

МИНОБРНАУКИ РОССИИ
Глазовский инженерно-экономический институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Ижевский государственный технический университет имени М.Т.Калашникова»
(ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)



ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ДИАГНОСТИЧЕСКИХ РАБОТ

по дисциплине

**МДК.02.02 «Инструментальные средства разработки
программного обеспечения»**

09.02.07 Информационные системы и программирование

Квалификация Программист

Фонд оценочных средств разработан на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.07 "Информационные системы и программирование", утвержденного приказом Министерства образования и науки Российской Федерации 09 декабря 2016 г. № 1547 с изменениями и дополнениями (приказ Министерства просвещения Российской Федерации от 17.12.2020 № 747 «О внесении изменений в федеральные государственные образовательные стандарты среднего профессионального образования» (зарегистрирован 22.01.2021 № 62178), приказ Министерства просвещения Российской Федерации от 01.09.2022 № 796 «О внесении изменений в федеральные государственные образовательные стандарты среднего профессионального образования» (зарегистрирован 11.10.2022 № 70461))

Организация ГИЭИ (филиал) ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

разработчик:

Разработчик: Горбушин Денис Шарибзянович,
преподаватель СПО

Утверждено: Протокол Ученого совета филиала № 7, от 11 марта 2026 г.

Руководитель образовательной программы

 Т.А. Савельева
13 марта 2026 г.

Согласовано: Начальник отдела по учебно-методической работе

 И.Ф. Яковлева

13 марта 2026 г.

Паспорт фонда оценочных средств
по дисциплине **МДК.02.02 «Инструментальные средства разработки программного обеспечения»**
(наименование дисциплины)

Результаты освоения дисциплины соотносятся с планируемыми результатами освоения образовательной программы, представленными в матрице компетенций выпускника.

В результате освоения дисциплины обучающийся должен:

Код ОК, ПК	Уметь	Знать	Владеть навыками
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.	– распознавать задачу и/или проблему в профессиональном и/или социальном контексте; – анализировать задачу и/или проблему и выделять её составные части; – определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; – составить план действия; определить необходимые ресурсы; – владеть актуальными методами работы в профессиональной и смежных сферах; – реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	– актуальный профессиональный и социальный контекст, в котором приходится работать и жить; – основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; – алгоритмы выполнения работ в профессиональной и смежных областях; – методы работы в профессиональной и смежных сферах; – структуру плана для решения задач; – порядок оценки результатов решения задач профессиональной деятельности	-
ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.	– определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации	– номенклатура информационных источников, применяемых в профессиональной деятельности	-

ОК 3 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях	– определять актуальность нормативно-правовой документации в профессиональной деятельности; – применять современную научную профессиональную терминологию; – определять и выстраивать траектории профессионального развития и самообразования	– содержание актуальной нормативно-правовой документации; – современная научная и профессиональная терминология; – возможные траектории профессионального развития и самообразования	
ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами	– организовывать работу коллектива и команды; – взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности	– психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	
ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	– грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе	– особенности социального и культурного контекста; правила оформления документов и построения устных сообщений.	
ОК-6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения	– описывать значимость своей специальности	– сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности	

ОК-7 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях	– соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности	– правила экологической безопасности при ведении профессиональной деятельности; – основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения	
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	– использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; – применять рациональные приемы двигательных функций в профессиональной деятельности; – пользоваться средствами профилактики перенапряжения характерными для данной специальности	– роль физической культуры в общекультурном, профессиональном и социальном развитии человека; – основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; – средства профилактики перенапряжения	
ОК 9. Использовать информационные технологии в профессиональной деятельности.	– применять средства информационных технологий для решения профессиональных задач; – использовать современное программное обеспечение	– современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности	
ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.	– понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; – участвовать в	– правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика);	

	диалогах на знакомые общие и профессиональные темы; – строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); – писать простые связные сообщения на знакомые или интересующие профессиональные темы	– лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; – особенности произношения; правила чтения текстов профессиональной направленности	
ПК 2.1 Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.	– Анализировать проектную и техническую документацию. – Использовать специализированные графические средства построения и анализа архитектуры программных продуктов. – Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. – Определять источники и приемники данных. – Проводить сравнительный анализ. – Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace). – Оценивать размер минимального набора тестов. – Разрабатывать тестовые пакеты и тестовые сценарии. – Выявлять ошибки в системных	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения. – Основные подходы к интегрированию программных модулей. – Виды и варианты интеграционных решений. – Современные технологии и инструменты интеграции. – Основные протоколы доступа к данным. – Методы и способы идентификации сбоев и ошибок при интеграции приложений. – Методы отладочных классов. – Стандарты качества	– Разрабатывать и оформлять требования к программным модулям по предложенной документации. – Разрабатывать тестовые наборы (пакеты) для программного модуля. – Разрабатывать тестовые сценарии программного средства. – Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

	компонентах на основе спецификаций.	программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. - Графические средства проектирования архитектуры программных продуктов. - Методы организации работы в команде разработчиков	
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.	– Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. – Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. – Использовать различные транспортные протоколы и стандарты форматирования сообщений. – Выполнять тестирование интеграции. – Организовывать постобработку данных. – Создавать классы-исключения на основе базовых классов. – Выполнять ручное и автоматизированное	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения. – Основные подходы к интегрированию программных модулей. – Основы верификации программного обеспечения. – Современные технологии и инструменты интеграции. – Основные протоколы доступа к данным. – Методы и способы идентификации сбоев и ошибок при интеграции	– Интегрировать модули в программное обеспечение. – Отлаживать программные модули. – Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

	тестирование программного модуля. – Выявлять ошибки в системных компонентах на основе спецификаций. – Использовать приемы работы в системах контроля версий.	приложений. – Основные методы отладки. – Методы и схемы обработки исключительных ситуаций. – Основные методы и виды тестирования программных продуктов. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Приемы работы с инструментальными средствами тестирования и отладки. – Методы организации работы в команде разработчиков	
ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств	– Использовать выбранную систему контроля версий. – Использовать методы для получения кода с заданной функциональностью и степенью качества. – Анализировать проектную и техническую документацию. – Использовать инструментальные средства отладки программных продуктов. – Определять источники и приемники данных. – Выполнять тестирование	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения. – Основные подходы к интегрированию программных модулей. – Основы верификации и аттестации программного обеспечения. – Методы и способы	– Отлаживать программные модули. – Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

	интеграции. – Организовывать постобработку данных. – Использовать приемы работы в системах контроля версий. – Выполнять отладку, используя методы и инструменты условной компиляции. – Выявлять ошибки в системных компонентах на основе спецификаций.	идентификации сбоев и ошибок при интеграции приложений. – Основные методы отладки. – Методы и схемы обработки исключительных ситуаций. – Приемы работы с инструментальными средствами тестирования и отладки. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. – Методы организации работы в команде разработчиков	
ПК 2.4. Осуществляют разработку тестовых наборов и тестовых сценариев для программного обеспечения	– Использовать выбранную систему контроля версий. – Анализировать проектную и техническую документацию. – Выполнять тестирование интеграции. – Организовывать постобработку данных. – Использовать приемы работы в системах контроля версий.	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного обеспечения. – Основные подходы к интегрированию программных модулей. – Основы верификации и	– Разрабатывать тестовые наборы (пакеты) для программного модуля. – Разрабатывать тестовые сценарии программного средства. – Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

	– Оценивать размер минимального набора тестов. – Разрабатывать тестовые пакеты и тестовые сценарии. – Выполнять ручное и автоматизированное тестирование программного модуля. – Выявлять ошибки в системных компонентах на основе спецификаций.	аттестации программного обеспечения. – Методы и способы идентификации сбоев и ошибок при интеграции приложений. – Методы и схемы обработки исключительных ситуаций. – Основные методы и виды тестирования программных продуктов. – Приемы работы с инструментальными средствами тестирования и отладки. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. – Методы организации работы в команде разработчиков.	
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам	– Использовать выбранную систему контроля версий. – Использовать методы для получения кода с заданной функциональностью и степенью качества.	– Модели процесса разработки программного обеспечения. – Основные принципы процесса разработки программного	– Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

кодирования	– Анализировать проектную и техническую документацию. – Организовывать постобработку данных. – Приемы работы в системах контроля версий. – Выявлять ошибки в системных компонентах на основе спецификаций.	обеспечения. – Основные подходы к интегрированию программных модулей. – Основы верификации и аттестации программного обеспечения. – Стандарты качества программной документации. – Основы организации инспектирования и верификации. – Встроенные и основные специализированные инструменты анализа качества программных продуктов. – Методы организации работы в команде разработчиков	
-------------	---	---	--

Раздел 1. Зачетно – экзаменационные материалы

Контрольные вопросы для оценки усвоения знаний

1. Необходимые инструментальные средства разработки программ
2. Часто используемые инструментальные средства разработки программ
3. Специализированные инструментальные средства разработки программ
4. Интегрированные среды разработки
5. Средства разработки программного обеспечения
6. Определение «разработка программ». Три этапа разработки программ
7. Средства проектирования приложений
8. Средства реализации программного кода
9. Средства тестирования программ
10. Классы инструментальных средств разработки программ
11. Четыре категории инструментальных программ, применяемые при проектировании экспертных систем
12. Оболочки экспертных систем
13. Языки программирования высокого уровня Среда программирования, поддерживающая несколько парадигм
14. Средства автоматизации разработки экспертных систем
15. Общее программное обеспечение
16. Специальное программное обеспечение
17. Инструментальная система технологии программирования
18. Четыре класса компьютерной поддержки инструментальных систем технологий программирования: комплексность, ориентированность на коллективную разработку, технологическая определенность, интегрированность
19. Компоненты инструментальных систем технологий программирования
20. База данных разработки. Инструментарий. Интерфейсы.
21. Общая архитектура инструментальных систем технологий программирования
22. Инструментальная система поддержки проекта
23. Языково-зависимая инструментальная система
24. Пользовательский интерфейс. Схема организации взаимодействия компьютера и пользователя.
25. Процедурно-ориентированный подход к разработке интерфейсов
26. Объектно-ориентированный подход к разработке интерфейсов
27. Типы интерфейсов. Интерфейс-меню. Интерфейсы со свободной навигацией. Критерии оценки интерфейса пользователем
28. Факторы появления Case-технологий. Что такое Case-технология?
29. Компоненты интегрированного Case-средства.
30. Классификация по категориям Case-средств. Классификация по типам Case-средств. Вспомогательные типы Case-средств.
31. Структурный подход к разработке ИС
32. Объектно-ориентированный подход к разработке ИС
33. Факторы, усложняющие определение возможного эффекта от использования Case-средств
34. Качества организации для успешного внедрения Case-средств
35. Проблемы использования Case-средств
36. Структурный системный анализ
37. Диаграммы «сущность-связь»
38. Диаграммы классов
39. Язык графического описания UML
40. Диаграмма компонентов
41. Диаграмма композитной структуры

42. Диаграмма развёртывания
43. Диаграмма объектов
44. Диаграмма пакетов
45. Диаграмма деятельности
46. Преимущества UML
47. IDEF
48. Диаграммы переходов состояний
49. Методология функционального моделирования ИС
50. Состав функциональной модели
51. Иерархия диаграмм
52. Типы связей между функциями
53. Характеристика современных Case-средств
54. Методология ARIS. Программный продукт ARIS Express.
55. Основные элементы, используемые в нотации ARIS. Архитектура ARIS
56. Имитационное моделирование. Применение имитационного моделирования. Виды имитационного моделирования
57. Дискретно-событийное моделирование
58. Системная динамика
59. Области применения имитационного моделирования
60. Основные этапы компьютерного моделирования
61. Построение концептуальной модели системы
62. Постановка задачи машинного моделирования. Анализ задачи моделирования.
63. Определение требований к исходной информации
64. Выдвижение гипотез и принятие предположений
65. Определение параметров и переменных
66. Установление основного содержания модели
67. Обоснование критериев оценки эффективности системы
68. Определение процедур аппроксимации
69. Описание концептуальной модели
70. Проверка достоверности модели
71. Составление технической документации

В качестве зачета по дисциплине МДК. 02.02. «Инструментальные средства разработки программного обеспечения» студенту предлагается устно ответить на два вопроса из выше предложенного перечня.

Критерии оценки:

Оценка «**отлично**» выставляется студенту, если грамотно изложены и обоснован выбор документов, учащийся решает поставленную задачу. Выполняет качественно весь объем работ. Ответил на контрольный вопрос для проведения зачета подробно и глубоко.

Оценка «**хорошо**» выставляется студенту, если имеются незначительные погрешности в вышеперечисленных критериях оценки; студент ответил на контрольный вопрос для проведения зачета подробно и глубоко.

Оценка «**удовлетворительно**» выставляется студенту, если имеются значительные погрешности в вышеперечисленных критериях оценки; студент ответил на контрольный вопрос для проведения зачета не полно.

Оценка «**неудовлетворительно**» выставляется студенту, если имеются грубые погрешности в вышеперечисленных критериях оценки и не ответил на контрольный вопрос для проведения зачета.

Типовые задания для оценки освоенных умений, компетенций:

Обучающийся демонстрирует знания, практические умения и сформированность профессиональных компетенций, развитие общих компетенций при выполнении заданий.

Образец задачи, предлагаемой для решения в рамках экзамена по предмету «Инструментальные средства разработки программного обеспечения»:

1. Создайте на основании темы в соответствии с вариантом смешанную диаграмму: Разработайте систему для описания порядка подготовки к экзамену, предполагающий получение отличной оценки (IDEF0 -> DFD).

2. Создайте на основании темы в соответствии с вариантом смешанную диаграмму: Разработайте систему, которая должна описывать порядок выполнения практической работы по дисциплине «Проектирование ИС» (IDEF0 -> IDEF3).

3. Создайте на основании темы в соответствии с вариантом смешанную диаграмму: Разработайте систему описания порядка получения водительских прав (DFD -> IDEF3).

4. Разработайте модель IDEF0 в соответствии с вариантом: Разработайте систему описания порядка организации городского спортивного соревнования.

