DFD, в котором данные сохраняются для последующего доступа. Хранилище данных допускает доступ к хранимым в нем данным в порядке, отличном от того, в котором они были туда помещены. Агрегатные хранилища данных, как, например, списки и таблицы, обеспечивают доступ к данным в порядке их поступления, либо по ключам (рисунок 13).

Потоки управления. DFD показывает все пути вычисления значений, но не показывает, в каком порядке значения вычисляются. Решения о порядке вычислений связаны с управлением программой, которое отражается в динамической модели. Эти решения, вырабатываемые специальными функциями, или предикатами, определяют, будет ли выполнен тот или иной процесс, но при этом не передают процессу никаких данных, так что их включение в функциональную модель необязательно. Тем не менее, иногда бывает полезно включать указанные предикаты в функциональную модель, чтобы в ней были отражены условия выполнения соответствующего процесса. Функция, принимающая решение о запуске процесса, будучи включенной в DFD, порождает в диаграмме поток управления и изображается пунктирной стрелкой (рисунок 14).

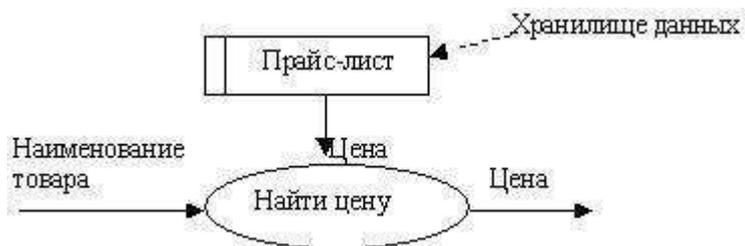


Рисунок 13

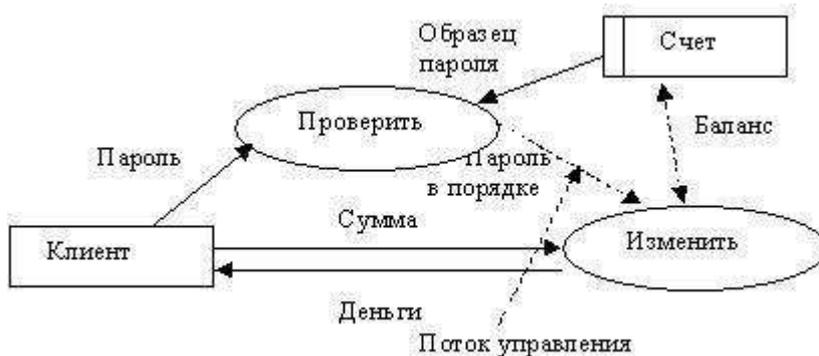


Рисунок 14

Первым шагом при построении иерархии DFD является построение контекстных диаграмм. Обычно при проектировании относительно простых информационных систем строится единственная контекстная диаграмма со звездобразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и, кроме того, главный единственный процесс не раскрывает структуры распределенной системы.

Для сложных информационных систем строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не главный единственный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип

Цель: получение навыков построения диаграмм потоков данных.

Теоретические вопросы

Понятие системы контроля версий (СКВ), решаемые задачи.

Основные понятия СКВ и их отношения: хранилище, commit, история, рабочая копия. Отличия централизованных и децентрализованных СКВ.

Примеры СКВ каждого вида. Действия с СКВ при единоличной работе с хранилищем.

Порядок работы с общим хранилищем в централизованной СКВ.

Задание № 1. Изучите систему контроля версий, установленную на компьютере (например, TortoiseSVN). При необходимости установите систему контроля версий TortoiseSVN. Опишите основные возможности системы контроля версий.

Задание № 2. Создайте новый проект. Создайте локальный репозиторий для своего проекта.

Удалите созданный проект на своем компьютере и обновите проект из репозитория.

Задание № 3. Внесите изменения в файлах с исходными кодами и сохраните изменения в репозитории. Обновите файлы с исходными кодами из репозитория. Внесите изменения в файлах с исходными кодами таким образом, чтобы у двух участников проекта изменения были в одном и том же файле. Попробуйте сохранить изменения в репозитории. Устраните обнаруженные конфликты версий. Повторно сохраните изменения в репозитории. Создайте отдельную ветку проекта. Внесите изменения в файлы с исходными кодами.

Задание № 4. Объедините созданную на предыдущем шаге ветку с основной веткой проекта. Выведите на экран данные изменений файла, в котором было наибольшее количество изменений. Отобразите на экране сравнение файла до и после внесения одного из изменений.

Задание № 5. Создайте репозиторий в сети Интернет. Удалите созданный проект на своем компьютере и обновите проект из репозитория. Внесите изменения в файлах с исходными кодами и сохраните изменения в репозитории. Обновите файлы с исходными кодами из репозитория. Внесите изменения в файлах с исходными кодами таким образом, чтобы у двух участников проекта изменения были в одном и том же файле. Попробуйте сохранить изменения в репозитории. Устраните обнаруженные конфликты версий. Повторно сохраните изменения в репозитории. Создайте отдельную ветку проекта. Внесите изменения в файлы с исходными кодами. *Задание № 6.* Оформите отчет.

ЛАБОРАТОРНАЯ РАБОТА № 6. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Цели: получение навыков проектирования и разработки интерфейса пользователя.

Теоретические вопросы

Понятие пользовательского интерфейса. Виды пользовательских интерфейсов.

Основные элементы пользовательского интерфейса.

Требования к разработке пользовательского интерфейса.

Задание № 1. Настроить среду разработки Visual Studio. Создать приложение для Windows, которое имитирует игровой автомат со «счастливыми» числами. Программа должна иметь следующий интерфейс (рисунок 16).

При нажатии на кнопку «Крутить» должны генерироваться три случайных числа от 0 до 9. Если хотя бы одно из них равно семи, на форме должны появляться надпись «Счастливая семерка» и картинка с изображением человека, платящего игроку деньги при выигрыше. При нажатии на кнопку «Выход» программа должна завершать работу. Решение сохранить под именем

«Игра». Создать исполняемый файл приложения.

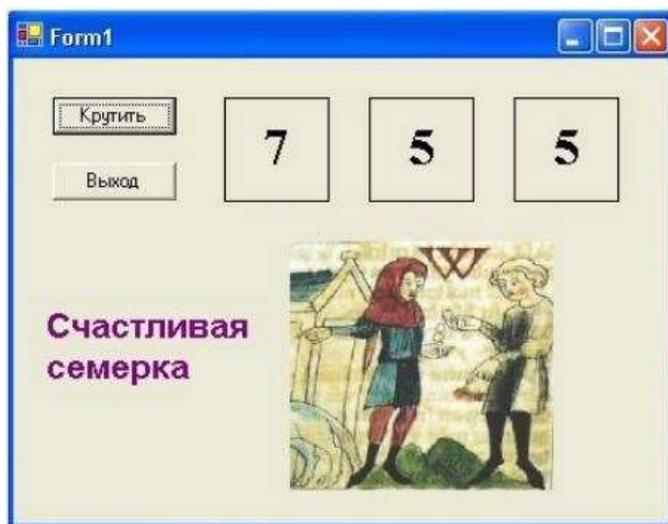


Рисунок 16

Задание № 2. Добавить в созданную форму метку и организовать отображение на ней процента выигрышей по отношению к общему числу нажатий на кнопку «Крутить».

Задание № 3. Добавить в программу оператор Randomize для того, чтобы программа при каждом запуске выдавала новую последовательность случайных чисел.

Задание № 4. Создать приложение для Windows «Продажи он-лайн», которое позволяет выбрать для заказа компьютер, офисную технику и периферийные устройства с выводом в форму изображения выбранного оборудования, указать способ оплаты и желаемую дату поставки. Возможные способы оплаты: рубли, доллары США, английские фунты. При выборе способа оплаты должно появляться его символическое изображение. Пользователь, выбрав товары для заказа, вводит название фирмы. Рекомендуемый интерфейс приложения приведен на рисунке 17.

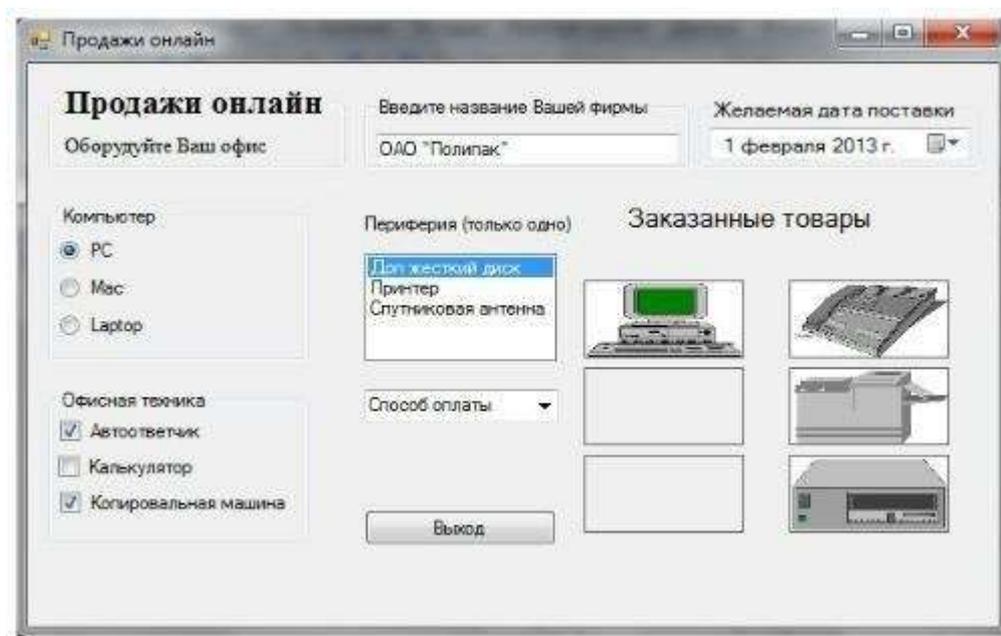


Рисунок 17 Решение сохранить под именем «Продажи». Создать исполняемый файл приложения.

Задание № 5. Добавить в список офисной техники «МФУ» и добавить еще один объект PictureBox для отображения рисунка МФУ. Соответствующим образом изменить программный код. Задание № 6. Добавить в способы оплаты «Чек».

ЛАБОРАТОРНАЯ РАБОТА № 7. РЕАЛИЗАЦИЯ АЛГОРИТМОВ ОБРАБОТКИ ЧИСЛОВЫХ ДАННЫХ. ОТЛАДКА ПРИЛОЖЕНИЯ

Цели: получение навыков реализации алгоритмов обработки числовых данных, отладки приложений.

Теоретические вопросы

Элементы управления, используемые для обработки числовых данных.

Задание № 1. Разработать приложение Windows, которое по заданным значениям: цены покупки, суммы первоначального платежа, годовой процентной ставки и срока кредита рассчитывает размер ежемесячных выплат по кредиту, а также строит схему

платежей за каждый период (месяц) с разделением на основные платежи и платежи по процентам. Рассчитать также сумму всех основных платежей (для контроля) и сумму платежей по процентам (размер переплаты). Рекомендуемый интерфейс приложения показан на рисунке 18.

Решение сохранить под именем «Платежи по кредиту».

Задание № 2. Внесите изменения в программный код так, чтобы в схеме платежей в 4-ом столбце отображалась общая сумма платежа за каждый период.

Задание № 3. Внесите изменения в форму и программный код так, чтобы платежи по кредиту осуществлялись не ежемесячно, а ежеквартально.

Задание № 4. Предусмотрите возможность пересмотра схемы платежей на оставшиеся периоды, если в некоторый период внесен платеж больше требуемой суммы. Рассмотреть такую схему погашения, при которой не уменьшается срок погашения кредита, а уменьшается сумма периодического платежа в последующих периодах.

Период	Основной платеж	Платеж по %
1	5565,04	3400,00
2	5639,24	3325,80
3	5714,43	3250,61
4	5790,63	3174,42
5	5867,83	3097,21
6	5946,07	3018,97
7	6025,35	2939,69
8	6105,69	2859,35

Рисунок 18

ЛАБОРАТОРНАЯ РАБОТА № 8. РЕАЛИЗАЦИЯ АЛГОРИТМОВ ПОИСКА. ОТЛАДКА ПРИЛОЖЕНИЯ

Цели: получение навыков реализации алгоритмов поиска данных, отладки приложений.