5. Разработайте модель IDEF3 в соответствии с вариантом: Разработайте систему, которая должна описывать порядок организации общеинститутского студенческого мероприятия.

6. Разработайте модель DFD в соответствии с вариантом: Разработайте систему составления учебного графика дисциплин, изучаемых на факультете (IDEF0 -> IDEF3; кооперации)

7. На основании темы в соответствии с вариантом разработать диаграмму: Разработайте систему, которая должна описывать порядок поставок товара в систему розничных киосков. Дайте обозначения для элементов системы IDEF0, разделяя их на основные и вспомогательные

8. Спроектируйте основные и вспомогательные процессы модели IDEF3 по теме: Разработайте систему описания порядка обработки заказов в службе быта.

9. Создайте элементы системы DFD и выделите основные и вспомогательные процессы: Разработайте систему описания работы одного из участков автосалона.

10. Выполните создание диаграммы классов UML по индивидуальному заданию: Разработайте систему описания работы приемного покоя в больнице.

11. Выполните создание диаграммы классов UML по индивидуальному заданию: Разработайте систему для описания порядка приема заявки на поставку продукции на хлебокомбинате.

12. Создает диаграмму вариантов использования для конкретной системы: Разработайте систему, описывающую процесс поставки сезонных товаров в оптовой фирме.

13. Создает диаграмму вариантов использования для конкретной системы: Разработайте систему описывающую процесс работы торгового отдела.

14. Проектирует деятельность системы, используя диаграммы деятельности по теме: Разработайте систему, которая должна описывать порядок поставок товара в систему розничных киосков

15. Проектирует деятельность системы, используя диаграммы деятельности по теме: Разработайте систему учета в видеопрокате.

16. Выполняет создание диаграммы развертывания UML по индивидуальному заданию: Разработайте систему учета проката на лыжной базе

17. Выполняет создание диаграммы развертывания UML по индивидуальному заданию: Разработайте систему описания процесса обслуживания клиента в банке

18. Выполняет моделирование системы с помощью диаграммы состояний: Разработайте систему описания процесса покупки товаров в Интернет-магазине

19. Выполняет моделирование системы с помощью диаграммы состояний: Разработайте систему, описывающей процесс ремонта компьютеров

20. Создайте в соответствии с вариантом смешанную диаграмму: Разработайте систему описания процесса выполнения курсовой работы (IDEF0 -> DFD).

Критерии оценки:

Оценка **«отлично»** выставляется студенту, если спроектирована диаграмма прецедентов, реализована модель базы данных.

Оценка **«хорошо»** выставляется студенту, если диаграмма прецедентов спроектирована с недочетами в части владения инструментарием программного средства или недочетами в части проектирования диаграммы, реализованная модель базы данных содержит недочеты.

Оценка **«удовлетворительно»** выставляется студенту, если диаграмма прецедентов не спроектирована в программном средстве, но верно указаны элементы диаграммы, реализованная модель базы данных содержит недочеты.

Оценка **«неудовлетворительно»** выставляется студенту, если диаграмма прецедентов не спроектирована в программном средстве, но верно указаны элементы диаграммы, реализованная модель базы данных содержит ошибки или не реализована.

Контрольные работы

Текущий контроль результатов усвоения УД в соответствии с рабочей программой происходит при использовании такой формы контроля, как текущая контрольная работа.

Контрольная работа № 1.

Вариант 1.

1. Классификация по сфере (области) использования программных продуктов
2. Сколько периодов обычно выделяют в истории развития Инструментальные средства разработки программного обеспечения и чем они отличаются?
3. Определение метода, нотации и средства
4. Какие основные компоненты входят в состав CASE – средств?
5. Основные функции репозитория
6. На что подразделяют все CASE – средства?

Контрольная работа № 1.

Вариант 2.

1. Перечислите три класса программных продуктов
2. Перечислите названия периодов
3. Что может быть отнесено к CASE – средствам?
4. Какими функциональными возможностями обладают компоненты CASE – средств?
5. Какими свойствами описывается каждый информационный объект, хранящийся в репозитории?
6. Классификация CASE – средств по типам.

Контрольная работа № 1.

Вариант 3.

1. С чем связано развитие инструментария технологии программирования?
2. Подробно опишите 5 и 6 этапы
3. Перечислите базовые принципы CASE – средств
4. С помощью чего осуществляется разработка диаграмм?
5. Основные типы отчетов
6. Классификация CASE – средств по категориям

Контрольная работа № 1.

Вариант 4.

1. Какие группы программных продуктов сформировались в рамках этого направления? Перечислите из и запишите определения.

2. На какой модели основано большинство CASE – средств?
3. Перечислите базовые положения CASE – средств
4. Какие типы контроля реализуются в CASE – средствах?
5. Какие свойства являются основой поддержки процесса разработки современных CASE – средств?
6. Классификация CASE – средств по уровням

Контрольная работа № 2.

Вариант 1.

1. Какие факторы имеют сильное воздействие на эффективность используемых методов синхронизации изменений?
2. Для каких целей используются системы контроля версий?
3. Виды моделей (диаграмм)
4. В чем заключается построение SADT – модели?
5. Что позволяет разрабатывать UML?
6. На какие виды подразделяются модели языка UML и что описывают?

Контрольная работа № 2.

Вариант 2.

1. Какую модель разработки предпочитают различные коллективы программистов?
2. Какая необходимость возникает при одновременной разработке нескольких программ?
3. Для чего используются данные модели?
4. Этапы построения диаграмм?
5. Для чего предназначен язык UML?
6. На какие уровни подразделяются модели языка UML и что они собой представляют?

Контрольная работа № 2.

Вариант 3.

1. Почему возникает потребность в ПО контроля и объединения версий?
2. Какой вариант является наиболее приемлемым для разработчика ПО?
3. Где используется метод SADT и на чем он реализован?
4. Какими правилами определяется синтаксис диаграмм?
5. На каких принципах основан язык UML?
6. Что называется рациональным унифицированным процессом RUP и кто его автор?

Контрольная работа № 2.

Вариант 4.

1. Для каких целей используют мощные архиваторы?
2. Перечислите основные варианты структуры и организации коллективной разработки программ?
3. Что представляет собой метод SADT?
4. Перечислите и опишите типы связей между функциями
5. Виды диаграмм в языке UML
6. Что описывается в диаграммах языка UML?

Критерии оценки:

Контрольная работа оценивается по пятибалльной шкале, стоимость одного вопроса – 1 балл. За правильный ответ можно получить 1 балл, за неверный ответ или его отсутствие баллы не начисляются.

Оценка «5» соответствует 86% – 100%

Оценка «4» соответствует 73% – 85%

Оценка «3» соответствует 53% – 72%

Оценка «2» соответствует 0% – 52%

Практические работы

Применение языка UML для проектирования программного обеспечения средствами IBM Rational Rose

Теоретическая часть

Rational Rose – CASE-средство, предназначенное для автоматизации этапов анализа и проектирования ПО, а также для генерации кодов на различных языках и выпуска проектной документации.

CASE-средство (Computer Aided Software Engineering) – это инструмент, который позволяет автоматизировать процесс разработки информационной системы и программного обеспечения.

UML (Unified Modeling Language) – это графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем. В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций, получивших название диаграмм.

Rational Rose позволяет генерировать программный код на различных языках программирования (Java, C++, VisualBasic, PowerBuilder), а так же осуществлять обратную генерацию диаграмм (реинжиниринг) на основе программного кода и выпускать проектную документацию.

В распоряжение проектировщика системы Rational Rose предоставляет следующие типы диаграмм, последовательное создание которых позволяет получить полное представление о всей проектируемой системе и об отдельных ее компонентах:

- Use case diagram (диаграммы прецедентов);
- Deployment diagram (диаграммы топологии);
- Statechart diagram (диаграммы состояний);
- Activity diagram (диаграммы активности);
- Interaction diagram (диаграммы взаимодействия);
- Sequence diagram (диаграммы последовательностей действий);
- Collaboration diagram (диаграммы сотрудничества);
- Class diagram (диаграммы классов);
- Component diagram (диаграммы компонент).

Диаграммы прецедентов (Use case diagram)

Этот вид диаграмм позволяет создать список операций, которые выполняет система. Часто этот вид диаграмм называют диаграммой функций, потому что на основе набора таких диаграмм создается список требований к системе и определяется множество выполняемых системой функций.

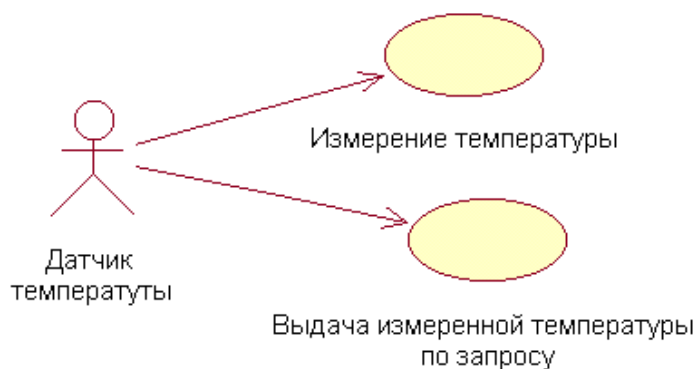


Рис.1. Диаграмма прецедентов

Каждая такая диаграмма или, как ее обычно называют, каждый Use case - это описание сценария поведения, которому следуют действующие лица (Actors).

Данный тип диаграмм используется при описании бизнес процессов автоматизируемой предметной области, определении требований к будущей программной системе. Отражает объекты как системы, так и предметной области и задачи, ими выполняемые.

Диаграммы топологии (Deployment diagram)

Этот вид диаграмм предназначен для анализа аппаратной части системы, то есть «железа», а не программ. В прямом переводе с английского Deployment означает «развертывание», но термин «топология» точнее отражает сущность этого типа диаграмм.

Для каждой модели создается только одна такая диаграмма, отображающая процессоры (Processor), устройства (Device) и их соединения.

Обычно этот тип диаграмм используется в самом начале проектирования системы для анализа аппаратных средств, на которых она будет эксплуатироваться.

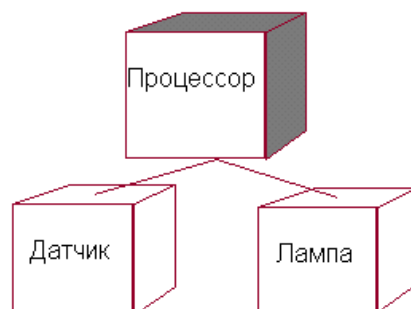


Рис.2. Диаграмма топологии

Диаграммы состояний (State Machine diagram)

Каждый объект системы, обладающий определенным поведением, может находиться в определенных состояниях, переходить из состояния в состояние, совершая определенные действия в процессе реализации сценария поведения объекта. Поведение большинства объектов реальных систем можно представить с точки зрения теории конечных автоматов, то есть поведение объекта отражается в его состояниях, и данный тип диаграмм позволяет отразить это графически. Для этого используется два вида диаграмм: Statechart diagram (диаграмма состояний) и Activity diagram (диаграмма активности).



Рис.3. Диаграмма состояний

Диаграмма состояний (Statechart) предназначена для отображения состояний объектов системы, имеющих сложную модель поведения. Это одна из двух диаграмм State Machine, доступ к которой осуществляется из одного пункта меню.

Диаграммы активности (Activity diagram)

Это дальнейшее развитие диаграммы состояний. Фактически данный тип диаграмм может использоваться и для отражения состояний моделируемого объекта, однако, основное назначение Activity diagram в том, чтобы отражать бизнес-процессы объекта. Этот тип диаграмм позволяет показать не только последовательность процессов, но и ветвление и даже синхронизацию процессов.

Этот тип диаграмм позволяет проектировать алгоритмы поведения объектов любой сложности, в том числе может использоваться для составления блок-схем.

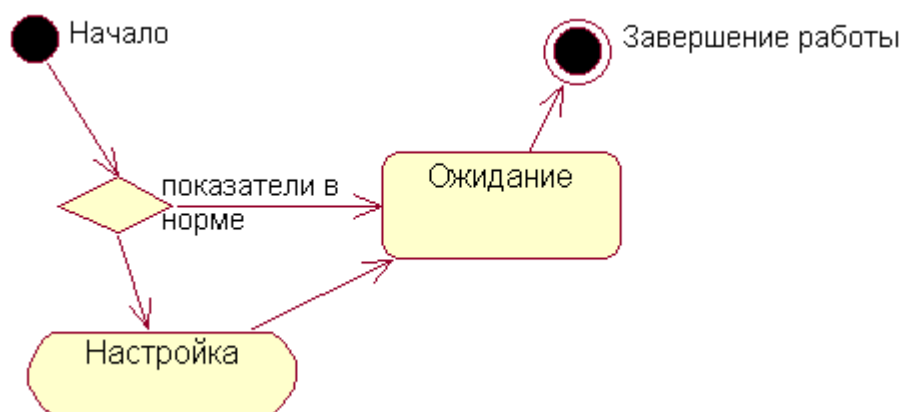


Рис.4. Диаграмма активности

Диаграммы взаимодействия (Interaction diagram)

Этот тип диаграмм включает в себя диаграммы Sequence diagram (диаграммы последовательностей действий) и Collaboration diagram (диаграммы сотрудничества). Эти диаграммы позволяют с разных точек зрения рассмотреть взаимодействие объектов в создаваемой системе.

Диаграммы последовательностей действий (Sequence diagram)

Взаимодействие объектов в системе происходит посредством приема и передачи сообщений объектами-клиентами и обработки этих сообщений объектами-серверами. При этом в разных ситуациях одни и те же объекты могут выступать и в качестве клиентов, и в качестве серверов.

Данный тип диаграмм позволяет отразить последовательность передачи сообщений между объектами.

Этот тип диаграмм не акцентирует внимание на конкретном взаимодействии, главный акцент уделяется последовательности приема/передачи сообщений. Для того чтобы окинуть взглядом все взаимосвязи объектов, служит Collaboration diagram.

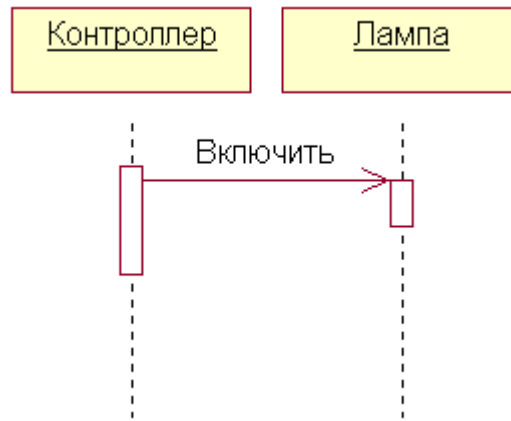


Рис.5. Диаграмма последовательностей действий

Диаграммы сотрудничества (Collaboration diagram)

Этот тип диаграмм позволяет описать взаимодействия объектов, абстрагируясь от последовательности передачи сообщений. На этом типе диаграмм в компактном виде отражаются все принимаемые и передаваемые сообщения конкретного объекта и типы этих сообщений.

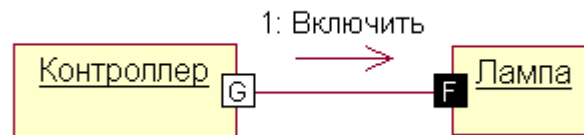


Рис.6. Диаграмма сотрудничества

По причине того, что диаграммы Sequence и Collaboration являются разными взглядами на одни и те же процессы, Rational Rose позволяет создавать из Sequence диаграммы диаграмму Collaboration и наоборот, а также производит автоматическую синхронизацию этих диаграмм.

Диаграммы классов (Class diagram)

Этот тип диаграмм позволяет создавать логическое представление системы, на основе которого создается исходный код описанных классов.

Значки диаграммы позволяют отображать сложную иерархию систем, взаимосвязи классов (Classes) и интерфейсов (Interfaces). Данный тип диаграмм противоположен по содержанию диаграмме Collaboration, на котором отображаются объекты системы. Rational Rose позволяет создавать классы при помощи данного типа диаграмм в различных нотациях. Похожего на облако. Таким образом Г.Буч пытается показать, что класс - это лишь шаблон, по которому в дальнейшем будет создан конкретный объект.

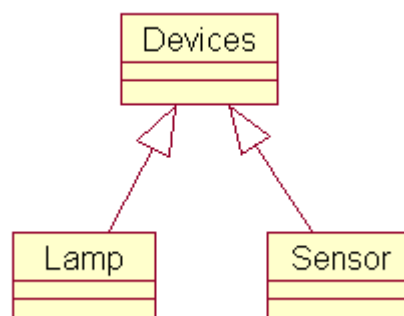


Рис.7. Диаграмма классов

Диаграммы компонентов (Component diagram)

Этот тип диаграмм предназначен для распределения классов и объектов по компонентам при физическом проектировании системы. Часто данный тип диаграмм называют диаграммами модулей.

При проектировании больших систем может оказаться, что система должна быть разложена на несколько сотен или даже тысяч компонентов, и этот тип диаграмм позволяет не потеряться в обилии модулей и их связей.

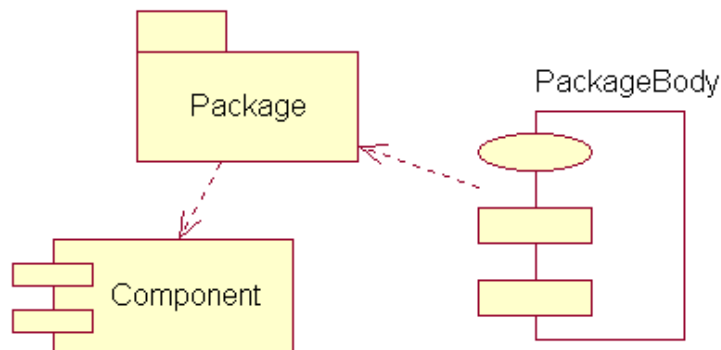


Рис.8. Диаграмма классов

1. Порядок выполнения лабораторной работы

Для выполнения всех лабораторных работ предлагается единый порядок, предусматривающий следующие шаги:

1. Ознакомиться с постановкой задачи и исходными данными.
2. Разработать предлагаемую в работе диаграмму.
3. Реализовать разработанную диаграмму в среде Rational Rose.
4. Сохранить файл модели.
5. Составить отчет по проделанной работе.
6. Представить отчет по лабораторной работе для защиты.

2. Защита отчета по лабораторной работе

Отчет по лабораторной работе должен состоять из следующих структурных элементов:

1. Титульный лист.
2. Текстовая часть.
3. Приложение: разработанная модель вариантов использования.

Текстовая часть отчета должна включать пункты:

- условие задачи;
- порядок выполнения
- краткие сведения о составе и компонентах построенной модели.

Защита отчета по лабораторной работе заключается в предъявлении преподавателю полученных результатов и демонстрации полученных навыков при ответах на вопросы преподавателя.

ЛАБОРАТОРНАЯ РАБОТА № 1

Построение диаграммы вариантов разрабатываемого программного обеспечения средствами IBM Rational Rose

Цель работы: разработать диаграмму вариантов использования в среде Rational Rose для своего индивидуального задания.

Теоретическая часть. Моделирование бизнес-процессов предметной области

Вариант использования представляет собой последовательность действий (транзакций), выполняемых системой в ответ на событие, инициируемое некоторым внешним объектом (действующим лицом). Вариант использования описывает типичное взаимодействие между

пользователем и системой. В простейшем случае вариант использования определяется в процессе обсуждения с пользователем тех функций, которые он хотел бы реализовать.

Действующее лицо (actor) – это роль, которую пользователь играет по отношению к системе. Действующие лица представляют собой роли, а не конкретных людей или наименования работ. Несмотря на то, что на диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок, действующее лицо может также быть внешней системой, которой необходима некоторая информация от данной системы. Показывать на диаграмме действующих лиц следует только в том случае, когда им действительно необходимы некоторые варианты использования.

Действующие лица делятся на три основных типа – пользователи системы, другие системы, взаимодействующие с данной, и время. Время становится действующим лицом, если от него зависит запуск каких-либо событий в системе.

В языке UML на диаграммах вариантов использования поддерживается несколько типов связей между элементами диаграммы.

Это связи коммуникации (communication), включения (include), расширения (extend) и обобщения (generalization).

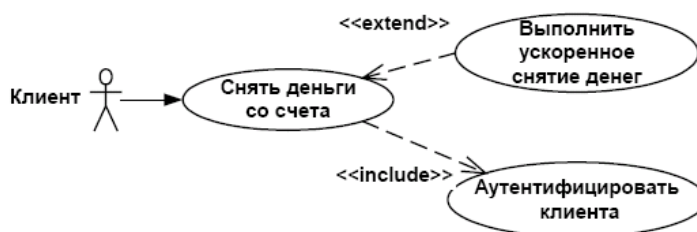
Связь коммуникации – это связь между вариантом использования и действующим лицом. На языке UML связи коммуникации показывают с помощью однонаправленной ассоциации (сплошной линии со стрелкой).

Направление стрелки позволяет понять, кто инициирует коммуникацию.

Связь включения применяется в тех ситуациях, когда имеется какой-либо фрагмент поведения системы, который повторяется более чем в одном варианте использования. С помощью таких связей обычно моделируют многократно используемую функциональность.

Связь расширения применяется при описании изменений в нормальном поведении системы. Она позволяет варианту использования только при необходимости использовать функциональные возможности другого.

На языке UML связи включения и расширения показывают в виде зависимостей с

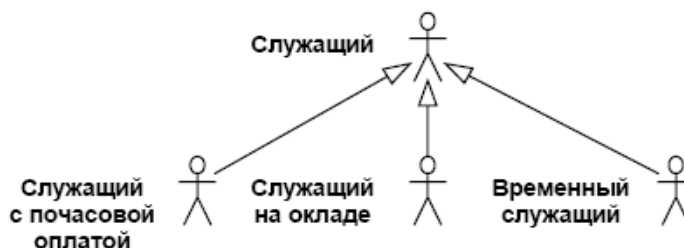


соответствующими стереотипами, как показано на рисунке 1.

Рисунок 1 - Связи использования и расширения

С помощью связи обобщения показывают, что у нескольких действующих лиц имеются общие черты. Например, клиенты могут быть двух типов: корпоративные и индивидуальные. Эту связь можно моделировать с помощью нотации, показанной на рисунке 3.

Рис.3. Обобщение действующего лица



Нет необходимости всегда создавать связи этого типа. В общем случае, они нужны, если поведение действующего лица одного типа отличается от поведения другого постольку, поскольку это затрагивает систему. Если оба подтипа используют одни и те же варианты использования, показывать обобщение действующего лица не требуется.

Порядок выполнения работы

1. Построить модель предметной области, согласно варианту (Приложение А) с помощью диаграммы вариантов использования UML.
2. Оформить отчет по лабораторной работе.
3. Представить отчет по лабораторной работе для защиты.

Порядок построения модели

Создание бизнес-схемы компании

1. Щелкните правой кнопкой мыши на представлении Use Case View в браузере.
2. Выберем пункт New далее Package
3. Назовем новый пакет «Общая схема»

Чтобы поместить действующее лицо в браузер:

1. Щелкните правой кнопкой мыши на пакете «Общая схема» представления Use Case View в браузере.
2. Выберите в открывшемся меню пункт New далее Actor
3. В браузере появится новое действующее лицо под названием NewClass. Слева от его имени вы увидите пиктограмму действующего лица UML.
4. Выделив новое действующее лицо, введите его имя.
5. Щелкните правой кнопкой мыши на действующем лице.
6. В открывшемся меню выберите пункт Open Specification.
7. В поле стереотипа выберите Business Actor и нажмите на кнопку ОК.
8. После создания действующих лиц сохраните модель с помощью пункта меню File затем Save.

Чтобы поместить вариант использования в браузер:

1. Щелкните правой кнопкой мыши на пакете «Общая схема» представления Use Case View в браузере.
2. Выберите в появившемся меню пункт New > Use Case
3. Новый вариант использования под названием NewUseCase появится в браузере. Слева от него будет видна пиктограмма варианта использования UML.
4. Выделив новый вариант использования, введите его название.
5. Щелкните правой кнопкой мыши на варианте использования.
6. В открывшемся меню выберите пункт Open Specification.
7. В поле стереотипа выберите Business Use Case и нажмите на кнопку ОК.

Для создания новой диаграммы вариантов использования:

1. Щелкните правой кнопкой мыши на пакете «Общая схема» представления Use Case View в браузере.
2. Из всплывающего меню выберите пункт New далее Use Case Diagram.
3. Выделив новую диаграмму, введите ее имя («Общая схема действий»).
4. Дважды щелкните на названии этой диаграммы в браузере, чтобы открыть ее.
5. Чтобы поместить действующее лицо или вариант использования на диаграмму, перетащите его мышью из браузера на диаграмму вариантов использования.
6. С помощью кнопки Unidirectional Association (Однонаправленная ассоциация) панели инструментов нарисуйте ассоциации между действующими лицами и вариантами использования.

Пример выполнения задания представлен в Приложении Б.

ЛАБОРАТОРНАЯ РАБОТА № 2

Построение UML диаграмм, изображающих логические схемы баз данных, средствами IBM Rational Rose

Цель работы: смоделировать базу данных на примере построения диаграмм классов средствами IBM Rational Rose.

Теоретическая часть. Моделирование требований к программному обеспечению

Разработка логической структуры. После завершения формирования принципов использования системы, наступает этап разработки ее логической структуры. В Rational Rose он именуется "Logical View". Логическое представление, концентрируется на том, как система будет реализовывать поведение, описанное в вариантах использования. Оно дает подробную картину составных частей системы и описывает взаимодействие этих частей. Логическое представление включает, помимо прочего, конкретные требуемые классы, диаграммы классов и диаграммы состояний. С их помощью конструируется детальный проект создаваемой системы.

Логическое представление содержит:

- классы,
 - диаграммы классов.
- Как правило, для описания системы используется несколько диаграмм классов, каждая из которых отображает некоторое подмножество всех классов системы,
- диаграммы взаимодействия, применяемые для отображения объектов, участвующих в одном потоке событий варианта использования,
 - диаграммы состояний,
 - пакеты, являющиеся группами взаимосвязанных классов.

Для каждого блока использования строится диаграмма взаимодействия, на которой отображается взаимодействие во времени объектов, выполняющих поставленную задачу. На подобных диаграммах идентифицируются объекты системы и определяются сообщения, с помощью которых эти объекты взаимодействуют

Диаграммы взаимодействия (interaction diagrams) описывают поведение взаимодействующих групп объектов. На такой диаграмме отображается ряд объектов и те сообщения, которыми они обмениваются между собой.

Сообщение (message) – это средство, с помощью которого объект-отправитель запрашивает у объекта получателя выполнение одной из его операций.

Информационное (informative) сообщение – это сообщение, снабжающее объект-получатель некоторой информацией для обновления его состояния.

Сообщение-запрос (interrogative) – это сообщение, запрашивающее выдачу некоторой информации об объекте-получателе.

Императивное (imperative) сообщение – это сообщение, запрашивающее у объекта-получателя выполнение некоторых действий.

Существует два вида диаграмм взаимодействия: диаграммы последовательности (sequence diagrams) и кооперативные диаграммы (collaboration diagrams).

Каждый объект на диаграмме последовательностей сопровождается именем класса, к которому он принадлежит. Конкретный объект является экземпляром некоторого класса. Классы образуют логическую структуру системы.

Результатом данного этапа должна стать главная диаграмма, и детализирующие диаграммы для ее элементов.

На этом этапе следует определить классы, которые необходимы в системе. Экземпляры этих классов уже указаны на диаграммах последовательностей. Классы и их связи отражаются в модели в виде диаграммы классов. Группы классов на этих диаграммах могут быть объединены в пакеты.