Теоретические вопросы

Алгоритмы поиска в тексте. Алгоритмы поиска в массивах.

Задание № 1. Написать программу «Результаты сессии», которая для выбранной из списка группы запрашивает ввод: – списка группы; – количества и названий предметов, по которым данная группа сдавала экзамены в последнюю сессию; – оценок студентов по предметам. Программа должна также:

- отображать результаты сессии по данной группе;
- вычислять качество знаний (процент студентов, успевающих на «хорошо» и «отлично»);
- вычислять процент успеваемости в группе (процент студентов, сдавших сессию);
- определять количество студентов, успевающих на «отлично».

Вычисление качества знаний, процента успеваемости и количества отличников оформить в виде соответствующих процедур – функций. По итогам сессии должна быть рассчитана стипендия. Размеры минимальной и повышенной стипендии вводятся с клавиатуры. Минимальную стипендию получают студенты, сдавшие сессию на «хорошо» и «отлично».

В программе должны быть созданы 3 формы: главная форма «Результаты сессии и расчет стипендии», форма для отображения результатов сессии и форма «Размер стипендии» (рисунки 19, 20, 21, 22, 23, 24, 25, 26).

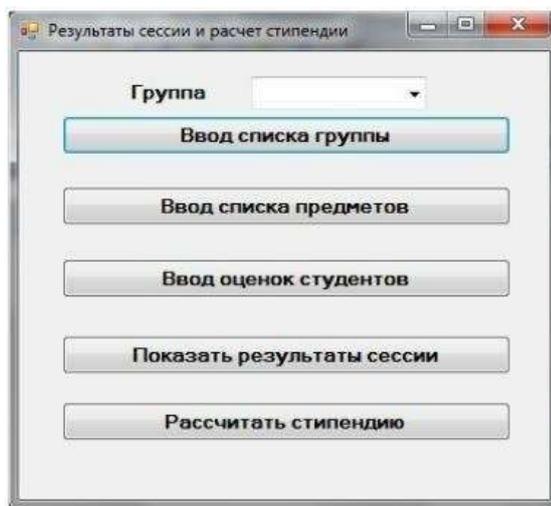


Рисунок 19

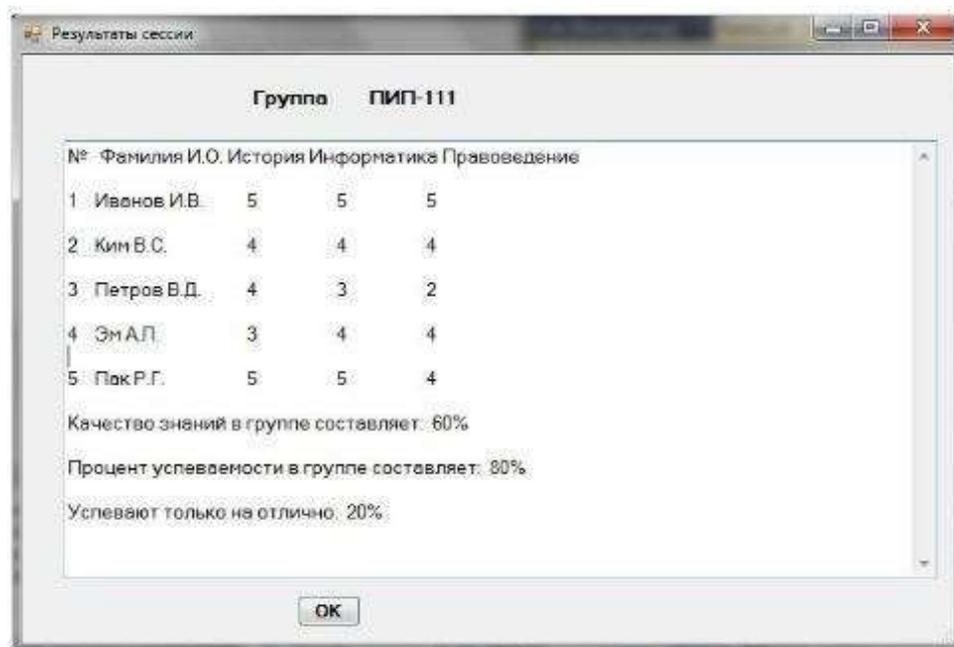


Рисунок 20

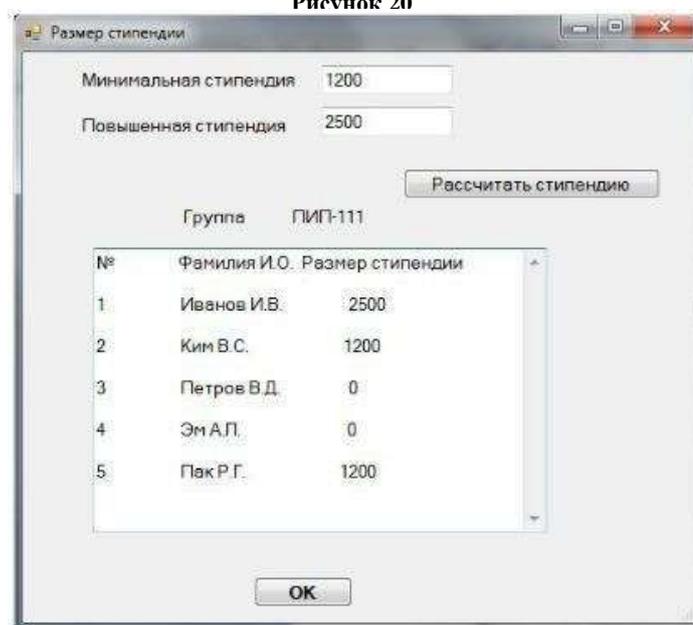


Рисунок 21

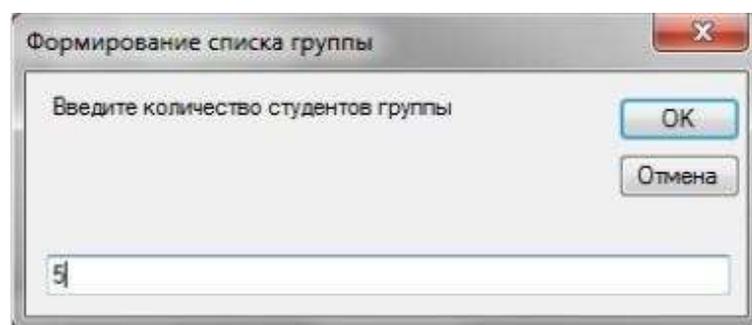


Рисунок 22

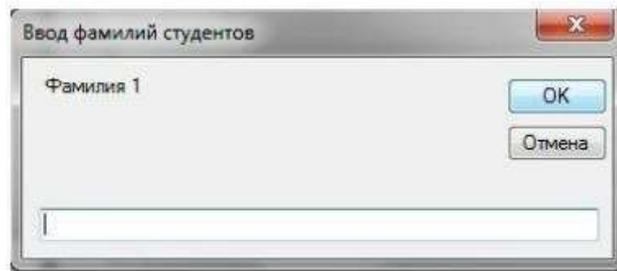


Рисунок 23

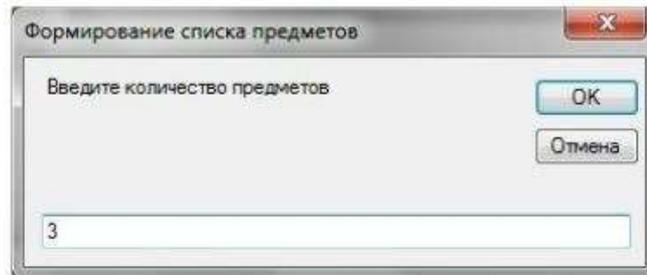


Рисунок 24

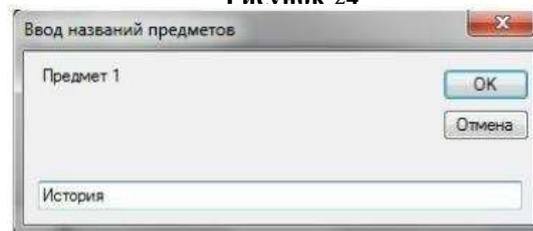


Рисунок 25

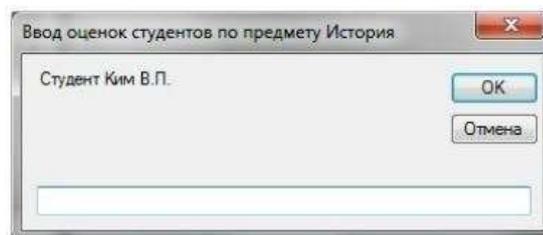


Рисунок 26

Задание № 2. Написать программы, иллюстрирующие применение методов линейного поиска, поиска делением пополам, а также различные методы сортировки массивов.

ЛАБОРАТОРНАЯ РАБОТА № 9. РЕАЛИЗАЦИЯ ОБРАБОТКИ ТАБЛИЧНЫХ ДАННЫХ. ОТЛАДКА ПРИЛОЖЕНИЯ

Цель: получение навыков обработки табличных данных, отладки приложений.

Теоретические вопросы

Обработка табличных данных в приложениях.

Задание № 1. Организовать работу с базой данных Студенты, которая храниться в текстовом файле. При выборе в списке ComboBox определенной группы на форме

Списки групп отобразит в сетке данных DataGridView только фамилии студентов данной группы. Рекомендуемый интерфейс приложения изображен на рисунке 27.

Задание № 2. Создать запрос, который будет отбирать из базы данных Студенты фамилии студентов заданного курса, записывать их вместе с названием группы во временный файл СтудентыВрем и отображать на форме с помощью элемента DataGridView.

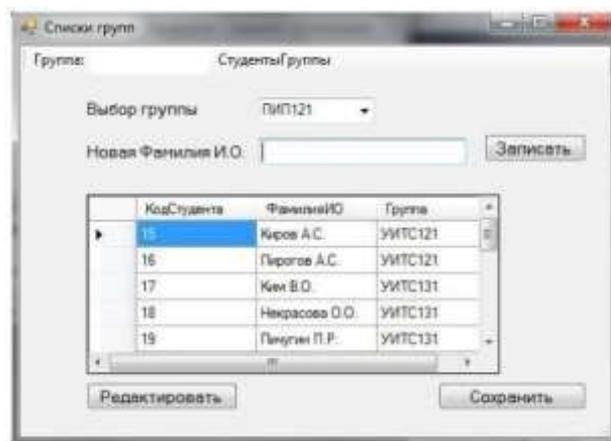


Рисунок 27

ЛАБОРАТОРНАЯ РАБОТА № 10. РАЗРАБОТКА И ОТЛАДКА ГЕНЕРАТОРА СЛУЧАЙНЫХ СИМВОЛОВ

Цель: получение навыков разработки и отладки генератора случайных символов.

Теоретические вопросы

Понятие генератора случайных символов. Управление генератором случайных символов.

Задание № 1. Разработать генератор случайных чисел.

Случайные числа в языке программирования C++ могут быть сгенерированы функцией rand() из стандартной библиотеки C++. Функция rand() генерирует числа в диапазоне от 0 до

RAND_MAX. RAND_MAX – это константа, определённая в библиотеке <cstdlib>. Для MVS RAND_MAX = 32767, но оно может быть и больше, в зависимости от компилятора. Ниже показана простая программка, использующая генератор случайных чисел rand():

```
#includ  
e  
"stdafx
```

```

.h"
#include
e
<iostre
am>
using
names
pace
std;

int main(int argc, char* argv[])
{
    cout << "RAND_MAX = " << RAND_MAX << endl; // константа, хранящая
максимальный предел из интервала случайных чисел cout << "random
    number = " << rand() << endl; // запуск генератора случайных чисел
    system("pause");
    return 0;
}

```

Максимальное случайное число в примере – это 32767. Зачастую, нам не нужен такой большой диапазон чисел от 0 до RAND_MAX. Например, в игре «Наперстки» необходимо отгадать, под каким из трёх напёрстков спрятан шарик, то есть генерация чисел должна выполняться в пределе от 1 до 3-х. Бросая монету, может возникнуть только два случая, когда монета упадёт «орлом» или «решкой» вверх, нужный интервал – от 1 до 2. Возникает потребность в масштабировании интервала генерации случайных чисел. Для того чтобы масштабировать интервал генерации чисел нужно воспользоваться, операцией нахождения остатка от деления «%»: // пример масштабирования диапазона генерации случайных чисел rand() % 3 + 1 // диапазон равен от 1 до 3 включительно

Число 3 является масштабируемым коэффициентом. То есть, какое бы не выдал число генератор случайных чисел rand() запись rand() % 3 в итоге выдаст число из диапазона от 0 до 2. Для того чтобы сместить диапазон, мы прибавляем единицу, тогда диапазон изменится на такой – от 1 до 3 включительно.