Проектирование логической структуры следует начинать с определения основных пакетов. Пакет – универсальное средство для группировки элементов модели. Применение

пакетов позволяет сделать модель более обозримой. Пакеты могут быть вложенными друг в друга. Классы, составляющие каждый пакет, детализируются на вложенной диаграмме.

Идентификация основных абстракций заключается в предварительном определении набора классов системы (классов анализа) на основе описания предметной области.

Все классы и диаграммы, описывающие системный проект, помещаются в пакет с именем Design Model.

Диаграммы классов, реализующие варианты использования и диаграммы взаимодействия, отражающие взаимодействие объектов в процессе реализации сценариев варианта использования, помещаются в кооперацию с именем данного варианта использования и стереотипом «use-case realization». Все кооперации помещаются в пакет с именем Use-Case Realization. Связь между вариантом использования и его реализацией изображается на специальной диаграмме трассировки (рис.4).

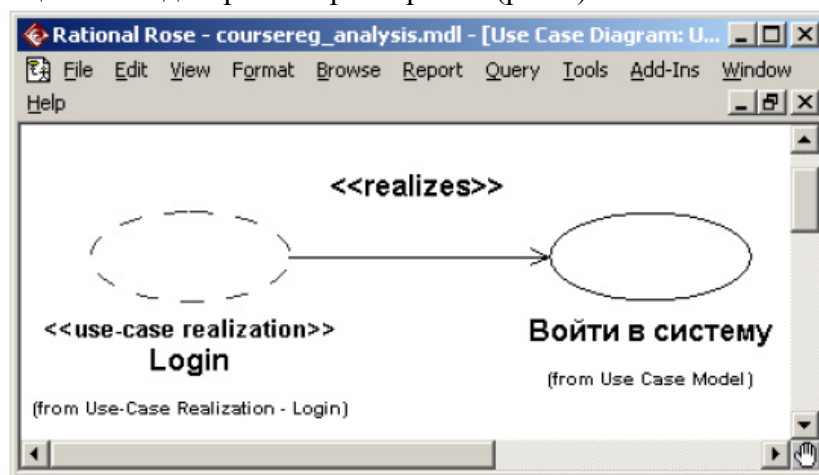


Рис.4. Диаграмма трассировки

Порядок выполнения работы

1. Для реализации сценариев варианта использования постройте
 - а. диаграммы классов, реализующих вариант использования,
 - б. диаграммы взаимодействия, отражающие взаимодействие объектов в процессе.
 - в. кооперативные диаграммы для построенных диаграмм взаимодействия
2. Все построенные диаграммы помещаются в кооперацию с именем данного варианта использования и стереотипом «use-case realization». Все кооперации помещаются в пакет с именем Use Case Realizations.
3. Оформить отчет по лабораторной работе.
4. Представить отчет по лабораторной работе для защиты.

Порядок построения модели

Создание классов

1. Щелкните правой кнопкой мыши на представлении Logical View.
2. Выберите в открывшемся меню пункт New\Class. Новый класс под названием NewClass появится в браузере.
3. Выделите его и введите имя класса.
4. Щелкните правой кнопкой мыши на созданном классе.
5. В открывшемся меню выберите пункт Open Specification.
6. В поле стереотипа выберите необходимый стереотип (Boundary, Control, Entity) и нажмите на кнопку ОК.
7. Откройте диаграмму Main и перетащите созданные классы.

Создание пакетов и диаграммы Traceabilities:

8. Щелкните правой кнопкой мыши на представлении Logic View.
9. В открывшемся меню выберите пункт New\Package.
10. Создайте пакет Use-Case Realizations, затем внутри него – пакеты соответствующие построенным вариантам использования.

11. В каждом из пакетов создайте соответствующие кооперации (каждая кооперация представляет собой вариант использования со стереотипом «use-case realization», который задается в спецификации варианта использования).
12. Создайте в пакете Use-Case Realizations новую диаграмму вариантов использования с названием Traceabilities, которая показывает связь между вариантом использования и его реализацией (диаграмма трассировки).

Создание диаграмм взаимодействия

Настройка

1. В меню модели выберите пункт Tools далее Options.
2. Перейдите на вкладку диаграмм.
3. Контрольные переключатели Sequence Numbering, Collaboration Numbering должны быть помечены, а Focus of Control – нет.
4. Нажмите ОК, чтобы выйти из окна параметров.

Создание диаграммы последовательности

1. Щелкните правой кнопкой мыши на кооперации
2. В открывшемся меню выберите пункт New далее Sequence Diagram.
3. Назовите новую диаграмму.
4. Дважды щелкните на ней, чтобы открыть ее.

Добавление на диаграмму действующего лица, объектов и сообщений

1. Перетащите действующее лицо из браузера на диаграмму.
2. Перетащите классы из браузера на диаграмму.
3. На панели инструментов нажмите кнопку Object Message (Сообщение объекта).
4. Проведите мышью от линии жизни действующего лица к линии жизни объекта
5. Выделив сообщение, введите его имя.
6. Повторите действия 3 – 5, чтобы поместить на диаграмму остальные сообщения (для рефлексивного сообщения используется кнопка Message to Self).

Соотнесение сообщений с операциями

1. Щелкните правой кнопкой на тексте сообщения
2. В открывшемся меню выберите пункт <new operation>. Появится окно спецификации операции.
3. В поле имени оставьте имя сообщения.
4. Нажмите на кнопку ОК, чтобы закрыть окно спецификации операции и вернуться на диаграмму.
5. Повторите действия 1 – 4, пока не соотнесете с операциями все остальные сообщения.

Создание примечаний

Чтобы поместить на диаграмму примечание:

1. Нажмите на панели инструментов кнопку Note.
2. Щелкните мышью в том месте диаграммы, куда собираетесь поместить примечание.
3. Выделив новое примечание, введите туда текст.
4. Чтобы прикрепить примечание к элементу диаграммы, на панели инструментов нажмите кнопку Anchor Notes To Item (Прикрепить примечания к элементу).
5. Нажав левую кнопку мыши, проведите указатель от примечания до элемента диаграммы, с которым оно будет связано. Между примечанием и элементом возникнет штриховая линия.
6. Чтобы создать примечание-ссылку на другую диаграмму создайте пустое примечание (без текста) и перетащите на него из браузера нужную диаграмму.

Чтобы поместить на диаграмму текстовую область:

1. На панели управления нажмите кнопку Text Box.
2. Щелкните мышью внутри диаграммы, чтобы поместить туда текстовую область.
3. Выделив эту область, введите в нее текст.

Создание кооперативной диаграммы

Для создания кооперативной диаграммы достаточно открыть диаграмму последовательности и нажать клавишу F5.

Пример выполнения задания представлен в Приложении В.

ЛАБОРАТОРНАЯ РАБОТА № 3

Разработка диаграммы компонентов и генерация программного кода в среде IBM Rational Rose

Цель работы: разработать диаграмму компонентов и сгенерировать программный код спроектированного программного обеспечения средствами IBM Rational Rose.

Диаграммы компонентов показывают, как выглядит модель на физическом уровне. На них изображены компоненты программного обеспечения и связи между ними. При этом на такой диаграмме выделяют два типа компонентов: исполняемые компоненты и библиотеки кода.

Каждый класс модели (или подсистема) преобразуется в компонент исходного кода. После создания они сразу добавляются к диаграмме компонентов. Между отдельными компонентами изображают зависимости, соответствующие зависимостям на этапе компиляции или выполнения программы. Пример диаграммы компонентов показан на рисунке 5

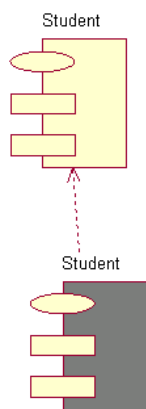


Рисунок 5 – Пример диаграммы компонентов.

Представление компонентов содержит:

- Компоненты, являющиеся физическими модулями кода.
- Диаграммы компонентов.
- Пакеты, являющиеся группами связанных компонентов.

Порядок выполнения работы

5. Для реализации построенной системы
 - а. постройте диаграмму компонентов
 - б. выполните проверку корректности модели
 - в. выполните генерацию кода,
6. Оформить отчет по лабораторной работе.
7. Представить отчет по лабораторной работе для защиты.

Порядок построения модели

Создание диаграммы компонентов:

1. Дважды щелкните мышью по главной диаграмме компонентов в представлении компонентов.
2. На панели инструментов нажмите кнопку Package Specification.
3. Поместите спецификацию пакета на диаграмму.
4. Введите имя спецификации пакета и укажите в окне спецификации язык программирования для генерации кода.
5. На панели инструментов нажмите кнопку Package Body.
6. Поместите тело пакета на диаграмму.
7. Введите имя тела пакета и укажите в окне спецификации язык генерации кода
8. На панели инструментов нажмите кнопку Dependency.

9. Проведите линию зависимости от тела пакета к спецификации пакета.

Соотнесение классов с компонентами:

1. В логическом представлении браузера найдите необходимый для генерации класс.
2. Перетащите этот класс на спецификацию пакета компонента в представлении компонентов браузера. В результате класс будет соотнесен со спецификацией пакета компонента.

Процесс генерации кода состоит из четырех основных шагов:

1. Проверка корректности модели.
2. Установка свойств генерации кода.
3. Выбор класса, компонента или пакета.
4. Генерация кода.

Для проверки модели:

1. Выберите в меню Tools\Check Model.
2. Проанализируйте все найденные ошибки в окне журнала, используя команду View\Log.

К наиболее распространенным ошибкам относятся такие, например, как сообщения на диаграмме последовательности или кооперативной диаграмме, не соотнесенные с операцией, либо объекты этих диаграмм, не соотнесенные с классом.

С помощью пункта меню Check Model можно выявить большую часть неточностей и ошибок в модели. Пункт меню Access Violations позволяет обнаруживать нарушения правил доступа, возникающие тогда, когда существует связь между двумя классами разных пакетов, но связи между самими пакетами нет.

Для того чтобы обнаружить нарушение правил доступа:

1. Выберите в меню Report\Show Access Violations.
2. Проанализируйте все нарушения правил доступа в окне.

Можно установить несколько параметров генерации кода для классов, атрибутов, компонентов и других элементов модели. Этими свойствами определяется способ генерации программ. Для каждого языка в Rose предусмотрен ряд определенных свойств генерации кода. Перед генерацией кода рекомендуется анализировать эти свойства и вносить необходимые изменения.

Для анализа свойств генерации кода

1. выберите Tools\Options, а затем вкладку соответствующего языка.
2. в окне списка можно выбрать класс, атрибут, операцию и другие элементы модели.

Для каждого языка в этом списке указаны свои собственные элементы модели. При выборе разных значений на экране появляются разные наборы свойств.

Для изменения свойства генерации кода для одного класса, атрибута, одной операции и т.д.

нужно

1. открыть окно спецификации элемента модели. Выбрать вкладку языка (C++, Java,...) и изменить свойства. Все изменения, вносимые в окне спецификации элемента модели, оказывают влияние только на этот элемент.

При генерации кода за один раз можно создать класс, компонент или целый пакет. Код генерируется с помощью диаграммы или браузера. При генерации кода из пакета можно выбрать или пакет логического представления на диаграмме классов, или пакет представления компонентов на диаграмме компонентов. При выборе пакета логического представления генерируются все классы этого пакета. При выборе пакета представления компонентов генерируются все компоненты этого пакета.

После выбора класса или компонента на диаграмме выберите в меню соответствующий вариант генерации кода. Сообщения об ошибках, возникающих в процессе генерации кода, будут появляться в окне журнала.

Во время генерации кода Rose выбирает информацию из логического и компонентного представлений модели и генерирует большой объем «скелетного» (skeletal) кода:

Генерация кода

3. Откройте диаграмму компонентов системы.
4. Выберите все объекты на диаграмме компонентов.

5. Выберите Tools\(\язык для генерации кода)\Code Generation в меню.
6. Выполните генерацию кода.
7. Просмотрите результаты генерации (меню Tools\(\язык для генерации кода)\ Browse Header и Tools\(\язык для генерации кода)\Browse Body.

Варианты заданий

Номер варианта	Задание
1.	Опишите процесс организации работы детективного агентства с точки зрения ее работников. Разработать программный модуль «Детективное агентство», содержащий сведения о клиентах агентства и об оказанных услугах. Программный модуль предназначен для учета средств за оказанные услуги.
2.	Опишите процесс работы музея с точки зрения его служащего. Разработать программный модуль «Музей», предназначенный для использования работниками музея. В базе содержатся сведения об экспонатах музея и вносятся данные при поступлении новых экземпляров. При выполнении инвентаризации данные заносятся в базу, проводится сверка и выдаются отчеты по учету экспонатов в музее.
3.	Опишите процесс работы книжного магазина с точки зрения его служащего. Разработать программный модуль «Книжный магазин», содержащий сведения о книгах (автор, название, издательство, год издания, цена). Покупатель оформляет заявку на нужные ему книги, если таковых нет, он заносится в базу и оповещается, когда нужные книги поступают в магазин.
4.	Опишите процесс работы автостоянки с точки зрения ее служащего. Разработать программный модуль «Автостоянка». В программе содержится информация о марке автомобиля, его владельце, дате и времени въезда, стоимости стоянки, скидках, задолженности по оплате и др.
5.	Опишите процесс организации работы гостиницы с точки зрения администратора. Разработать программный модуль «Гостиница», содержащий сведения о наличии свободных мест и о проживающих в гостинице. Программный модуль предназначен для бронирования мест в гостинице и оформления проживающих.
6.	Опишите процесс работы АТС с точки зрения ее служащего. Разработать программный модуль «Картотека абонентов АТС». Картотека содержит сведения о телефонах и их владельцах. Фиксирует задолженности по оплате (абонентской и временной). Считается, что временная оплата местных телефонных разговоров уже введена.
7.	Опишите процесс организации работы с нарушителями правил дорожного движения с точки зрения работника полиции. Разработать программный модуль «Учет нарушений правил дорожного движения». Для каждой автомашины (и ее владельца) в базе хранится список нарушений. Для каждого нарушения фиксируется дата, время, вид нарушения и размер штрафа. При оплате всех штрафов машина удаляется из базы.
8.	Опишите процесс работы химчистки с точки зрения ее служащего. Разработать программный модуль «Химчистка». При записи на обслуживание заполняется заявка, в которой указываются ФИО владельца, описание изделия, вид услуги, дата приема заказа и стоимость услуги. После выполнения работ распечатывается квитанция.
9.	Опишите процесс организации рабочего дня руководителя с точки зрения его секретаря. Разработать приложение «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение

	предназначено для организации рабочего дня руководителя.
10.	Создать информационную систему, содержащую сведения о деятельности турагенств продающих путевки в разные страны, курорты, отели и, включая характеристики этих отелей.
11.	Программное средство Call-центра технической поддержки сотового оператора, обеспечивающее клиента (актёр) функциями связи с менеджером (актёр), выдачи баланса, консалтинга по тарифам, подключения / отключения услуг и не менее 5 других функций. Функция со справкой по тарифам должна предполагать внутреннее голосовое меню.
12.	Программное средство банкомата, выполняющего только функцию выдачи денег. Взаимодействуют клиент, система обслуживания, купюроприёмник, кнопочная панель, определяющая снимаемую сумму и кнопки подтверждения или отмены. Функции клиента – вставить карту, ввести код, запросить баланс, набрать сумму, получить деньги. Функции система обслуживания – получать и распознавать сигналы с датчиков кнопок, выводить на кран распознанные цифры, выводить на экран необходимые приглашения и сообщения (на схеме они должны быть расшифрованы), анализировать наличие в банкомате средств, отдавать в купюроприёмник информацию о нужном количестве денег на выдачу, формировать и распечатывать выписку, функция купюроприёмника – выдача денег; функции кнопок – формирование соответствующих сигналов нажатия.
13.	Программное средство Call-центра салона красоты, обеспечивающее взаимодействие клиента, менеджера и базы учёта. Call-центр выдаёт стандартное приветствие и предоставляет клиенту голосовое меню, с помощью которого он может получить информацию об услугах салона, записаться на сеанс к стилисту, вейзажисту, массажисту, парикмахеру и т.д., связаться с менеджером, отменить, изменить время сеанса, записаться на повторный приём, получить справку. Менеджер может напрямую работать с базой учёта, внося туда сведения о клиентах и мастерах.
14.	Программное средство формирования командировочного удостоверения преподавателя вуза. Программное средство управляется бухгалтером. Преподаватель заполняет электронную анкету с типом командировки (по России или международная), местом назначения, временем пребывания, целью и способом проезда к месту командировки. Руководство вуза выдаёт (или не выдаёт) разрешение на командировку, в результате чего на основе анкетных данных преподавателя и реквизитов вуза, полученных из учётной базы данных, формируется проект приказа на командировку преподавателя, который получает преподаватель и предоставляет бухгалтеру. Бухгалтер оценивает по справочникам примерную стоимость проезда, проживания и величину суточных расходов и формирует величину аванса, который начисляется программным средством на зарплатную карточку преподавателя. По истечении командировки преподаватель предоставляет бухгалтеру чеки, подтверждающие использование аванса. Если потрачено меньше, чем запланировано, то осуществляется удержание с преподавателя разницы, если больше – то доначисление.
15.	Программное средство библиотечного каталога вуза, взаимодействующее со студентами, библиотекарями и преподавателями. Функции преподавателей – поиск учебников, заказ на закупку учебников; функции библиотекарей – внесение (исключение) записей в базу данных (или подтверждение по запросам студентов через интернет), распределение полномочий; функции студентов – алфавитный, тематический и не менее 2-х видов других поисков, запрос электронной версии, запрос на заказ учебника, запрос на справку, связь

	с библиотекарем через интернет или телефон.
16.	Разработать программное средство для регистратуры детской стоматологической клиники, имеющее агентов: секретарь, врачи и клиенты. Функции секретаря – внесение, исключение записей в базе данных (или подтверждение по запросам клиентов через интернет), запрос расписания врачей, определение времени приёма по справочнику, исключение или перенос записей по требованию врача с уведомлением клиентов, связь с врачом. Функции врача – анализ заявок на обслуживание и формирование предложений по оптимизации записей при наличии ошибок или иным обстоятельствам, заполнение карточки по результатам приёма; функции клиентов – поиск информации о расписании врачей, получение информации об опыте, образовании и достижениях врачей, запись на приём (по телефону, лично у секретаря или через интернет), запрос на справку у врача, связь с секретарём через интернет или телефон.
17.	Разработать программное средство регистрации клиентов в гостинице, включающее агентов: администратор, персонал, клиент. Функции администратора: внесение паспортных и анкетных данных клиента, получение сведений о свободном номерном фонде, бронирование или немедленное предоставление ключа доступа в комнату клиента, выписка клиента, выдача справок клиенту, приём заявок на вызов персонала гостиницы (горничные, плотники, слесаря и т.д.); функции персонала – регистрация времени посещения клиента и сделанных работ; функции клиента – бронирование по телефону, лично у администратора или через интернет номера в гостинице; заполнение через интернет анкеты для заселения или передача информации администратору, получение справки у администратора, связь с администратором, получение карты доступа в комнату, использование карты доступа в номер.
18.	Опишите процесс работы автомагазина с точки зрения его служащего. Разработать программный модуль «Картотека автомагазина», предназначенный для использования работниками магазина. В базе содержатся сведения об автомобилях (марка, объем двигателя, дата выпуска и др.). При поступлении заявки на покупку производится поиск подходящего варианта. Если такого нет, клиент заносится в клиентскую базу и оповещается, когда вариант появляется.
19.	Опишите процесс организации работы автостанции с точки зрения ее служащего. Разработать программный модуль «Автокасса», содержащий сведения о наличии свободных мест на автобусные маршруты. В базе должны содержаться сведения о номере рейса, маршруте, водителе, типе автобуса, дате и времени отправления, а также стоимости билетов. При поступлении заявки на билеты программа производит поиск подходящего рейса.
20.	Опишите процесс учета посещения студентов учебных занятий и успеваемости студентов с точки зрения работника деканата. Разработать программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию деканом, заместителями декана и сотрудниками деканата. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

Пример выполнения лабораторной работы № 1

Ознакомьтесь с окном Rational Rose (рис. 1):

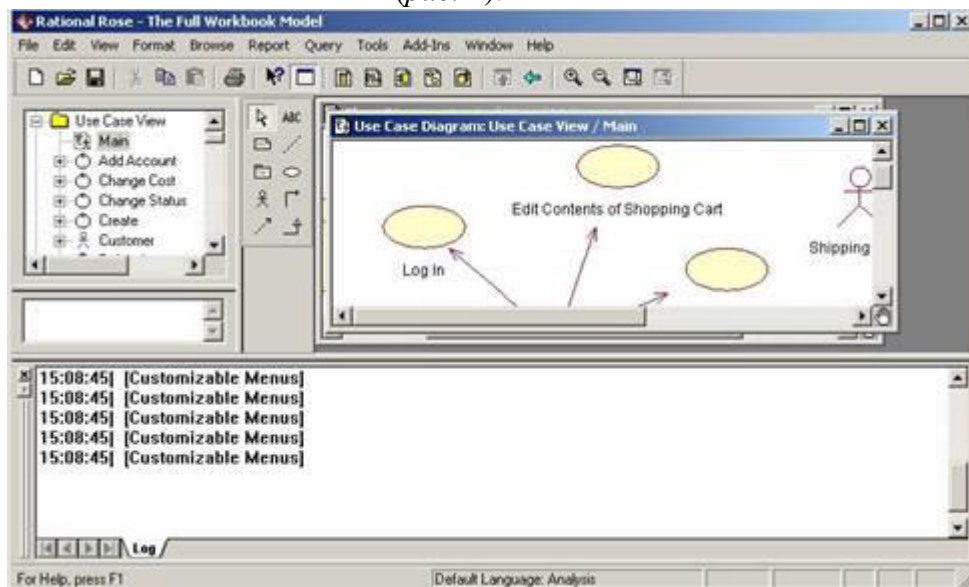


Рис. 1. Окно Rational Rose

- *браузер программы* - располагается под строкой меню и строкой инструментов в левой части окна приложения. Браузер (навигатор) является инструментом навигации по иерархической структуре модели. Знак + рядом с папкой означает, что в ней имеются дополнительные элементы;

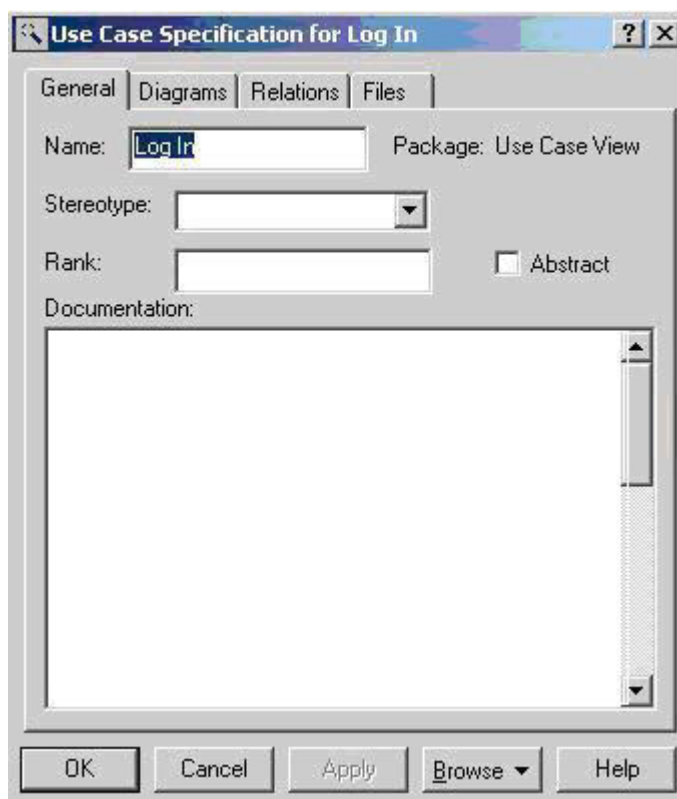


Рис. 2. Окно спецификаций

- *панель "инструменты диаграммы"* - с права от окна браузера. Она меняется в зависимости от выбранной диаграммы;
- *окно диаграммы*, используется для создания, отображения и изменения диаграмм на языке UML. В нем находятся разные диаграммы. Активная диаграмма имеет синий заголовок;
- *окно спецификации* – позволяет задавать характеристики элемента диаграммы (рис.2);

- окно документации — окно для вывода словесного описания данного элемента. Описание можно вводить и в поле Documentation окна спецификации.

Как показано на *рис. 3*, кнопки строки инструментов позволяют выполнять стандартные и специальные действия.



Рис. 3. Строка инструментов.

Ознакомьтесь с постановкой задачи.

Постановка задачи:

Построить модель Университетской системы для регистрации учебных курсов. Система регистрации учебных курсов используется:

- *профессором* — для задания читаемого курса;
- *студентом* — для выбора изучаемого курса;
- *регистратором* — для формирования учебного плана и расписания;
- *учетной системой* — для определения денежных затрат.

Постановка задачи:

Построить модель Университетской системы для регистрации учебных курсов. Система регистрации учебных курсов используется:

- *профессором* — для задания читаемого курса;
- *студентом* — для выбора изучаемого курса;
- *регистратором* — для формирования учебного плана и расписания;
- *учетной системой* — для определения денежных затрат.

Построение диаграммы Use Case

Диаграмма Use Case строится с целью определения объектов системы и их взаимодействия. Они представляются актерами, элементами Use Case и отношениями между ними.

Первый шаг построения этой диаграммы состоит в определении *актеров*, фиксирующих роли внешних объектов, взаимодействующих с системой. В рассматриваемом примере можно выделить 4 актера: *Student* (*Студент*), *Professor* (*Профессор*), *Registrar* (*Регистратор*) и *Billing System* (*Учетная система*) (*рис. 5*).

Элемент Use Case представляет определенную часть функциональности, обеспечиваемой системой при взаимодействии с ней каждого актера.

Откройте главную диаграмму Use Case (*рис. 4*).

В окне браузера щелкните по значку + слева от пакета Use Case View.

Откройте диаграмму, выполнив двойной щелчок по значку Main.

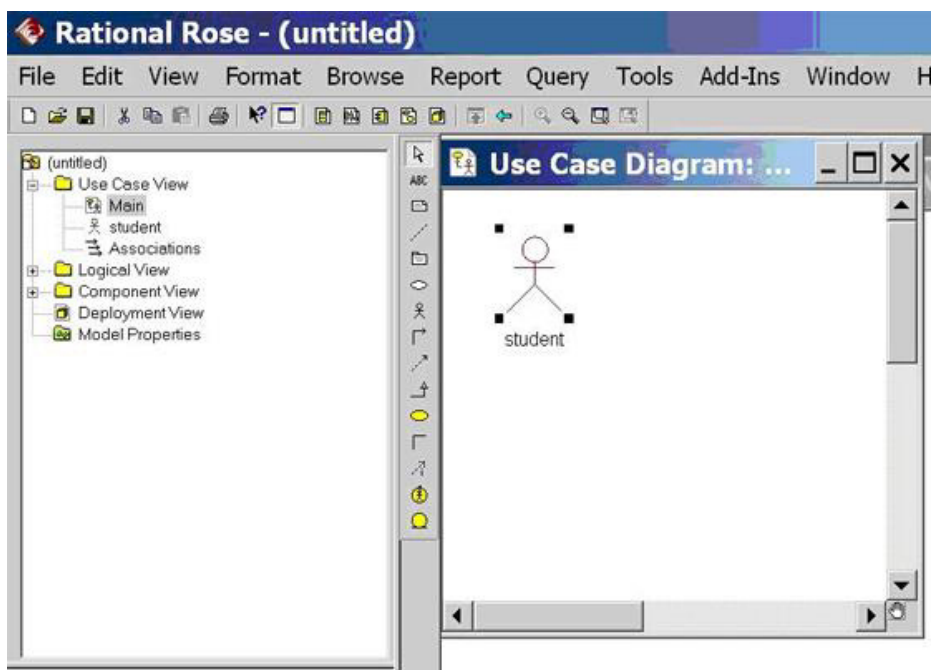


Рис. 4. Главная диаграмма Use Case

Проблемная область Университетской системы взаимодействует с 4 актерами:

1. *Student (Студент)* – регистрируется на курсы (Register for Courses);
2. *Professor (Профессор)*. Professor запрашивает список курса (Request a Course Roster);
3. *Registrar (Регистратор)*. Registrar управляет учебным планом (Manage Curriculum);
4. *Billing System (Учетная система)* - получает информацию о регистрации (рис. 5).