Задание № 2. Разработать программу, использующую масштабируемый генератор случайных чисел. Ниже показан код программы, которая несколько раз запускает функцию rand().

```
// rand_ost.cpp: определяет точку входа для консольного приложения.
```

```

#include
e
"stdafx

```

```

.h"
#include
e
<iostre
am>
using
names
pace
std;

int main(int argc, char* argv[])
{ cout << "1-random number = " << 1 + rand() % 3 << endl; // первый запуск
генератора случайных чисел cout << "2-random number = " << 1 + rand() % 3 << endl; //
второй запуск генератора
случайных чисел cout << "3-random number = " << 1 + rand() % 3 << endl; // третий запуск
генератора
случайных чисел cout << "4-random number = " << 1 + rand() % 3 << endl; //
четвёртый запуск
генератора случайных чисел cout << "5-random number = " << 1 + rand() % 3 << endl; //
пятый запуск генератора
случайных чисел cout << "6-random number = " << 1 + rand() % 3 << endl; // шестой запуск
генератора
случайных чисел cout << "7-random number = " << 1 + rand() % 3 << endl; //
седьмой запуск
генератора случайных чисел cout << "8-random number = " << 1 + rand() % 3 <<
endl; // восьмой запуск
генератора случайных
чисел sys te
m(
"p aus
e")
;
ret
urn
0;
}

```

При повторном запуске программы, печатаются те же самые числа. Суть в том, что функция rand() один раз генерирует случайные числа, а при последующих запусках

программы всего лишь отображает сгенерированные первый раз числа. Такая особенность функции `rand()` нужна для того, чтобы можно было правильно отладить разрабатываемую программу. При отладке программы, внося какие-то изменения, необходимо удостовериться, что программа срабатывает правильно, а это возможно, если входные данные остались те же, то есть сгенерированные числа. Когда программа успешно отлажена, нужно, чтобы при каждом выполнении программы генерировались случайные числа. Для этого нужно воспользоваться функцией `srand()` из стандартной библиотеки C++. Функция `srand()` получив целый положительный аргумент типа `unsigned` или `unsigned int` (без знаковое целое) выполняет рандомизацию, таким образом, чтобы при каждом запуске программы функция `srand()` генерировала случайные числа. Программа, использующая функцию `srand()` для рандомизации генератора случайных чисел `rand()`:

```
// srand.cpp: определяет точку входа для консольного приложения.

#include
e
"stdafx
.h"
#include
e
<iostre
am>
using
names
pace
std;

int main(int argc, char* argv[])
{
    unsigned rand_value = 11; srand(rand_value); //
    рандомизация генератора случайных чисел
    cout << "rand_value = " << rand_value << endl;
    cout << "1-random number = " << 1 + rand() % 10 << endl; // первый запуск
генератора случайных чисел cout << "2-random number = " << 1 + rand() % 10
    << endl; // второй запуск
генератора случайных
чисел sys te
m(
```

```
"pause")
;
return
0;
}
```

Задание № 3. Разработать обобщённый пример использования автоматического генератора случайных чисел с масштабированием. Пример работы программы:

```
// srand_time.cpp: определяет точку входа для консольного
приложения. #include "stdafx.h"
#include <iostream>
#include
<ctime>
using
namespace
std;

int main(int argc, char* argv[])
{
    srand( time( 0 ) ); // автоматическая рандомизация
    cout << "rand_value = "
    << 1 + rand() % 10 << endl;
    system("pause");
    return 0; }
```

Теперь при каждом срабатывании программы будут генерироваться совершенно случайные числа в интервале от 1 до 10, включительно.

Задание № 4. Разработать генератор случайных символов. Сформировать случайную символьную последовательность.

ЛАБОРАТОРНАЯ РАБОТА № 11. ИНТЕГРАЦИЯ МОДУЛЯ В ИНФОРМАЦИОННУЮ СИСТЕМУ

Цель: получение навыков интеграции модулей в информационную систему.

Теоретические вопросы

Понятие модуля.

Управление

модулями

.

Задание № 1. Создать файл, содержащий сведения о сдаче студентами сессии. Структура записи: индекс группы, фамилия студента с его инициалами, оценки по четырем экзаменам и пяти зачетам («з» означает зачет, «н» – незачет). Экзамены и зачеты нумеровать цифрами. Количество записей в файле не менее двадцати.

При разработке приложения использовать стандартные модули.

Задание № 2. Разработать программу, интегрирующую модули из приложения, разработанного в рамках задания №1, выводящую следующую информацию:

- фамилии неуспевающих студентов с указанием индексов групп и вида задолженности;
- фамилии студентов, сдавших все зачеты и получившие на экзаменах четверки и пятерки; – средний бал, полученный каждым студентом.

ЛАБОРАТОРНАЯ РАБОТА № 12. РАЗРАБОТКА ПРИЛОЖЕНИЙ ДЛЯ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ И ЯВЛЕНИЙ. ОТЛАДКА ПРИЛОЖЕНИЯ

Цель: получение навыков разработки и отладки приложений для моделирования процессов и явлений. Теоретические вопросы

Понятие модели.

Моделирование процессов и явлений.

Технологии моделирования процессов и явлений в приложениях.

Задание № 1. Разработать физико-математическую модель системы при свободном падении физического тела, брошенного с высоты h и падающего свободно в течение t времени. При построении модели принять следующие гипотезы:

- 1) падение происходит в вакууме (то есть коэффициент сопротивления воздуха равен нулю);
- 2) ветра нет;
- 3) масса тела неизменна;
- 4) тело движется с одинаковым постоянным ускорением g в любой точке. Слово "модель" (лат. *modelium*) означает "мера", "способ", "сходство с какой-то вещью".

Проблема моделирования состоит из трех взаимосвязанных задач: построение новой (адаптация известной) модели; исследование модели (разработка метода исследования или адаптация, применение известного); использование (на практике или теоретически) модели.

Схема построения модели M системы S с входными сигналами X и выходными сигналами Y изображена на рисунке 28.

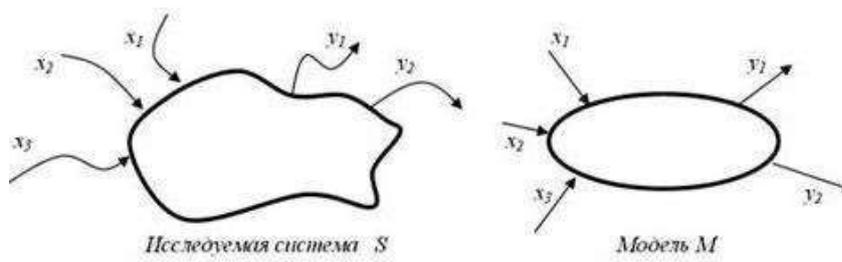


Рисунок 28

Если на вход M поступают сигналы из X и на выходе появляются сигналы из Y , то задан закон, правило f функционирования модели, системы.

Классификацию моделей проводят по различным критериям.

Модель – статическая, если среди параметров описания модели нет (явно) временного параметра.

Модель – динамическая, если среди параметров модели явно выделен временной параметр.

Модель – дискретная, если описывает поведение оригинала лишь дискретно, например, в дискретные моменты времени (для динамической модели).

Модель – непрерывная, если описывает поведение оригинала на всем промежутке времени.

Модель – детерминированная, если для каждой допустимой совокупности входных параметров она позволяет определять однозначно набор выходных параметров; в противном случае – модель недетерминированная, стохастическая (вероятностная).

Модель – функциональная, если представима системой функциональных соотношений (например, уравнений).

Модель – теоретико-множественная, если представима некоторыми множествами и отношениями их и их элементов.

Модель – логическая, если представима предикатами, логическими функциями и отношениями.

Модель – информационно-логическая, если она представима информацией о составных элементах, подмоделях, а также логическими отношениями между ними.

Модель – игровая, если она описывает, реализует некоторую игровую ситуацию между элементами (объектами и субъектами игры).

Модель – алгоритмическая, если она описана некоторым алгоритмом или комплексом алгоритмов, определяющим ее функционирование, развитие. Введение такого, на первый взгляд, непривычного типа моделей (действительно, кажется, что

любая модель может быть представлена алгоритмом ее исследования), на наш взгляд, вполне обосновано, так как не все модели могут быть исследованы или реализованы алгоритмически.

Модель – графовая, если она представима графом (отношениями вершин и соединяющих их ребер) или графами и отношениями между ними.

Модель – иерархическая (древовидная), если она представима иерархической структурой (деревом).

Модель – языковая, лингвистическая, если она представлена некоторым лингвистическим объектом, формализованной языковой системой или структурой. Иногда такие модели называют вербальными, синтаксическими и т.п.

Модель – визуальная, если она позволяет визуализировать отношения и связи моделируемой системы, особенно в динамике.

Модель – натурная, если она есть материальная копия оригинала.

Модель – геометрическая, если она представима геометрическими образами и отношениями между ними.

Модель – имитационная, если она построена для испытания или изучения, проигрывания возможных путей развития и поведения объекта путем варьирования некоторых или всех параметров модели.

Задание № 2. Разработать статическая модель движения тела по наклонной плоскости $F = am$. Динамическая модель типа закона Ньютона: $F(t) = a(t)m(t)$ или, еще более точно и лучше, $F(t) = s''(t)m(t)$. Если рассматривать только $t = 0.1, 0.2, \dots, 1$ (с), то модель $S_t = gt^2/2$ или числовая последовательность $S_0 = 0, S_1 = 0.01g/2, S_2 = 0.04g, \dots, S_{10} = g/2$ может служить дискретной моделью движения свободно падающего тела. Модель $S = gt^2/2, 0 < t < 10$ непрерывна на *промежутке времени (0;10)*.

Задание № 3. Разработать модель популяции рыб, из которой в текущий момент времени изымается некоторое количество особей (идет лов рыбы). Динамика такой системы определяется моделью вида: $x_{i+1} = x_i + ax_i - kx_i, x_0 = c$, где k – коэффициент вылова (скорость изъятия особей). Стоимость одной пойманной рыбы равна b руб. Цель моделирования – прогноз прибыли при заданной квоте вылова. Для этой модели можно проводить имитационные вычислительные эксперименты и далее модифицировать модель, например следующим образом.

Эксперимент 1. Для заданных параметров a, c изменяя параметр k , определить его наибольшее значение, при котором популяция не вымирает.

Эксперимент 2. Для заданных параметров c, k изменяя параметр a , определить его наибольшее значение, при котором популяция вымирает.

Модификация 1. Учитываем естественную гибель популяции (за счет нехватки пищи, например) с коэффициентом смертности, равным, b : $x_{i+1} = x_i + ax_i - (k + b)x_i, x_0 = c$.

Модификация 2. Учитываем зависимость коэффициента k от x (например, $k = dx$):

$$x_{i+1} = x_i + ax_i - dx_i^2, x_0 = c$$

ЛАБОРАТОРНАЯ РАБОТА № 13. ПРОГРАММИРОВАНИЕ ОБМЕНА СООБЩЕНИЯМИ МЕЖДУ МОДУЛЯМИ

Цели: получение навыков программирования обмена сообщениями между модулями.

Теоретические вопросы

Понятие и структура сообщения. Обмен сообщениями между модулями.

Задание № 1. Составить программу, помогающую сотрудникам Государственной инспекции безопасности дорожного движения (ГИБДД) обработать следующие данные: регистрационный номер автомобиля, марка автомобиля, цвет автомобиля, год выпуска, адрес владельца. Программа должна по требованию пользователя выдавать следующие сведения:

- адреса владельцев автомобилей заданной марки, определенного цвета;
- все данные об автомобиле с заданным регистрационным номером;
- все данные об автомобилях с известной цифровой частью регистрационного номера.

Задание № 2. Программу, разработанную в задании №1, разбить на модули. Например, создать такие модули, как главный (содержащий функцию `main()`), чтения из файла в массив структур, вывод на экран содержимого массива структур, сортировка данных (при необходимости), меню, формирование документов и т.д.

Задание № 3. Разработать схему межмодульных вызовов.

Задание № 4. Проанализировать способы передачи аргументов между функциями и целесообразность использования глобальных данных.

Критерии оценки выполнения лабораторных работ

Оценка	Критерий оценки
--------	-----------------

«5» (отлично)	обучающийся свободно применяет полученные знания при выполнении практических заданий; выполнил работу в полном объеме с соблюдением необходимой последовательности действий; в письменном отчете по работе правильно и аккуратно выполнены все записи; при ответах на контрольные вопросы правильно понимает их сущность, дает точное определение и истолкование основных понятий, использует специальную терминологию дисциплины, не затрудняется при ответах на видоизмененные вопросы, сопровождает ответ примерами.
«4» (хорошо)	выполнены требования к оценке «отлично», но допущены 3-5 недочетов при выполнении практических заданий и обучающийся может их исправить самостоятельно или при небольшой помощи преподавателя; в письменном отчете по работе делает незначительные ошибки; при ответах на контрольные вопросы не допускает серьезных ошибок, легко устраняет отдельные неточности, но затрудняется в применении знаний в новой ситуации, приведении примеров.
«3» (удовлетворительно)	практическая работа выполнена не полностью, но объем выполненной части позволяет получить правильные результаты и выводы; в ходе выполнения работы обучающийся продемонстрировал слабые практические навыки, были допущены ошибки; обучающийся умеет применять полученные знания при решении простых задач по готовому алгоритму; в письменном отчете по работе допущены ошибки; при ответах на контрольные вопросы правильно понимает их сущность, но в ответе имеются отдельные пробелы и при самостоятельном воспроизведении материала требует дополнительных и уточняющих вопросов преподавателя.
«2» (неудовлетворительно)	практическая работа выполнена не полностью и объем выполненной работы не позволяет сделать правильных выводов, у обучающегося имеются лишь отдельные представления об изученном материале, большая часть материала не усвоена; в письменном отчете по работе допущены грубые ошибки, либо он вообще отсутствует; на контрольные вопросы обучающийся не может дать ответов, так как не овладел основными знаниями и умениями в соответствии с требованиями программы.

3.2.3 Задания для текущей аттестации по МДК 05.03 Тестирование информационных систем

Тема 08.01.01 Основы web-технологий.

Лабораторная работа №1, 2, 3, 4, 5

**«Разработка тестового сценария проекта», «Разработка тестовых пакетов»
«Использование инструментария анализа качества», «Анализ и обеспечение
обработки исключительных ситуаций», «Функциональное тестирование»**

Цель работы – получения навыков создания тестовых сценариев на основе методики управления требованиями.

Задачи работы – освоение принципов применения программного продукта IBM Rational Requisite Pro для управления требованиями к программному обеспечению.

Теоретическая часть

Процесс создания тестовых сценариев включает в себя четыре шага:

1. Определение переменных для каждого шага сценариев использования.
2. Определение существенно разных вариантов для каждой переменной.
3. Комбинирование вариантов для тестирования в тестовые сценарии.
4. Определение значений переменных.

Первым делом нужно определить все входные переменные во всех шагах в представленном сценарии (алгоритме). Например, если на некотором шаге пользователь вводит свой идентификатор и пароль, это две переменных. Первая переменная – Идентификатор, вторая – Пароль. Переменной также может быть выбор, который пользователь должен сделать (такой как выбор рейса из списка).

Количество переменных может зависеть от введенных на предыдущем шаге значений.

На следующем шаге следует определить существенно различные варианты для каждой переменной. Варианты считаются «существенно разными», если они могут вызывать различное поведение системы.

Например, если выбрать Идентификатор, который предполагается быть длиной до 10ти символов, следующие приведенные значения будут существенно разными:

Alex – слишком короткий и мы ожидаем появления сообщения об ошибке. Alexandra – верный Идентификатор.

Alexandrena – слишком длинный, и мы ожидаем, что система не позволит нам вводить слишком длинный Идентификатор.

Однако, Alexandria или JohnGordon различны не существенно, т.к. оба являются верными идентификаторами, вызывающими одно поведение системы.

Вариант может считаться существенно другим, если:

- Он вызывает другой ход процесса (обычно альтернативный поток).

Пример: Ввод неверного пароля вызовет Альтернативный Поток 2.

Он вызывает другое сообщение об ошибке.

Пример: Если адрес электронной почты слишком длинный, сообщение должно быть: «E-mail должен быть не более 50-ти символов».

Пример: Если адрес электронной почты не содержит символа «@», сообщение должно быть: «Неверный E-mail адрес».

- Он вызывает другой вид пользовательского интерфейса.

Пример: Если кредитная карта была выбрана в качестве способа оплаты, система должна отображать экран с полями для ввода номера кредитной карты, даты окончания срока действия и название организации, обслуживающей карту.

- Он вызывает различный набор значений списков.

Пример: Экран регистрации пользователя должен содержать список Страна и Штат/Провинция. Список Штат/Провинция будет содержать значения на основе выбранной страны: для США он должен содержать все штаты, для Канады – все провинции, для других стран он должен быть недоступен.

– Он является условием ограничения.

Пример: Пароль должен быть не менее 6-ти символов.

В этом случае мы должны протестировать следующее: Пароль с пятью символами.

Пароль с шестью символами

– Что-то должно быть изменено вместо использования значения по умолчанию.

Пример: На экране оплаты кредитной картой название организации, обслуживающей кредитную карту, должно быть заполнено именем лица, размещающего заказ. Пользователь должен иметь возможность хранить это значение по умолчанию или ввести новое.

– Это создает два различных варианта:

Хранить установленное по умолчанию название организации, обслуживающей кредитную карту.

Изменить установленное по умолчанию название организации на другое.

– Формат ввода точно не определен и может быть интерпретирован разными способами.

Пример: поле ввода номера телефона должно принимать текст в свободной форме.

Номера телефонов разными лицами пишутся по-разному:

Использования скобок: (973) 123-4567 Использование

тире: 973-123-4567

Использование пробелов: 973 123 4567

Без пробелов: 9731234567

Все доступные варианты должны быть протестированы.

– Стандартные варианты могут быть разными для разных стран.

Формат даты срока окончания действия кредитной карты может быть разным для США и Европы.

На следующем шаге нужно скомбинировать их в последовательность шагов тестового сценария. Один из способов это сделать – создать Матрицу Распределения Тестовых Сценариев (Test Case Allocation Matrix). Строки этой матрицы содержат все переменные для всех шагов, требующие ввода данных от пользователя. Первая колонка содержит номер шага, вторая – название переменной, а остальные – тестовые сценарии. Их можно назвать T1, T2, и т.д. Нужно оценить, насколько много тестовых сценариев нужно для охвата данного сценария (алгоритма). Жестким вариантом оценки будет максимальное количество существенно различных вариантов, определенное для переменной. Не проблема, если при оценке будет допущена ошибка, т.к. можно добавить или удалить колонку при заполнении матрицы. Обычно типичный сценарий охватывают от пяти до семи тестовых сценариев. Тем не менее, иногда в особых случаях требуется больше тестовых сценариев.

На четвертом шаге следует заменить неопределенные варианты, такие как «очень длинная фамилия» или «длинный номер телефона с ext. (дополнительным номером)» на действительные значения, например «Georgiamistopolis» и «011-48 (242) 425-3456 ext.

1234» соответственно. На этом шаге также разделяют все тестовые сценарии из Матрицы Распределения Тестовых Сценариев, создавая отдельную таблицу для каждого тестового сценария.

Метод извлечения функциональных тестовых сценариев (test cases) из сценариев использования (use cases) имеет несколько преимуществ:

- Тестовые сценарии получаются с использованием более автоматического подхода.
- Уход от дублирования тестовых сценариев.
- Достигается больший охват тестового пространства.
- Легкость мониторинга тестового процесса.
- Легкость распределения работы между тестерами.
- Легкость регрессионного тестирования.
- Раннее обнаружение пропущенных требований.

Созданные тестовые сценарии могут быть использованы для ручного тестирования, также как и для автоматического тестирования с использованием таких инструментов, как IBM Rational Robot или IBM Rational Functional Tester.

Описание задачи

На основе сценариев использования предыдущей лабораторной работы построить тестовые сценарии.

Методика выполнения работы

А. Определение переменных для каждого шага сценариев использования

1. Переменные для основного потока сценария использования:

Переписать основной поток сценария «Бронирование билетов» с указанием используемых переменных (с префиксом V):

- V1. Турист вводит URL сайта.
- V2. Система отображает домашнюю страницу сайта.
- V3. Турист вводит информацию о полете:
 - V3.1. Дата вылета
 - V3.2. Время вылета
 - V3.3. Дата прибытия
 - V3.4. Время прибытия
 - V3.5. Число путешествующих взрослых
 - V3.6. Число путешествующих детей.
 - V3.7. Наличие путешествующих животных.
- Турист выбирает «Поиск рейсов».
- V4. Система отображает рейсы вылета, отсортированные по цене.
- V5. Турист выбирает рейс.
 - V5.1. Рейс вылета
- V6. Система отображает рейс прибытия.
- V7. Турист выбирает рейс прибытия.
 - V7.1. Рейс прибытия.

V8. Система отображает детали рейса.

V9. Турист подтверждает рейс.

V10. Пользователь предоставляет Идентификатор и Пароль для покупки билета.

V10.1. Идентификатор V10.2.

Пароль

V11. Турист предоставляет информацию пассажира.

V11.1. Фамилия. V11.2.

Имя.

V11.3. Отчество.

V11.4. Пол.

V11.5. Дата рождения.

V11.6. Номер паспорта.

V11.7. Серия паспорта.

V12. Система отображает свободные места.

V13. Турист выбирает места.

V13.1. Место.

V14. Система отображает доступное меню.

V15. Турист выбирает меню.

V15.1. Тип меню.

V15.2. Тип напитков.

V14. Турист предоставляет информацию по кредитной карте и расчетный адрес.

V14.1. Тип кредитной карты

V14.2. Номер кредитной карты

V14.3. Дата окончания срока действия

V14.4. Название карты

V14.5. Адрес

V15. Система предоставляет номер подтверждения.

2. Аналогичным образом ввести перечень переменных для каждого альтернативного потока.

Б. Определение различных вариантов для каждой переменной

1. Определить значения переменных для тестирования:

V3.1. Дата вылета

– Верная дата в будущем, установленная вручную

– Верная дата в будущем, установленная из календаря

– Дата в прошлом

– Сегодняшняя дата

– Февраль 30 или 31 число

– Пустое поле

V3.2. Время вылета:

– Верное время для будущей даты, установленное вручную

– Верное время для текущей даты, установленное вручную

– Неверный формат даты

– Пустое поле

V3.5. Число путешественников взрослых

– 0

- 1
- 2
- Максимально допустимое

V10.1. Идентификатор

- Верный идентификатор пользователя
- Идентификатор, содержащие недопустимые символы
- Несуществующий идентификатор пользователя
- Пустое поле

V10.2. Пароль

- Правильный пароль пользователя (с правильным идентификатором)
- Неправильный пароль (с правильным идентификатором)
- Верный пароль (с неверным идентификатором)
- Пароль, содержащий недопустимые символы
- Пустое поле

По аналогии определить возможные значения для остальных переменных (не менее трех значений на переменную).