Главная диаграмма прецедентов

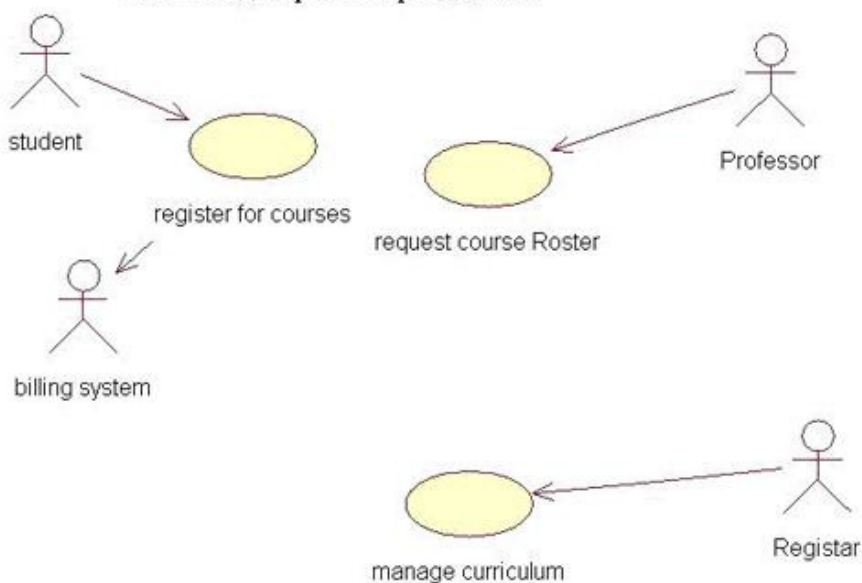


Рис. 5. Актеры и элементы Use Case

Последовательно создайте 4 актеров:

- щелкните по значку актера (человечек) на панели инструментов;
- щелкните в нужном месте диаграммы для добавления актера в диаграмму;
- введите имя Student (Студент), пока актер остается выделенным;
- повторите предыдущие шаги для ввода трех других актеров (Professor, Registrar и Billing System — Профессор, Регистратор, Учетная система).

Определите для каждого актера соответствующие элементы Use Case:

- добавьте элемент Use Case в диаграмму щелчком по значку элемента Use Case и затем щелкните в нужном месте диаграммы;

- введите имя (Register for Courses), пока элемент Use Case остается выделенным;
- повторите предыдущие шаги для ввода других элементов Use Case (Request Course Roster, Manage Curriculum).

Отобразите отношения между актерами и элементами Use Case, указав направление взаимодействия (кто инициирует взаимодействие), используя однонаправленные стрелки (unidirectional arrows):

- щелкните по значку однонаправленной ассоциации (стрелке) на панели инструментов;
- щелкните по актеру Student и переместите линию на элемент Use Case Register for Courses.

Повторите предыдущие шаги для ввода других отношений:

- от актера Professor к элементу Use Case Request Course Roster;
- от актера Registrar к элементу Use Case Manage Curriculum

Комментарий:

В системе регистрации курсов:

- актер Student инициирует элемент Use Case Register for Courses, этот же элемент, в свою очередь, взаимодействует с актером Billing System;
- актер Professor инициирует элемент Use Case Request Course Roster;
- актер Registrar инициирует элемент Use Case Manage Curriculum (рис. 5).

Диаграммы последовательностей

Функциональность элемента Use Case отображается графически в диаграмме последовательности (Sequence Diagram). Она отображает один из возможных путей в потоках событий элемента Use Case — например, добавление студента к курсу.

Диаграммы последовательности содержат объекты и сообщения между объектами, которые показывают реализацию поведения.

Сценарий предполагает, что студент должен заполнить информацией (fill in info) регистрационную форму (registration form), а затем форма предьявляется на рассмотрение (submitted).

Очевидно, что необходим объект *registration form*, который принимает информацию от студента.

Создайте диаграмму последовательности для элемента Use Case Register for Courses (рис. 6):

- Щелкните правой кнопкой по элементу Use Case Register for Courses в окне браузера;
- Выберите команду New: Sequence Diagram в появившемся контекстном меню;
- Убедитесь в добавлении в дерево окна браузера диаграммы последовательности с именем New Diagram;

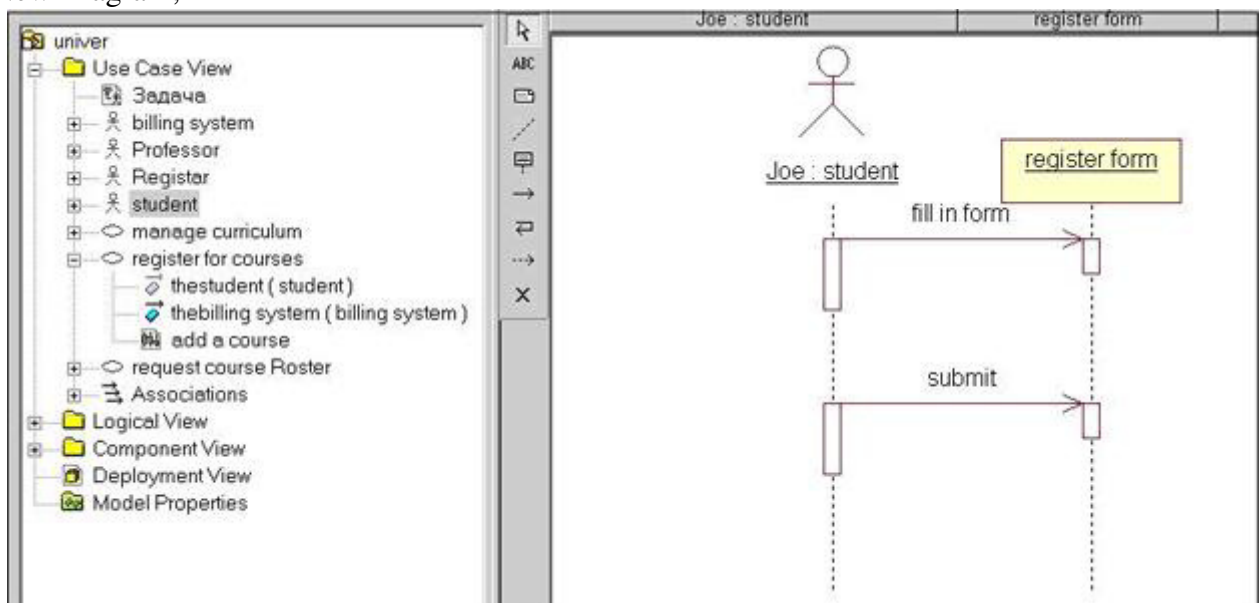


Рис. 6. Диаграмма последовательностей

- Введите имя диаграммы Add a Course, пока значок новой диаграммы остается выделенным (рис. 6);
- Откройте диаграмму двойным щелчком по ее значку в окне браузера;
- Переместите актера Student в окно диаграммы (он инициирует сценарий);
- Присвойте экземпляру актера конкретное имя: Joe;
- Создайте объект регистрационная форма (registration form):
 - щелкните сначала по значку объекта (прямоугольнику) на панели инструментов и затем в нужном месте диаграммы;
 - введите имя registration form, пока объект остается выделенным;
- Создайте объектные сообщения "fill in info" (форма должна заполняться студентом) и "submit":
 - щелкните по значку объектного сообщения (стрелке) на панели инструментов;
 - щелкните по пунктирной линии под актером Student и переместите стрелку на пунктирную линию под объектом registration form;
 - введите имя сообщения — fill in information, пока стрелка остается выделенной;
 - повторите предыдущие шаги пункта 9 для создания сообщения submit.

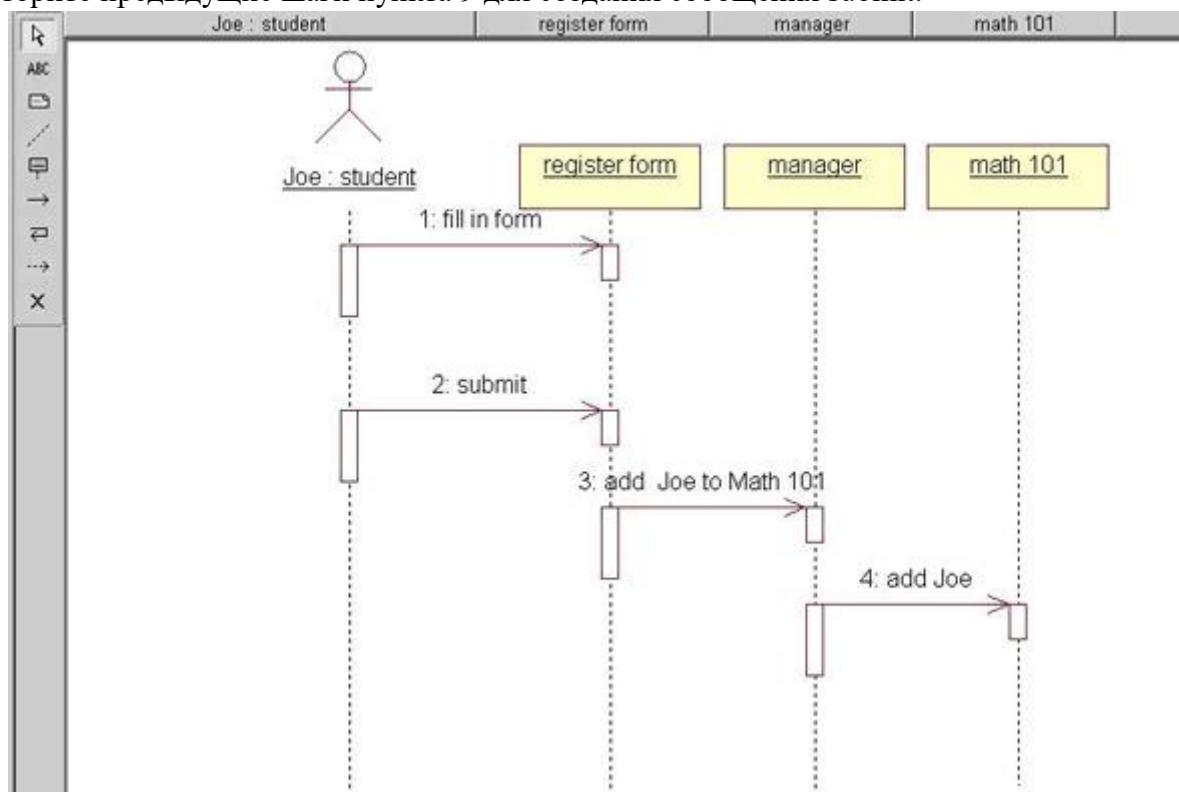


Рис. 7. Добавление на диаграмму объектов и сообщений

Объект Регистрационная форма является промежуточным звеном в цепи передачи информации;

- добавьте в поле диаграммы объект-менеджер (Manager);
- создайте сообщение менеджеру (manager) (Рис. 7). Менеджер узнает, что студент должен быть добавлен к какому-то курсу, (например, Джо хочет подписаться на курс math 101):
 - щелкните по значку объекта (прямоугольнику) на панели инструментов;
 - щелкните в нужном месте диаграммы для добавления объекта в диаграмму;
 - введите имя manager, пока объект остается выделенным;
 - щелкните по значку объектного сообщения (стрелке) на панели инструментов;
 - щелкните по пунктирной линии под объектом registration form и переместите стрелку на пунктирную линию под объектом manager;
 - введите имя сообщения — add Joe to Math 101 (рис. 7), пока стрелка остается выделенной.

Менеджер — один из объектов системы. Он взаимодействует как с регистрационной формой, так и с набором объектов — учебных курсов. Для обслуживания студента Джо

должен быть объект Math 101. Если такой объект существует, то Менеджер обязан передать объекту-курсу Math 101, что Джо должен быть добавлен к курсу;

- добавьте на диаграмму объекты и сообщения таким же образом, как в п. 8 - 11 в соответствии с *рис. 7*.

Объект-курс не принимает самостоятельных решений о возможности добавления студентов. Этим занимается экземпляр класса Предложение курса (*course offering*). Назовем такой экземпляр (объект) Section 1.

Объект-курс обращается к объекту Section 1, если он открыт (в этом сценарии — ответ "да") с предложением добавить студента Джо (*рис. 8*).