В. Создание тестовых сценариев

1. Построить матрицу распределения тестовых сценариев:

Шаг	Переменная	T1	T2	T3	T4	T5	T6
V3	Время вылета						
V3	Дата прибытия						
V3	Время прибытия						
V3	Число путешествующих взрослых						

2. Для каждой строки ввести все необходимые для тестирования варианты:

Шаг	Переменная	T1	T2	T3	T4	T5	T6
V3	Время вылета	Верная дата в будущем, установленная вручную	Верная дата в будущем, установленная из календаря	Дата в прошлом	Сегодняшняя дата	Февраль 30 или 31 число	Пустое поле
				Верная дата в будущем, установленная из календаря		Верная дата в будущем, установленная из календаря	Верная дата в будущем, установленная вручную

B1 0	Пароль	Правильный пароль пользователя (с правильным идентификатором)	Неправильный пароль (с правильным идентификатором)	Верный пароль (с неверным идентификатором)	Пароль, содержащий недопустимые символы (с неверным идентификатором)	Пароль, содержащий недопустимые символы (с верным идентификатором)	Пустое поле
		Правильный пароль пользователя (с правильным идентификатором)	Правильный пароль пользователя (с правильным идентификатором)	Правильный пароль пользователя (с правильным идентификатором)			

3. Определить значения переменных:

3.1. Для первого тестового сценария (Т1) определить конкретные значения переменных:

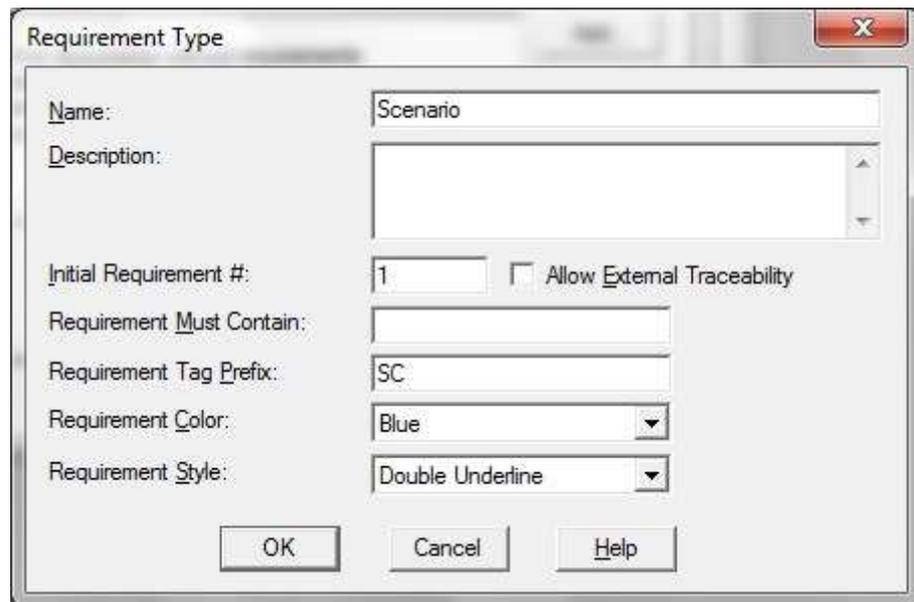
Шаг	Переменная	Т1	Значение	Ожидаемый результат	Фактический результат	Удачный / неудачный	Комментарии
B3	Время вылета	Верная дата в будущем, установленная вручную	16.04.2015	Принято			
B10	Пароль	Правильный пароль пользователя (с правильным идентификатором)	hcvbbsnj	Принято			

3.2. Заполнить аналогичные таблицы по остальным тестовым сценариям.

Г. Создание тестовых сценариев (Test Case) в RequisitePro

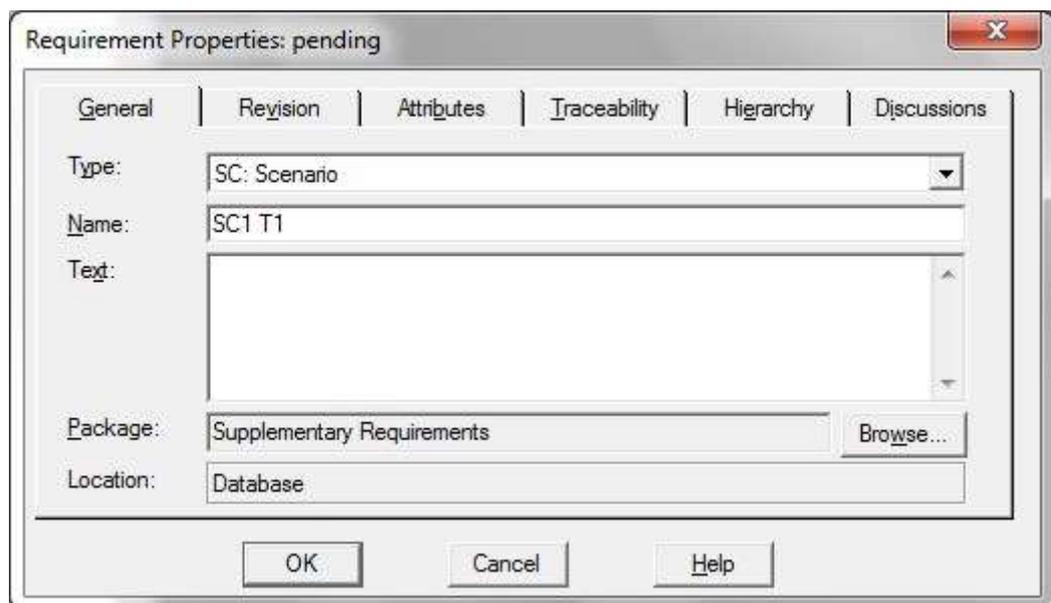
1. Создать новый тип требований

конт. меню SpaceTravel → Properties → ф. Project Properties | вкл. Requirements Type | кн. Add → ф. Requirement Type | Name ← Scenario, Requirement Tag Prefix ← SC, остальные параметры – по умолчанию, кн. Ok → ф. Project Properties | кн. Ok

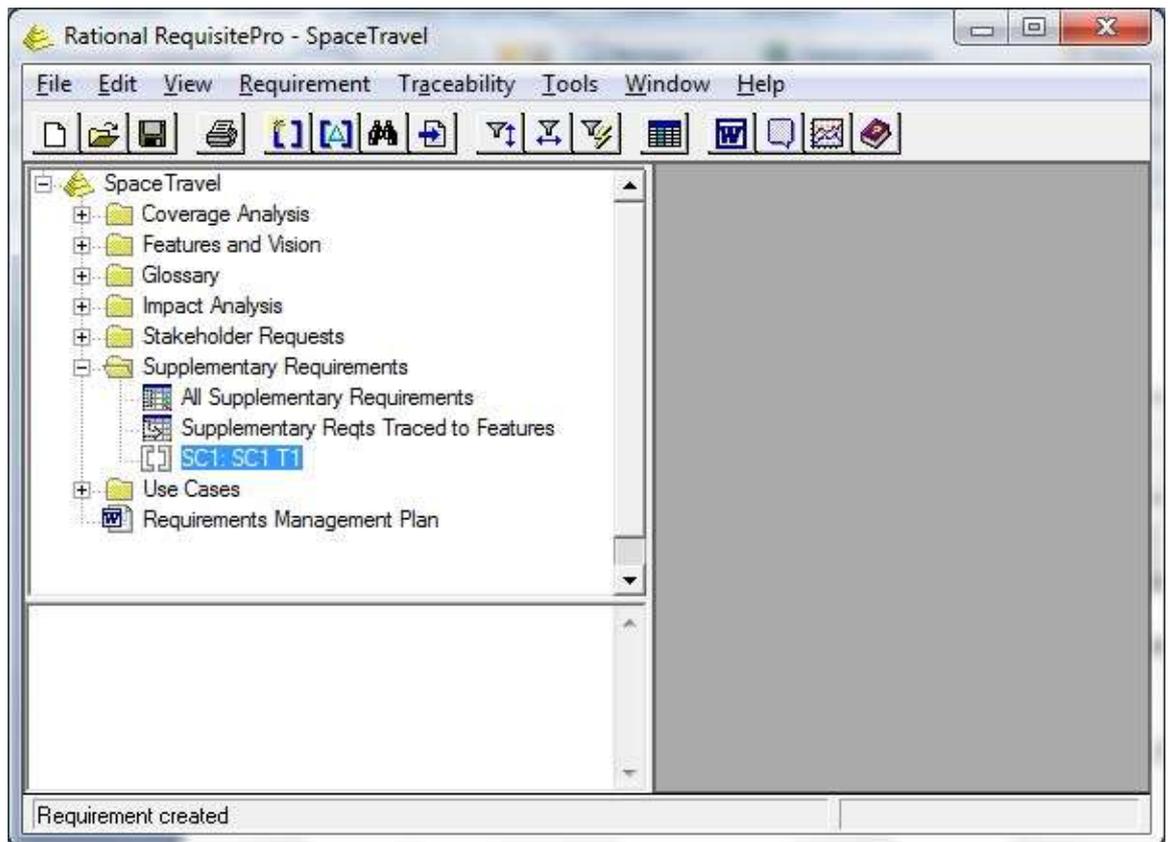


2. Создать новое требование:

SpaceTravel → Supplementary Requirements → конт. меню → New → Requirement... → ф. Requirement Properties: pending | Type ← SC: Scenario, Name ← SC1 T1 (номер сценария использования + номер тестового сценария), кн. Ok



Проверить наличие нового требования в окне проекта.

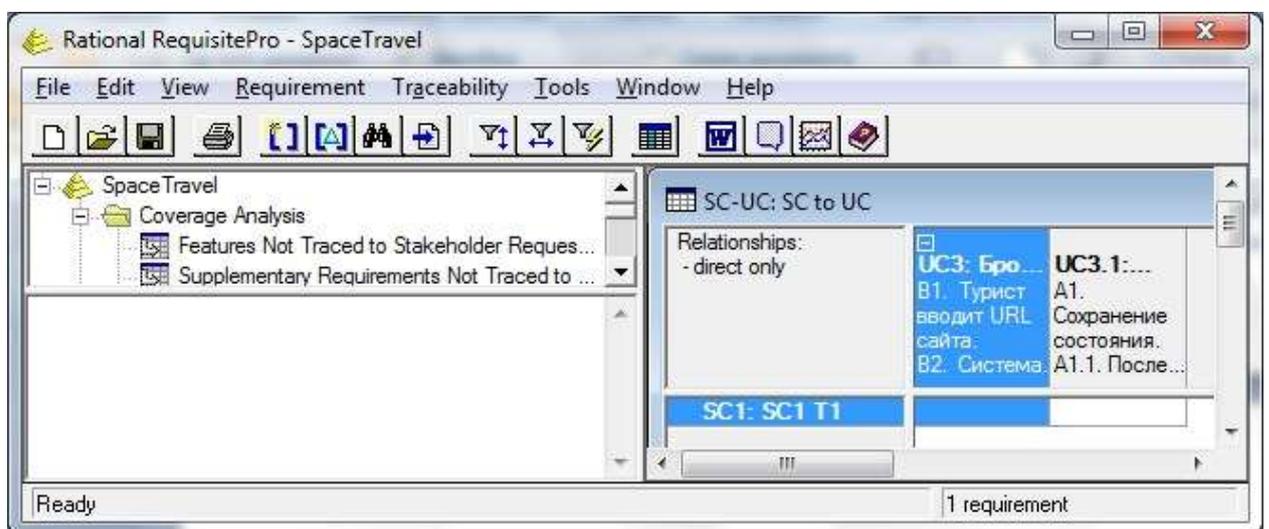
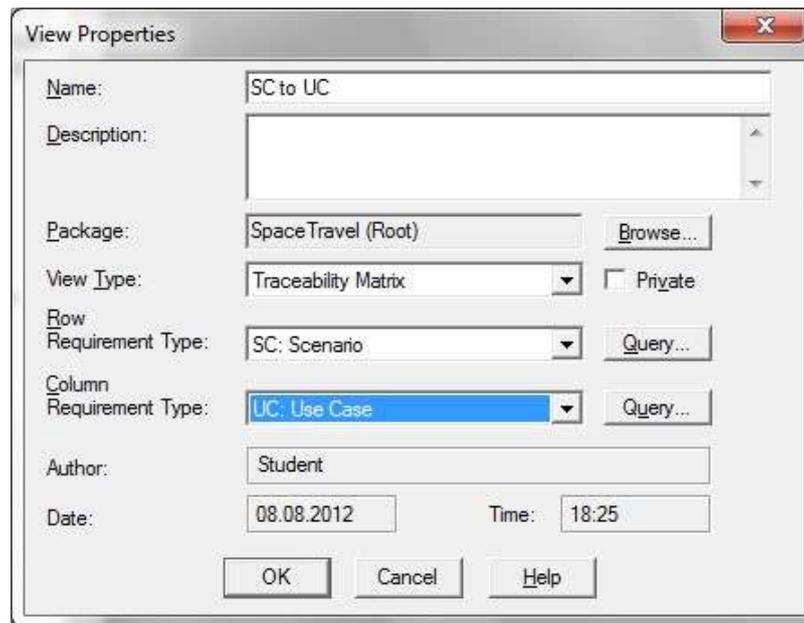


Аналогичным образом определить требования типа Scenario для остальных сценариев использования и тестовых сценариев.

3. Создать матрицу трассировки для связи требований Scenario с требованиями других типов:

3.1. Создать матрицу трассировки

ф. Rational RequisitePro – SpaceTravel → конт. меню → New → View... → ф. View Properties | Name ← SC to UC, View Type ← Traceability Matrix, Row Requirement Type ← SC: Scenario, Column Requirement Type ← UC: Use Case, кн. Ok → матрица трассировки на экране



3.2. Установить связи между требованиями типа SC и UC

Сценарий использования (Use Case) и основанный на нем тестовый сценарий (Scenario) → установить курсор на пересечении строки, соответствующей тестовому сценарию, и столбца, соответствующего сценарию использования → контекстное меню → Trace From

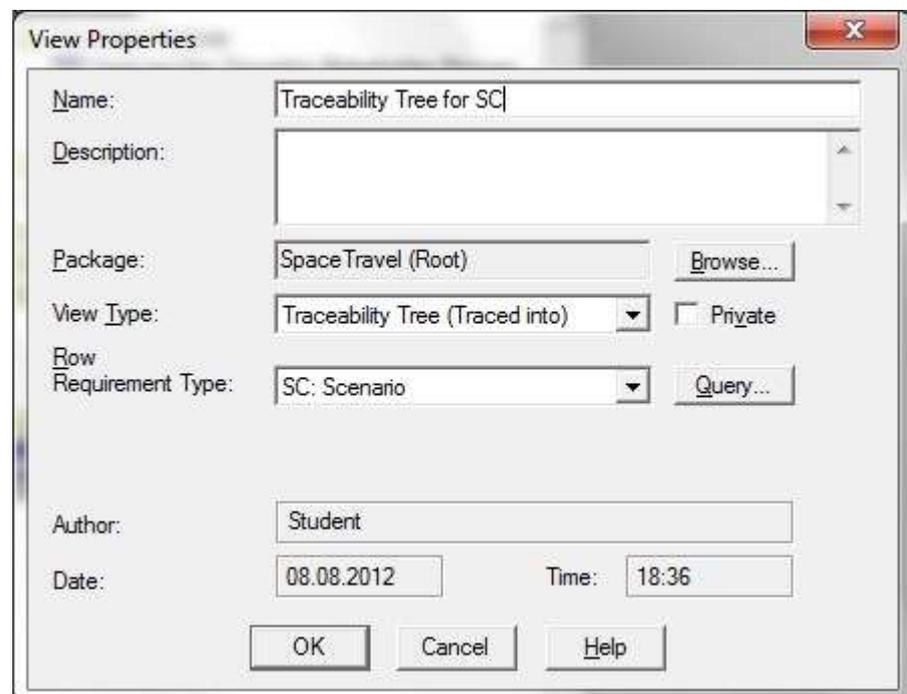


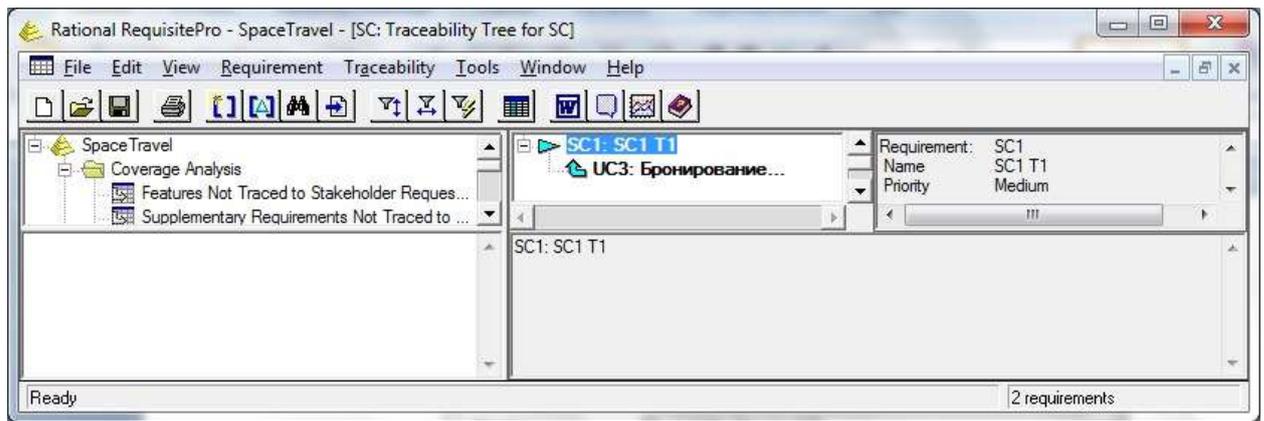
3.3. Аналогичным образом установить связи для всех тестовых сценариев (один сценарий использования может привести к нескольким тестовым сценариям)

3.4. Закрыть матрицу трассировки.

4. Создать дерево трассировки

ф. Rational RequisitePro – SpaceTravel → конт. меню → New → View... → ф. View Properties | Name ← Traceability Tree for SC, View Type ← Traceability Tree (Traced into), Row Requirement Type ← SC: Scenario, кн. Ok → результат на экране





Требования к отчету

Отчет должен содержать:

- название, цель и задачи лабораторной работы;
- краткую теоретическую часть;
- переменные для основного потока сценария использования;
- значения переменных для тестирования;
- матрицу распределения тестовых сценариев;
- распределение значений переменных по тестовым сценариям; – экранные формы работы с RequisitePro; – выводы по результатам работы.

3.3.1 Задания для промежуточной аттестации по МДК 05.01

Проектирование и дизайн информационных систем

В 5 семестре дифференцированный зачет в форме выполнения практического задания.

1. Создайте новый проект.
2. Создайте контекстную диаграмму – рис. 1.
3. Создайте диаграмму декомпозиции – рис. 2.
4. Заполните в MSWord таблицу и отправьте её на почт.ящик 325. Стрелки диаграммы декомпозиции

Имя стрелки	Источник стрелки	Назначение стрелки

5. Создайте диаграмму декомпозиции для работы «Сборка и тестирование» - рис. 3.

6. Сохраните проект. **Критерии оценки выполнения практического задания:**

«Отлично» - самостоятельное выполнение задания; работа соответствует заданию и выполнена в полном объеме; оформлена в соответствии с рекомендациями, отсутствуют грамматические ошибки.

«Хорошо» - самостоятельное выполнение задания; работа соответствует заданию и выполнена в полном объеме; присутствуют недочеты в оформлении и грамматические ошибки.

«Удовлетворительно» - выполнение практического задания не соответствует заданию и выполнена не в полном объеме; присутствуют недочеты в оформлении и грамматические ошибки. Отсутствие экспорта диаграмм.

«Неудовлетворительно» - выполнение практического задания не соответствует описанию, присутствуют недочеты в оформлении и грамматические ошибки. Отсутствие экспорта диаграмм.

В 6 семестре в форме экзамена.

Экзаменационный билет состоит из теоретического вопроса и практического задания.

Теоретические вопросы для проведения экзамена:

1. Понятие АИС. Компоненты АИС.
2. Классификация информационных систем.
3. Жизненный цикл информационных систем. Модели ЖЦ.
4. Методы анализа предметных областей ИС.
5. Типы моделей систем.
6. Постановка задачи обработки информации.
7. Содержание и состав технического задания и технического проекта.
8. Типовое проектирование ИС.
9. Основные модели построения ИС, их структура, особенности, области применения.
10. Структурная модель предметной области.
11. Функционально-ориентированные и объектно-ориентированные методологии описания предметной области.
12. Моделирование бизнес-процессов средствами VPwin.
13. Элементы и их свойства в контекстной диаграмме.
14. Назначение диаграммы дерева узлов. Этапы её создания.
15. Назначение и компоненты стоимостного анализа.

16. Информационное обеспечение ИС.
17. Программное обеспечение АИС.
18. Математическое обеспечение АИС.
19. Техническое обеспечение АИС.
20. Экспертные системы.
21. Системы реального времени.
22. Основные понятия качества ИС.
23. Международная система стандартизации и сертификации качества продукции.
24. Перечень и комплектность документов на ИС.
25. Виды документации.
26. Назначение, виды, и оформление сертификатов.

Практические задания для проведения экзамена:

1. Провести анализ предметной области.
2. Построение структурной схемы сложной системы.
3. Построение графа классификации.
4. Провести оценку экономической эффективности внедрения ИС.
5. Создание контекстной диаграммы.
6. Создание диаграммы декомпозиции.
7. Создание диаграммы узлов.
8. Создание FEO-диаграммы.
9. Выполнить расщепление модели.
10. Выполнить слияние модели.
11. Рассчитать стоимость работ ИС.
12. Создание диаграммы DFD.
13. Создание диаграммы IDEF3.
14. Создание сценария для диаграмм IDEF3.
15. Проектирование спецификации ИС.
16. Разработка общего функционального описания ПС.
17. Разработка руководства пользователя ПС.

Проверяемые результаты обучения.

В процессе выполнения заданий экзаменационного билета обучающиеся осваивают следующие компетенции: ПК 5.1., ПК 5.2., ПК 5.6., ПК 5.7., ОК 1.

– ОК 11.

Критерии оценки усвоения знаний и сформированности умений: «Отлично» - полно раскрыто содержание теоретического вопроса в объеме, предусмотренном программой. Показал навыки работы с используемым программным средством в практическом вопросе, задание выполнено полностью без ошибок и недочетов.

«Хорошо» - в изложении теоретического вопроса допущены небольшие недочеты, исправленные по замечанию преподавателя. Выполнение практического задания полностью, но при наличии не более одной негрубой ошибки и одного недочета, не более трех недочетов.

«Удовлетворительно» - неполно и непоследовательно раскрыто содержание теоретического вопроса, но показано общее понимание вопроса и продемонстрированы практические умения и правильное выполнение не менее 2/3 практической части билета.

«Неудовлетворительно» - обучающийся не смог достаточно полно и правильно ответить на поставленные вопросы, а при выполнении практической части – задание выполнено менее 2/3 всей работы.

3.2.2. Задания для промежуточной аттестации по МДК 05.02

Проектирование и разработка информационных систем

В 5 семестре дифференцированный зачет в форме выполнения теста. К зачету допускаются студенты выполнившие практические работы. Вариант 1

1. Оператор вывода cout может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

А) cout<< "Привет" + name + "n";

Б) cout<< "Привет" << name << "n";

В) cout<< ("Привет" &name& "n");

Г) cout<< "Привет", name, "n"; 2. Какое значение будет содержать переменная x? 1#include

2

3 intx;

4

5intmain()

6{

7 inty;

```
8 std::cout<< x << std::endl;
9 std::cout<< y << std::endl;
10 return 0;
11 }
```

А) неопределённое

Б) 0

В) 1

Г) True

3. Тело оператора выбора if, будет выполняться, если его условие:

А) ложно (false)

Б) Равно NULL

В) истинно (true)

4. Вывод данных в C++

А) cout<< <переменная> << “< строка выводится на экран>” << <выражение> << endl;

Б) cout<< <переменная> ,< “< строка выводится на экран>” ,<выражение> ,endl;

В) cout<< <переменная> ,< “< строка выводится на экран>” ,<выражение> ,endl;

5. Каков будет результат выражения !(1 && !(0

|| 1))?

А) False

Б) True

В) неоднозначность

6. В каком случае можно не использовать фигурные скобки в операторе выбора if?

А) если в теле оператораif всего один оператор

Б) если в теле оператораif два и более операторов

В) если в теле оператораif нет ни одного оператора

Г) нет правильного ответа

7. Какое из следующих значений эквивалентно зарезервированному слову true?

А) 66

Б) Все варианты ответов

В) 0.1

Г) 1

8. Какое значение будет напечатано? #include <iostream>

```
int main()
```

```
{ int x =
```

```
0; int y =
```

```
0;
```

```
if (x++ && y++)
```

```
{ y +=
2; }
std::cout<< x + y << std::endl;
return 0;}
```

А) 3

Б) 2

В) 1 Г)

4

9. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - *.

```
int i, N = 20; for(i
= 0; i < N; i--)
printf("*"); А)
```

```
int i, N = 20; for(i =
0; i < N; N--)
printf("*"); Б) int i,
```

```
N = 40; for(i = 0;
i < N; i--)
```

```
printf("*"); В) int i,
N = 20; for(i = 20;
i < N; i--)
```

Г) int i, N = 20; for(i = 19; i < N; i--) printf("*"); 10. Какой из следующих операторов - оператор сравнения двух переменных? А) =

Б) ==

В) equal

Г) :=

11. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI

А) void main(void)

Б) int main(void)

В) void main()

Г) int main()

12. Простые типы данных в C++.