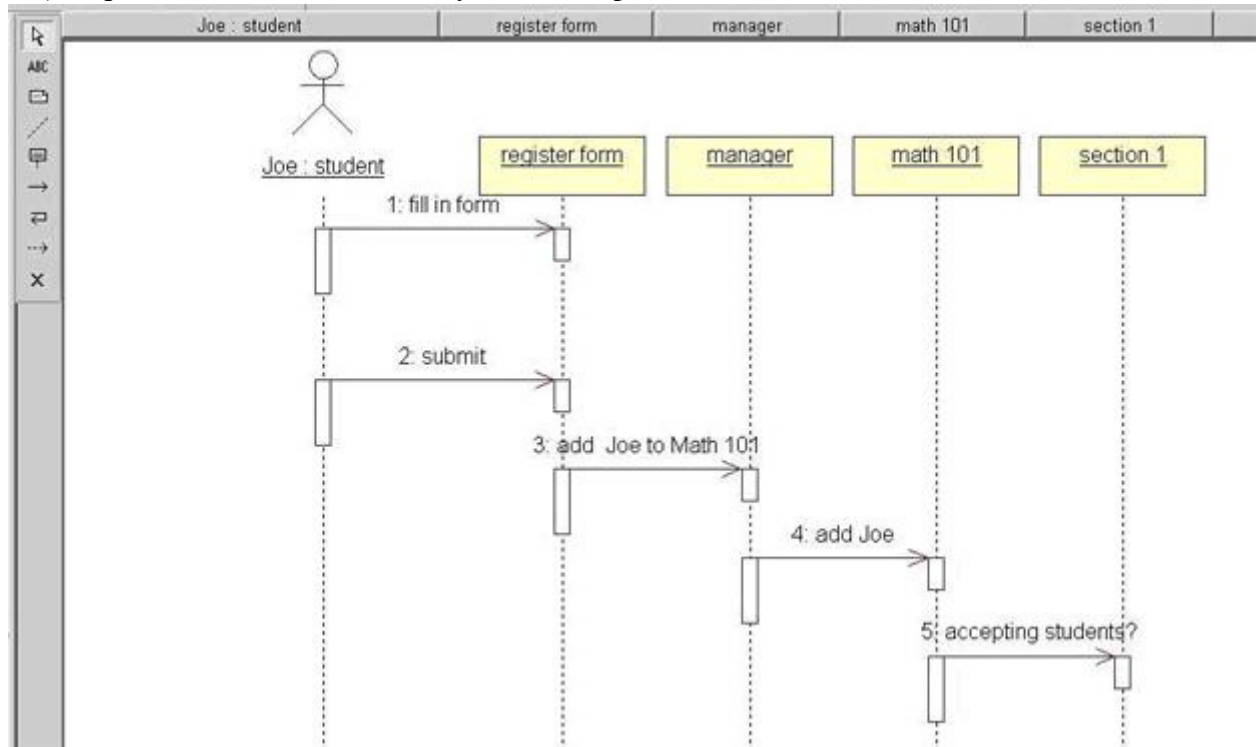


Рис. 8. Создание диаграммы последовательностей

- щелкните по значку объекта (прямоугольнику) на панели инструментов;
- щелкните в нужном месте диаграммы для добавления объекта;
- введите имя — *section 1*, пока объект остается выделенным;
- щелкните по значку объектного сообщения (стрелке) на панели инструментов;
- щелкните по пунктирной линии под объектом *math 101* и переместите стрелку на пунктирную линию под объектом *section 1*;
- введите имя сообщения — *accepting students*, пока стрелка остается выделенной;
- повторите шаги 4-6 для создания сообщения *add Joe*.

После того, как Менеджер убеждается в том, что Студенту Джо предоставляется возможность изучать курс *math 101*, он уведомляет Учетную систему (*billing system*) о необходимости выписки счета.

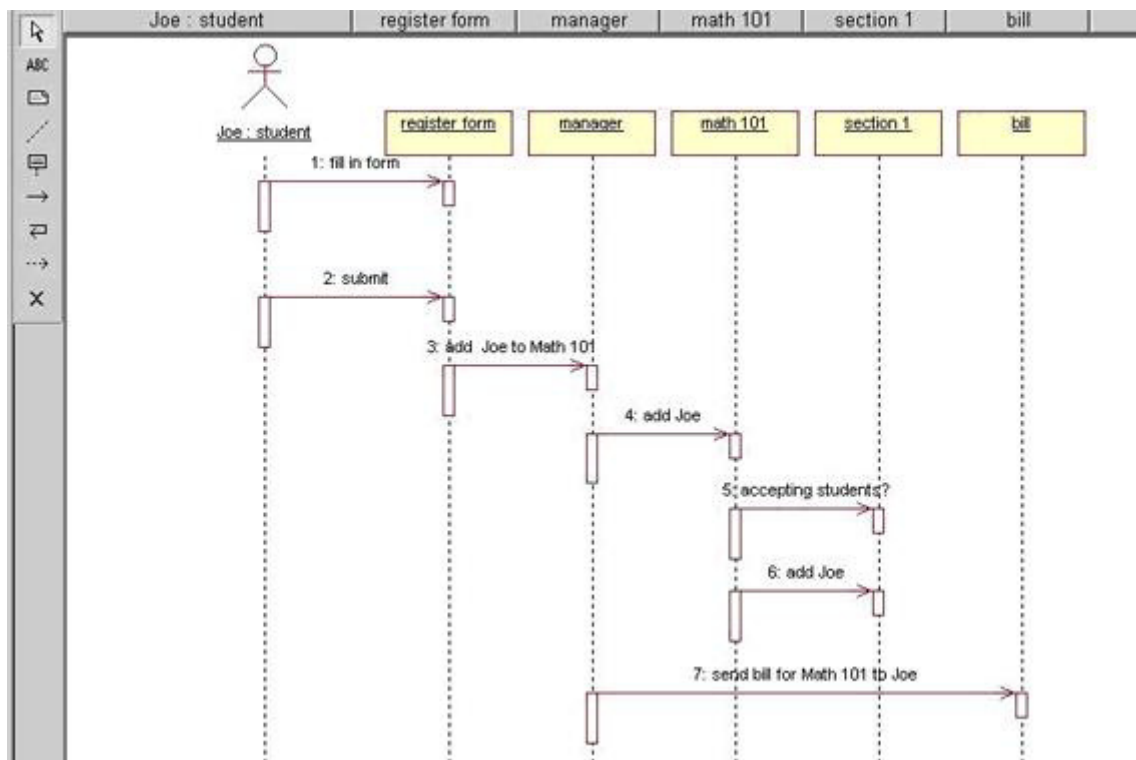


Рис. 9. Диаграмма последовательностей Add a Course

- щелкните по значку объекта (прямоугольнику) на панели инструментов;
- щелкните в нужном месте диаграммы для добавления объекта в диаграмму;
- введите имя — **bill**, пока объект остается выделенным;
- щелкните по значку объектного сообщения (стрелке) на панели инструментов;
- щелкните по пунктирной линии под объектом **manager** и переместите стрелку на пунктирную линию под объектом **bill**;
- пока стрелка остается выделенной, введите имя сообщения — **send bill for Math 101 to Joe**,
- сравните результат своей работы с *рис. 9*.

Диаграмма классов

Объекты из диаграмм последовательности группируются в классы. Из созданной в предыдущих заданиях диаграммы последовательности можно идентифицировать следующие объекты и классы:

registration form - является объектом класса *RegForm*;

manager является объектом класса *Manager*;

math 101 является объектом класса *Course*;

section 1 является объектом класса *CourseOffering*;

bill является интерфейсом к внешней учетной системе, имя его класса *BillingSystem*.

Классы создаются в логическом представлении системы.

- В окне браузера щелкните правой кнопкой по значку пакета **Logical View** (рис. 10).

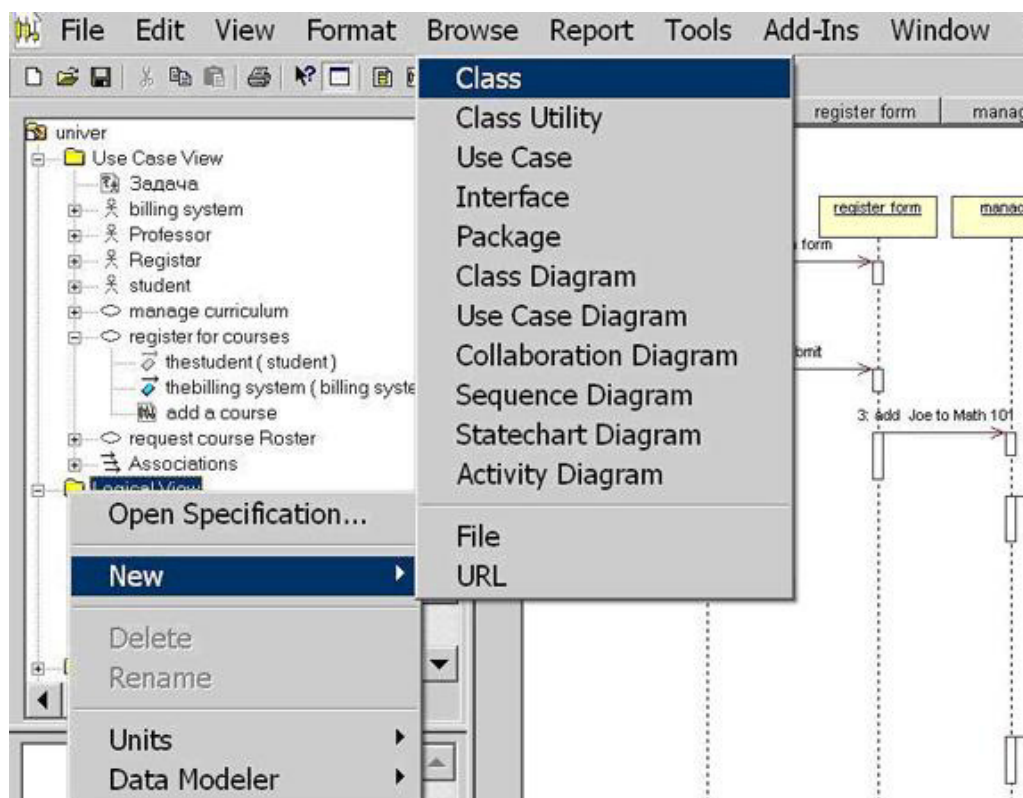


Рис. 10. Окно создания класса

- Выберите команду New Class в появившемся контекстном меню. В результате в дерево окна браузера будет добавлен класс с именем NewClass.
- Введите имя RegForm, пока значок класса остается выделенным.
- Повторите предыдущие шаги для добавления других классов: Manager, Course, CourseOffering и BillingSystem.

После создания классов они описываются (документируются). Описания добавляются с помощью Documentation Window (рис. 11).

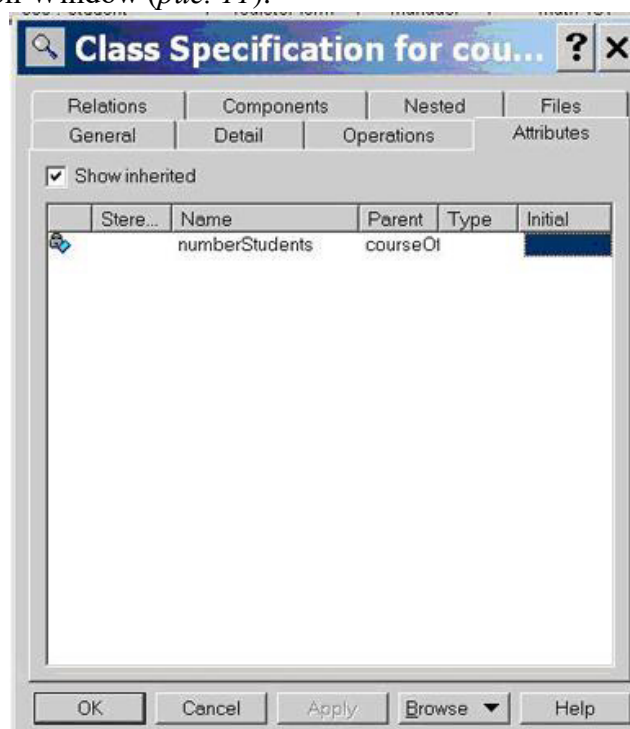


Рис. 11. Описание классов

- В окне браузера щелкните по значку класса CourseOffering, введите описание класса в Documentation Window.

Добавление классов:

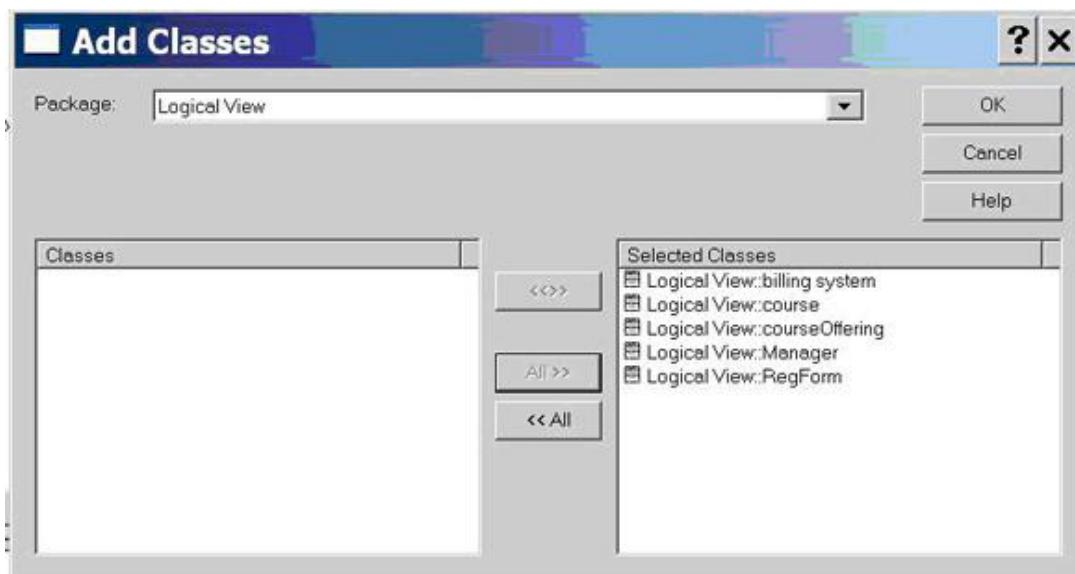


Рис. 12. Добавление классов.

- Откройте главную диаграмму классов двойным щелчком по значку Main в окне браузера.
- Добавьте на нее классы, выбрав в главном меню команду Query: Add Classes (рис. 12). Для этого нажмите кнопку “ All ” (выбрать все) (рис. 12).
- Закройте окно нажав, кнопку OK.
- Переупорядочите классы в диаграмме, выделяя конкретный класс и перетаскивая его на новое место (рис. 13).