А) целые - int, вещественные - float или double, символьные - string

Б) целые - bool, вещественные - float или double, символьные - string

В) целые - int, вещественные - float или real, символьные - char

Г) целые - int, вещественные - float или double, символьные - char

13. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:

- А) `#include <> с iostream` внутри скобок
- Б) `include #iostream,h;`
- В) `#include <>; ciostream.h`внутрискобок Г)
`include (iostreamh)`

14. Цикл с предусловием?

- А) `while`
- Б) `do while` В)
`for`

15. Укажите строку, которая возвращает адрес первого элемента в массиве `arr`? А) `arr[0];`

- Б) `&arr;`
- В) `arr;`
- Г) `arr[1];`

Вариант 2

1. Оператор `ifelse` позволяет определить действие ...

- А) только для ложного условия
- Б) для истинного и ложного условий
- В) только для истинного условия
- Г) нет правильного ответа 2. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b =2;
```

```
if (a == b);
```

```
cout<< a << " = " << b <<endl;
```

- А) синтаксическая ошибка
- Б) `a = b`
- В) вывод на экран не выполнится
- Г) `1 = 2`

3. Ввод данных в C++

- А) `cin >><выражение1> >> <выражение2> >> endl >>...;`
- Б) `cin >><выражение1> >> <выражение2>...;`
- В) `cin >><выражение1>,<выражение2>,...;`

4. Результат выполнения следующего фрагмента кода: `!((1 || 0) && 0)`

- А) 1
- Б) результат не может быть заранее определен
- В) 0

5. Какой из следующих логических операторов - логический оператор И?

- А) `&&`
- Б) `&`

В) |

Г) |& 6. Результат выполнения следующего фрагмента кода: `cout<< 22 / 5 * 3;`

А) 12

Б) 1.47

В) 13.2

Г) другое 7. Какое значение будет напечатано, в результате выполнения программы? `#include <iostream>`

```
int main()
{ int x =
3;
switch(x)
{ case 0:
int x = 1;
std::cout<< x << std::endl;
break; case 3: std::cout<<
x << std::endl; break;
default: x = 2;
std::cout<< x << std::endl;
}
return 0;
}</iostream>
```

А) 0

Б) 3

В) ничего не напечатается, программа вообще не будет работать

Г) 1

8. Чему будет равна переменная `a`, после выполнения этого кода `int a; for(a = 0; a < 10; a++) {}`?

А) 1

Б) 9

В) 10

9. Какие служебные символы используются для обозначения начала и конца блока кода?

А) ()

Б) <>

В) { }

Г) `beginend`

10. Какие среды программирования (IDE) предназначены для разработки программных средств?

А) MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse

Б) MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD

В) MVS, NetBeans, QT Creator, RAD Studio, Dev-C++ 11.

Цикл с постусловием?

А) dowhile

Б) while В)

for

12. Какая из следующих записей - правильный комментарий в C++?

А) `** Комментарий **`

Б) `{комментарий}`

В) `*/ Комментарий */`

Г) `/* комментарий */`

13. До каких пор будут выполняться операторы в теле цикла while ($x < 100$)?

А) Пока x больше ста

Б) Пока x строго меньше ста

В) Пока x равен ста

Г) Пока x меньше или равен ста

14. Какой порядковый номер последнего элемента массива, размер массива 19?

А) 19

Б) порядковый номер определяется программистом

В) 18 15. Укажите правильное объявление

массива!

А) `int anarray;`

Б) `anarray{10};`

В) `int anarray[10];`

Г) `array an array[10];`

Ответ в1: 1-Б, 2-А, 3-В, 4-А, 5-Б, 6-А, 7-Г, 8-В, 9-В, 10-Б, 11-Г, 12-Г, 13-А, 14- А, 15-Г.

Ответ в2: 1-Б, 2-Г, 3-Б, 4-А, 5-А, 6-А, 7-В, 8-Б, 9-В, 10-В, 11-А, 12-Г, 13-Б, 14- Б, 15-В.

Критерии оценивания:

Оценка	Критерии, %
«Отлично» - 5	90-100
«Хорошо» - 4	80-89
«Удовлетворительно» - 3	71-79
«Неудовлетворительно» - 2	0-70

В 6 семестре экзамен в форме экзаменационных билетов.

Билет состоит из двух теоретических вопросов и одного практического задания.

Теоретические вопросы для проведения экзамена по МДК.05.02 «Разработка кода информационных систем».

1. Программирование в средах современных информационных систем: создание модульных программ, элементы теории модульного программирования.
2. Программирование в средах современных информационных систем: объектно-ориентированное проектирование и программирование.
3. Языки объектно-ориентированной парадигмы.
4. Достоинства и недостатки ООП.
5. Объектно-ориентированный подход к проектированию и разработке программ:
проектирование по образцам.
6. Сущность объектно-ориентированного подхода в программировании.
7. Соотношение основных понятий объектно-ориентированного подхода: объект-класс-подкласс-суперкласс.
8. Важнейшие принципы объектно-ориентированного подхода:
инкапсуляция, наследование, полиморфизм.
9. Динамическое и статическое связывание.
10. Определение термина «инкапсуляция» в языках объектноориентированной парадигмы программирования.
11. Модульность. Примеры инкапсуляции.
12. Наследование. Определение термина «наследование».
13. Определение базового класса.
14. Соотношение подкласса и суперкласса. Пример записи наследования в программе.
15. Типы наследования: внутреннее, защищенное и публичное наследование, назначение и примеры записи в программе.
16. Виртуальные методы. Абстрактные классы. Множественное наследование:
примеры использования, проблемы использования (конфликт имен методов или атрибутов нескольких базовых классов; многократное включение базового класса).
17. Определение термина «полиморфизм». Разновидности полиморфизма в парадигме объектно-ориентированного программирования. Полиморфные переменные.
18. Перегрузка функций. Параметрическая перегрузка. Чистый полиморфизм.

19. Понятие объекта в объектно-ориентированной парадигме программирования.
20. Атрибуты, методы, интерфейс, время жизни.
21. Определение объекта, свойства, метода, события, класса объектов.
22. Передача сообщений. Инициализация объектов.
23. Операция обращения к атрибуту объекта. Оператор this.
24. Классы в языках программирования объектно-ориентированной парадигмы: понятие, интерфейс, экземпляры класса, ассоциация, наследование.
25. Определение класса: внешняя, внутренняя и защищенная части.
26. Интерфейс и реализация класса.
27. Дружественные и виртуальные функции.
28. Описание класса – атрибуты и методы на примере комплексных чисел.
29. Понятие конструктора в языках объектно-ориентированной парадигмы программирования. Назначение, свойства.
30. Копирующий конструктор. Деструкторы.
31. Назначение, свойства. Пример программы с использованием конструктора и деструктора.
32. Среда разработки; система окон разработки; система меню.
33. Отладка и тестирование программ. Классификация окон системы визуальной разработки приложений.
34. Классификация меню и способы их создания.
35. Классификация ошибок.
36. Инструмент для обнаружения ошибок в приложении.
37. Понятие события в объектно-ориентированной парадигме программирования: природа событий, виды событий, маршрутизация событий.
38. Модификация и обработка событий. Реакция на события.
39. Понятия «модель данных», «база данных», «банк данных», «система управления базами данных». Компоненты среды СУБД. Функции СУБД. Настольные и серверные СУБД. Примеры СУБД.
40. Трехуровневая архитектура БД (внешний, концептуальный, внутренний уровни). Функции СУБД. Типовая организация современной СУБД. Языки БД. Архитектура многопользовательских СУБД.

Практические задания для проведения экзамена по МДК.05.02 «Разработка кода информационных систем» в 6 семестре

Вариант 1.

1. Наиболее точным аналогом реляционной базы данных может служить:
 - 1) неупорядоченное множество данных;
 - 2) вектор;
 - 3) генеалогическое дерево;
 - 4) двумерная таблица.
2. Для чего предназначены формы:
 - 1) для хранения данных базы;
 - 2) для отбора и обработки данных базы;
 - 3) для ввода данных базы и их просмотра;
 - 4) для автоматического выполнения группы команд;
 - 5) для выполнения сложных программных действий;
 - 6) для вывода обработанных данных базы на принтер.
3. Приведены запросы к поисковому серверу. Для каждого запроса указан его код – соответствующая буква от А до Г. Запишите в таблицу коды запросов слева направо в порядке возрастания количества страниц, которые нашёл поисковый сервер по каждому запросу. По всем запросам было найдено разное количество страниц. Для обозначения логической операции «ИЛИ» в запросе используется символ «|», а для логической операции «И» – символ «&». Вариант 2.

1. Назовите области применения ИС

2. База данных - это:

- 1) совокупность данных, организованных по определенным правилам;
- 2) совокупность программ для хранения и обработки больших массивов информации;
- 3) интерфейс, поддерживающий наполнение и манипулирование данными;
- 4) определенная совокупность информации.

3. Укажите количество и номера записей, которые в данном фрагменте удовлетворяют условию: $((\text{Площадь, тыс.км}^2 > 30) \text{ И } (\text{Численность населения, тысяч чел.} > 5000)) \text{ И } (\text{Часть света} = \text{Европа})?$

Пример экзаменационного билета

Специальность 09.02.07 Информационные системы и программирование

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ

№ 1
на 20__/20__ учебный год

МДК.05.02 «Разработка кода информационных систем»

1. Программирование в средах современных информационных систем: создание модульных программ, элементы теории модульного программирования.
2. Программирование в средах современных информационных систем: объектно-ориентированное проектирование и программирование.
3. Задача:

Порядок выполнения практической работы: 1. Собрать предварительную информацию об исследуемом предприятии. 2. Сформулировать видение выполнения проекта и границы проекта. 3. Составить отчет об обследовании. 4. Получить следующие данные: Краткая информация о компании (профиль клиента). Цели проекта. Подразделения и пользователи системы. 5. На основе предварительной информации сформировать и согласовать с заказчиком общее представление о проекте. 6. Оформить результаты в виде отдельного документа – отчета об обследовании.

Специальность 09.02.7 Информационные системы и программирование

ЭКЗАМЕНАЦИОННЫЙ Б И Л Е Т № 2

на 20__/20__ учебный год

МДК.05.02 «Разработка кода информационных систем»

1. Языки объектно-ориентированной парадигмы.
2. Достоинства и недостатки ООП.
3. Задача: Составить DFD диаграмму предметной области обследуемого предприятия.

КРИТЕРИИ ОЦЕНКИ ЭКЗАМЕНА

Ответы на теоретический вопрос и выполнение практического задания оцениваются количеством баллов.

Максимальное количество - 10 баллов (5 баллов – теоретические вопросы, 5 баллов – выполнение практического задания). **Критерии оценки устного ответа:**

«5» (отлично) Обучающийся в полном объеме ответил на все вопросы и дополнительные вопросы, поставленные преподавателем, умеет работать со всеми видами источников, проявив самостоятельность и знания межпредметного характера, применять принципы учебной дисциплины в жизни.

«4» (хорошо) Обучающийся раскрыл содержание вопросов, но в его ответе содержатся недочеты или одна не грубая ошибка; при ответе на поставленные вопросы имеются незначительные замечания и поправки со стороны преподавателя. Обучающийся может самостоятельно добывать знания, пользуясь различными источниками, имеет развитые практические умения, но необязательно их применять.

«3» (удовлетворительно) Обучающийся раскрыл более, чем на 50% содержание вопросов, но его ответ содержит недочеты или 2-3 негрубые ошибки, при ответе на поставленные вопросы преподаватель оказывал ему значительную помощь в виде наводящих вопросов.

Обучающийся знает только основные принципы, умеет добывать знания лишь из основных источников, частично сформированы знания и умения.

«2» (неудовлетворительно) Обучающийся раскрыл менее, чем на 50% содержание вопросов, его ответ содержит более двух грубых ошибок, при ответе на поставленные вопросы преподаватель оказывал ему постоянную помощь. Обучающийся не умеет самостоятельно работать с источниками, не знает принципов учебной дисциплины, у него не сформированы знания и умения.

Критерии оценки выполнения практического задания

Количество баллов	Полнота, системность знаний
5	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу.
4	Задание выполнено полностью самостоятельно и полностью соответствует поставленной задаче или образцу, но при этом допущены несущественные неточности, устраненные помощью преподавателя. без
3	Задание выполнено не в полном объеме или не полностью соответствует поставленной задаче или образцу, при этом могут быть допущены несущественные неточности, устраненные помощью преподавателя.
2	Задание не выполнено и полностью не соответствует поставленной задаче или образцу, допущены существенные неточности, которые обучающийся не может устранить.

Итоговая оценка выставляется с учетом суммы коэффициентов усвоения по каждому заданию (устный ответ и практическое задание).

Критерии оценки экзамена:

Вес оценки устного ответа 0,5. Вес оценки решения задачи 0,5. При выводе итоговой оценки результат округляется, например: устный ответ 4, оценка за задачу 4. Экзаменационная оценка: $4*0,5+4*0,5=2+2=4$. При наличии неудовлетворительной оценки за устный ответ либо задачу проставляется итоговая оценка «неудовлетворительно».

3.2.3. Задания для промежуточной аттестации по МДК 05.03

Тестирование информационных систем

7 семестр – другие формы контроля

Тестовое задание

1. Определение процесса тестирования.
2. Место тестирования в жизненном цикле ПО.
3. Стадии тестирования.
4. Объекты тестирования.
5. Тест-кейсы.
6. Отчет о тестировании.
7. Метрики тестирования.
8. Позитивные и негативные тест-кейсы.
9. Метод эквивалентного разделение.
10. Анализ граничных значений.
11. Метод причина – следствие.
12. Предугадывание ошибки.
13. Исчерпывающее тестирование.
14. Статическое и динамическое тестирование.
15. Методы белого и черного ящика.
16. Тестирование команд, ветвей, путей и условий.
17. Модульное и Интеграционное тестирование.
18. Системное тестирование.
19. Дымовое тестирование.
20. Тестирование критического пути.
21. Расширенное тестирование.
22. Альфа-тестирование и Бета-тестирование.
23. План тестирования.
24. Назначение Плана тестирования.
25. Цели План тестирования

Критерии выставления оценки за тестовое задание:

Оценка	Количество баллов
«5» (отлично)	22-25
«4» (хорошо)	18-21
«3» (удовлетворительно)	12-17
«2» (неудовлетворительно)	Менее 12

8 семестр – экзамен

Задание: разработать Test Suite (не менее 5 Test Case) для ИС бронирования-покупки авиабилетов по вариантам.

Варианты:

1. www.aviasales.ru
2. avia.tutu.ru
3. www.skyscanner.ru
4. aviabilet.ru
5. aviapoisk.ru

Критерии выставления оценки за экзаменационное задание:

Оценка	Показатели
«5» (отлично)	Разработаны 5 тест-кейсов Тест-кейсы отображают корректную информацию Тест-кейсы предусматривают не менее 3 вариантов исхода
«4» (хорошо)	Разработаны 3-4 тест-кейсов Тест-кейсы отображают корректную информацию с ограничениями Тест-кейсы предусматривают 2 вариантов исхода
«3» (удовлетворительно)	Разработан 2 тест-кейс Тест-кейсы отображают корректную информацию с ограничениями Тест-кейсы предусматривают 1 вариантов исхода
«2» (неудовлетворительно)	Разработан 1 тест-кейс Тест-кейсы отображают корректную информацию с ограничениями Тест-кейсы предусматривают 1 вариантов исхода

4. Требования к дифференцированному зачету по учебной и производственной практике

4.1. Общие положения

Дифференцированный зачет по учебной практике выставляется как средний балл по результатам выполнения заданий в течении учебной практики.

Дифференцированный зачет по производственной практике выставляется на основании оценки проставленной руководителем производственной практики от предприятия в аттестационном листе-характеристике и защиты отчета по производственной практике.

Оценка за производственную практику выставляется студенту при предъявлении следующих заполненных документов:

1. задание на производственную практику;
2. дневник прохождения практики (заполняют обучающиеся);
3. аттестационный лист-характеристика (с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика);
4. отчет по производственной практике (написан в соответствии с методическими рекомендациями по написанию отчета по ПП и индивидуальным заданием на ПП).

4.2. Виды работ практики и проверяемые результаты обучения по междисциплинарному курсу

4.2.1. Учебная практика:

Таблица 4.1

Виды работ	Коды проверяемых результатов (ПК, ОК, ПО, У)
сбор исходных данных для разработки проектной документации на информационную систему;	ПК 5.1, ОК 1 – ОК 51, ПО 1 – ПО 3, ПО 8, У 1 – У 2
разработка проектной документации на разработку информационной системы;	ПК 5.1, ПК 5.2., ПО 3 – ПО 5, ПО 8, У 2 – У 4
разработка модели архитектуры информационной системы;	ПК 5.2., ОК 1 – ОК 11, У5
описание бизнес-процессов заданной предметной области;	ПК 5.6., ПО 1 – ПО 6, ПО 8, У 1 – У 4

разработка графического интерфейса пользователя;	ПК 5.1, ПК 5.2., ОК5, У6
тестирование информационной системы, фиксация выявленных ошибок;	ПК 5.6., ПК 5.7., ОК 1 – ОК 11
разработка технической документации на эксплуатацию информационной системы	ПК 5.7., ПО 1 – ПО 6, ПО 8, У 1 – У 4

4.2.2. Производственная практика:

Таблица 4.2

Виды работ	Коды проверяемых результатов (ПК, ОК, ПО, У)
организационные вопросы оформления на предприятии;	ПК 5.1., ОК 1 – ОК 5, У 1 – У 4
ознакомление со структурой и характером деятельности предприятия;	ПК 5.1., ОК 1 – ОК 5, У 1 – У 4
участие в разработке информационной системы для заданной предметной области;	ПК 5.1., ОК 6 – ОК 11, У 1 – У 4
проектирование и разработка интерфейса пользователя;	ПК 5.2., ОК 6 – ОК 10, У 1 – У 4
разработка подсистемы безопасности информационной системы в соответствии с техническим заданием;	ПК 5.2., ПК 5.7., ОК 6 – ОК 11, У 1 – У 4
обобщение материалов прохождения практики.	ПК 5.7., ОК 6 – ОК 10

4.3. Форма аттестационного листа

АТТЕСТАЦИОННЫЙ ЛИСТ-ХАРАКТЕРИСТИКА по производственной практике

1. ФИО обучающегося _____
2. Специальность 09.02.07 Информационные системы и программирование
3. Группа _____ Курс _____
4. Место прохождения практики (организация) _____
5. Время проведения практики с _____ по _____
6. Вид практики: производственная в объеме _____ часа
7. Наименование ПМ.05. Проектирование и разработка информационных систем
8. Виды и объемы работ выполненные обучающимися во время практики:

9. Уровень освоения профессиональных компетенций по ПМ.05.

Проектирование и разработка информационных систем

10. Характеристика (заполняется по результатам освоения общих компетенций)

Результат:

Производственная практика пройдена в объеме _____, с оценкой _____

Руководитель практики от предприятия _____

должность подпись расшифровка подписи

М.П.

Руководитель практики от ОУ _____

должность подпись расшифровка подписи

С результатами прохождения практики ознакомлен _____

дата подпись расшифровка подписи

4.4. Требования к отчету по практике

Выполнение отчета по практике производится в соответствии методическими рекомендациями по организации и прохождению производственной практики по модулю.

5. Структура контрольно-оценочных материалов для экзамена по модулю

Экзамен по модулю предназначен для контроля и оценки результатов освоения профессионального модуля **ПМ.05 Проектирование и разработка информационных систем** по специальности **09.02.07 Информационные системы и программирование**

Экзамен проводится в форме выполнения комплексного практического задания.

Итогом экзамена является однозначное решение: «вид профессиональной деятельности освоен / не освоен» с выставлением оценки.

Для вынесения положительного заключения об освоении ВПД, необходимо подтверждение сформированности всех компетенций, перечисленных в программе ПМ. При отрицательном заключении хотя бы по одной из профессиональных компетенций принимается решение «вид профессиональной деятельности не освоен». Итоговая оценка по профессиональному модулю отражается в сводной ведомости учета освоения профессионального модуля.

Задание экзамена по модулю ПМ 05 Проектирование и разработка информационных систем.

Вариант № 1

Описание предметной области

Вашей задачей является разработка и модификация информационной системы для компании ЗАО «Привет».

«Привет» использует несколько информационных систем для различных задач. Для учета и управления закупками и поставками применяется информационная система на платформе «1С: Предприятие», а для Отдела технического контроля необходимо разработать новую систему на выбранной вами платформе.

Основная задача сотрудника ОТК – предотвратить производственный брак, поэтому он обязан регулярно следить:

- за качеством сырья и рецептурой производства;
- проверять качество работы производственной техники;
- следить за качеством работы сотрудников;
- своевременно останавливать процесс во время выпуска бракованной продукции;
- устранять причины ее появления;
- следить за правильностью эксплуатации оборудования;
- соответствием производства технической документации и государственным стандартам.

ОТК отвечает за некачественную продукцию, за ее возврат на производство.

Вам необходимо разработать информационную систему для ОТК, с которой будут работать сотрудники отдела.

Пользователи ИС: начальник ОТК, администратор (web), лаборант, менеджер по работе с клиентами, контролер (desktop).

Порядок работы лаборатории:

На каждую единицу принятого материала создается заказ, который может содержать в себе услуги – исследования материала. У одного клиента может быть несколько заказов. Хранение данных о всех клиентах и заказах позволит формировать все необходимые отчеты, отслеживать динамику показателей и контроль качества производств, а также автоматизировать работу сотрудников ОТК.

Обеспечьте хранение в базе данных:

- услуги ОТК (наименование, стоимость);
- данные клиентов (ФИО контактного лица, телефон контактного лица, e-mail);

- заказ (дата создания, услуги, стоимость заказа);
- данные других сотрудников ОТК (логин, пароль, ФИО).

Кроме того, при выполнении всего задания демонстрационного экзамена вам нужно будет обеспечить хранение данных, указанных в задании дополнительно.

Практическое задание

Добавьте информационную базу компании, реализованную на технологической платформе «1С:Предприятия 8».

Произведите модификацию отдельных модулей информационной системы в соответствии с рабочим заданием:

– Измените справочник «Номенклатура» сделав его иерархическим и добавив новый реквизит «Характеристики».

– Измените справочник «Сотрудники» добавив в него табличную часть «Трудовая деятельность», который содержит реквизиты: «Предприятие», «Дата приема», «Дата увольнения», «Должность».

– Измените документ «Расходная Накладная» добавив в форму элемента расчет величины скидки за продажу, которая рассчитывается как 5% от итоговой суммы.

– Изменить документ «Приходная Накладная» добавив в табличную часть реквизит для расчёта стоимости цены с НДС, которая рассчитывается на 18% от цены поставщика за единицу.

– Создайте оборотный регистр накоплений «РасходНоменклатуры», который будет измерять количество и сумму проданной номенклатуры. Документом движения регистра является документ «Расходная Накладная».

– Создайте оборотный регистр накопления «Поставки поставщиков», который будет измерять количество поставляемой номенклатуры

поставщиками по виду номенклатуры. Документом движения регистра является документ «Приходная Накладная».

– Создайте отчёт «Поставки поставщиков», который позволяет отобразить все поставки поставщиков за период.

– Добавьте в регистр накопления «ПриходНоменклатуры», измерение «Поставщик» и ресурс «Цена». Измените обработку проведения документа «Приходная Накладная».

Создайте подсистему «Рабочий стол» в которой возможно загрузить все объекты конфигурации. Данная подсистема должна быть первой в списке панели разделов.

Критерии оценки выполнения задания

Оценка	Критерий оценки
«5» (отлично)	обучающийся свободно применяет полученные знания при выполнении практических заданий; выполнил работу в полном объеме с соблюдением необходимой последовательности действий
«4» (хорошо)	выполнены требования к оценке «отлично», но допущены 3-5 недочетов при выполнении практических заданий и обучающийся может их исправить самостоятельно или при небольшой помощи преподавателя
«3» (удовлетворительно)	работа выполнена не полностью, но объем выполненной части позволяет получить правильные результаты и выводы; в ходе выполнения работы обучающийся продемонстрировал слабые практические навыки, были допущены ошибки; обучающийся умеет применять полученные знания при решении простых задач по готовому алгоритму

«2» (неудовлетворительно)	работа выполнена не полностью и объем выполненной работы не позволяет сделать правильных выводов, у обучающегося имеются лишь отдельные представления об изученном материале, большая часть материала не усвоена
------------------------------	--

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 303540294533635982749676679132712847518854643065

Владелец Аскендерова Джамиля Букаровна

Действителен с 11.03.2025 по 11.03.2026