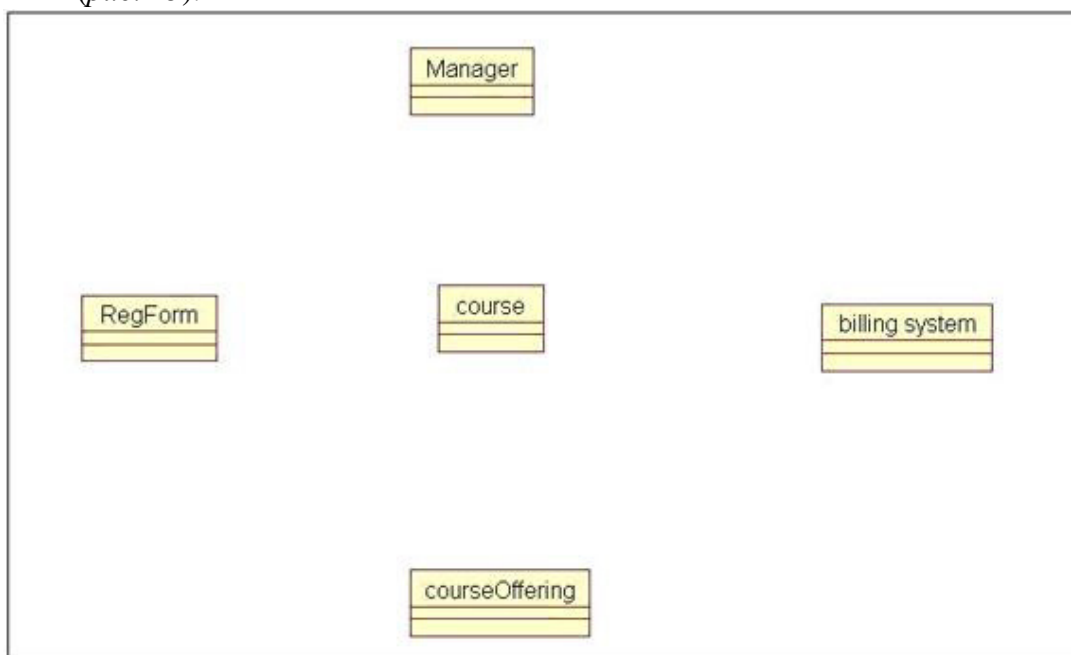


Рис. 13. Добавление классов в диаграмму

ПРИМЕЧАНИЕ.

Классы можно добавлять в диаграмму перетаскиванием их из окна браузера (по одному классу). Для создания новых типов моделирующих элементов в UML используется понятие стереотипа. С помощью стереотипа можно «нагрузить» элемент новым смыслом.

- Использование предопределенного класса
Класс BillingSystem определяет только интерфейс к внешней учетной системе (billing system). Для использования предопределенного стереотипа Interface для этого класса (рис. 14):

- выполните двойной щелчок по значку класса BillingSystem в главной диаграмме классов. Появляется окно спецификации класса (Class Specification);
- щелкните по стрелке раскрывающегося списка Stereotype (Рис. 14);
- наберите на клавиатуре слово-стереотип Interface;
- закройте окно спецификации, нажав кнопку OK.

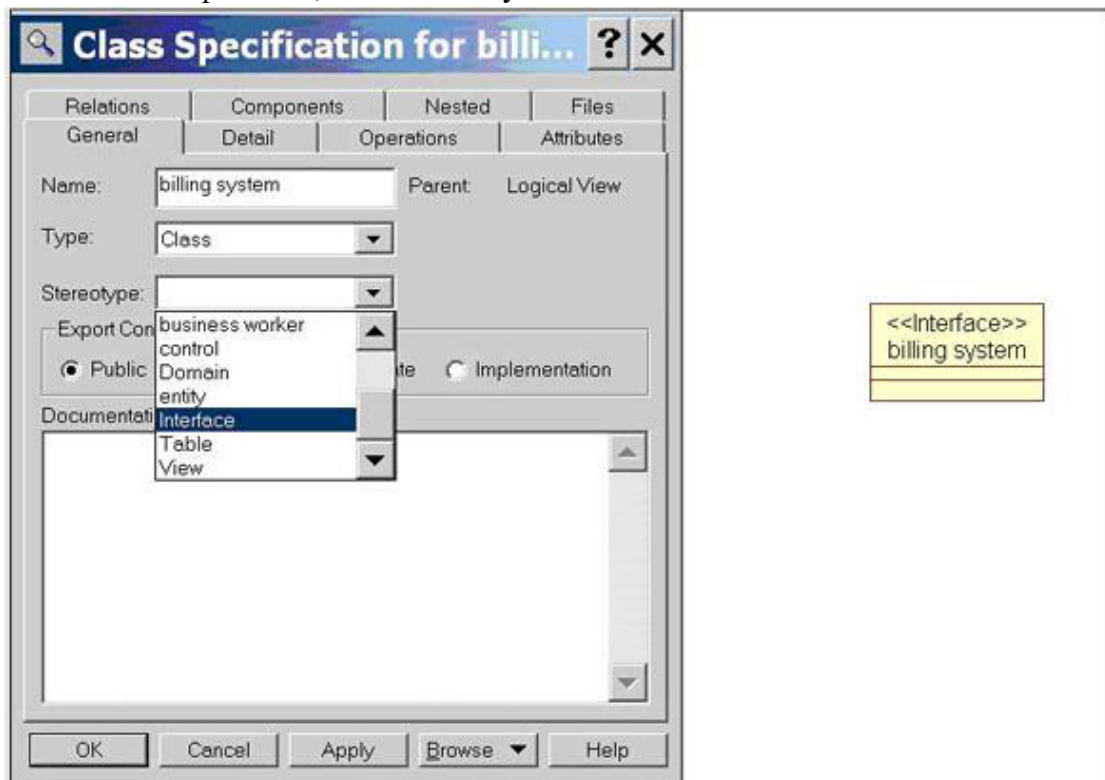


Рис. 14. Предопределенный класс

Если есть взаимодействия между объектами, определенные на диаграмме последовательностей, то должен быть определен путь для коммуникации между их классами.

Структурные отношения определяются двумя типами:

1. ассоциациями
2. агрегациями.

Ассоциация определяет соединение между классами.

- Создайте на диаграмме последовательности Add a Course следующие ассоциации: от RegForm к Manager, от Manager Course и от Manager к BillingSystem (рис. 16). Для этого:
 - щелкните по значку однонаправленной ассоциации (стрелке) на панели инструментов;
 - щелкните по классу RegForm и переместите линию ассоциации на класс Manager;
 - повторите предыдущие шаги для ввода следующих отношений:
 - от Manager к Course;
 - от Manager к BillingSystem.

Ассоциации задают пути между объектами-партнерами одинакового уровня.

Агрегация фиксирует неравноправные связи. Она показывает отношение между целым и его частями.

- Создайте отношение агрегации между классом Course и классом CourseOffering (рис. 15), т. к. предложение Курса CourseOfferings является частью агрегата — класса Course:
 - щелкните по значку агрегации (линии с ромбиком) на панели инструментов;
 - щелкните по классу, представляющему целое — Course;
 - переместите линию агрегации на класс, представляющий часть — CourseOffering.

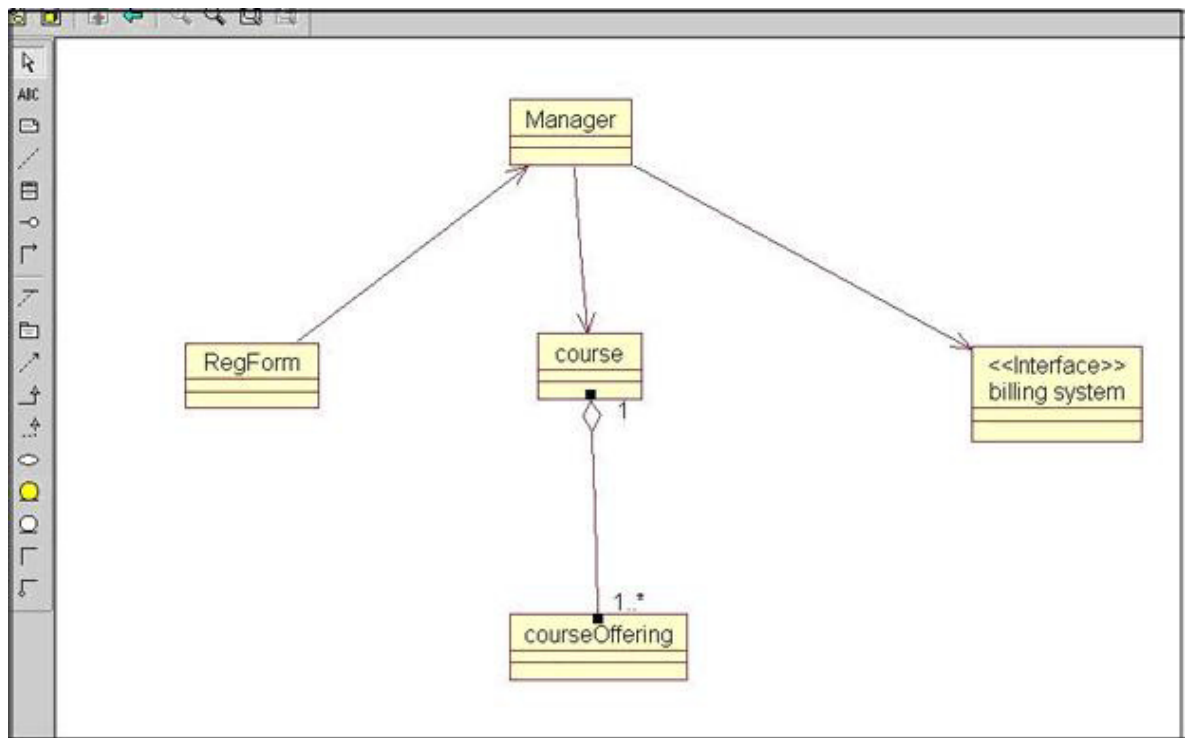


Рис. 15. Агрегация и ассоциация

- Добавьте на диаграмму индикаторы мощности (рис. 15) для отображения того, какое количество объектов участвует в отношении:
 - щелкните правой кнопкой по линии агрегации возле класса CourseOffering;
 - выберите из контекстного меню команду Multiplicity One or More;
 - щелкните правой кнопкой по линии агрегации возле класса Course;
 - из контекстного меню выберите команду Multiplicity.

Определение класса начинается с задания имени. Любой класс должен инкапсулировать в себе структуру данных и поведение, определяющее возможности обработки этой структуры. Второй шаг ориентируется на фиксацию структуры, а третий — на фиксацию поведения.

Структура класса представляется набором его свойств. Структура находится путем исследования проблемных требований и соглашений между разработчиками и заказчиками.

Каждое предложение курса (CourseOffering) в нашей модели является свойством (attribute) класса-агрегата Course. Класс CourseOffering тоже имеет свойства.

- Определите одно из свойств класса CourseOffering — количество студентов:
 - щелкните правой кнопкой по классу CourseOffering в диаграмме классов;
 - выберите из контекстного меню команду Insert New Attribute. Это приведет к добавлению в класс свойства;
 - введите его имя — numberStudents, пока новое свойство остается выделенным.
- Задайте поведение класса (третий шаг).

Поведение класса представляется набором его операций. Исходная информация об операциях класса находится в диаграммах последовательности. В операции отображаются сообщения из диаграмм последовательности. Необходимо привязать объекты (из диаграмм после довательности) к конкретным классам.

Для этого:

- откройте диаграмму последовательности Add a Course двукратным щелчком по ее значку в окне браузера;
- щелкните в окне браузера по значку класса CourseOffering;
- переместите класс CourseOffering на объект section 1;
- отметьте, что имя объекта удлинилось, в нем появились две части, разделенные двоеточием. Слева от двоеточия записывается имя объекта, а справа — имя класса (рис. 16).

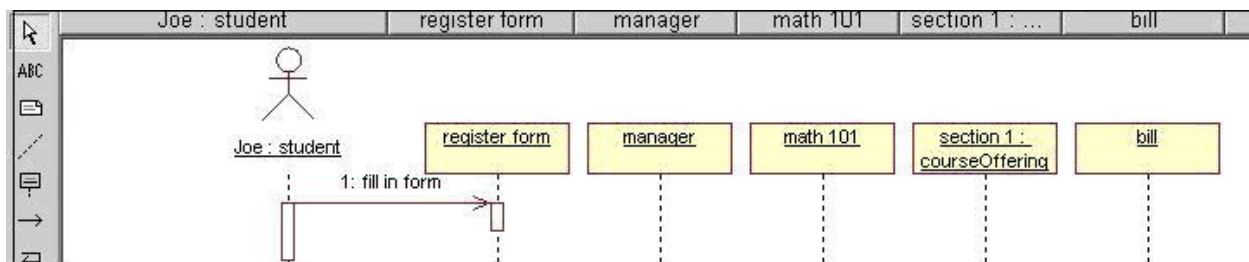


Рис. 16. Привязка классов

- Выполните второе действие — наполнение класса операциями. Как правило, в операции класса превращаются сообщения, получаемые его объектом. При этом сообщения переименовываются — производится согласование имени сообщения и имени операции:
 - имя операции должно отражать ее принадлежность к классу (а не к источнику соответствующего сообщения);
 - имя операции должно указывать на ее обязанность, а не на способ ее реализации;
 - имя должно быть допустимым с точки зрения синтаксиса языка программирования, который будет использоваться для кодирования класса.

Создайте новую операцию класса и свяжите с ней сообщение — оно автоматически поменяет имя;

- щелкните правой кнопкой по сообщению "add Joe". В результате станет видимым контекстное меню;
- выберите команду new operation. В результате станет видимой спецификация операции Operation Specification;
- введите имя новой операции — add;
- перейдите на вкладку Detail;
- щелкните правой кнопкой мышки по полю Arguments;
- выберите в контекстном меню команду Insert. В появившейся рамке наберите имя аргумента — Joe. Щелкните вне рамки;
- закройте окно спецификации, нажав кнопку ОК.

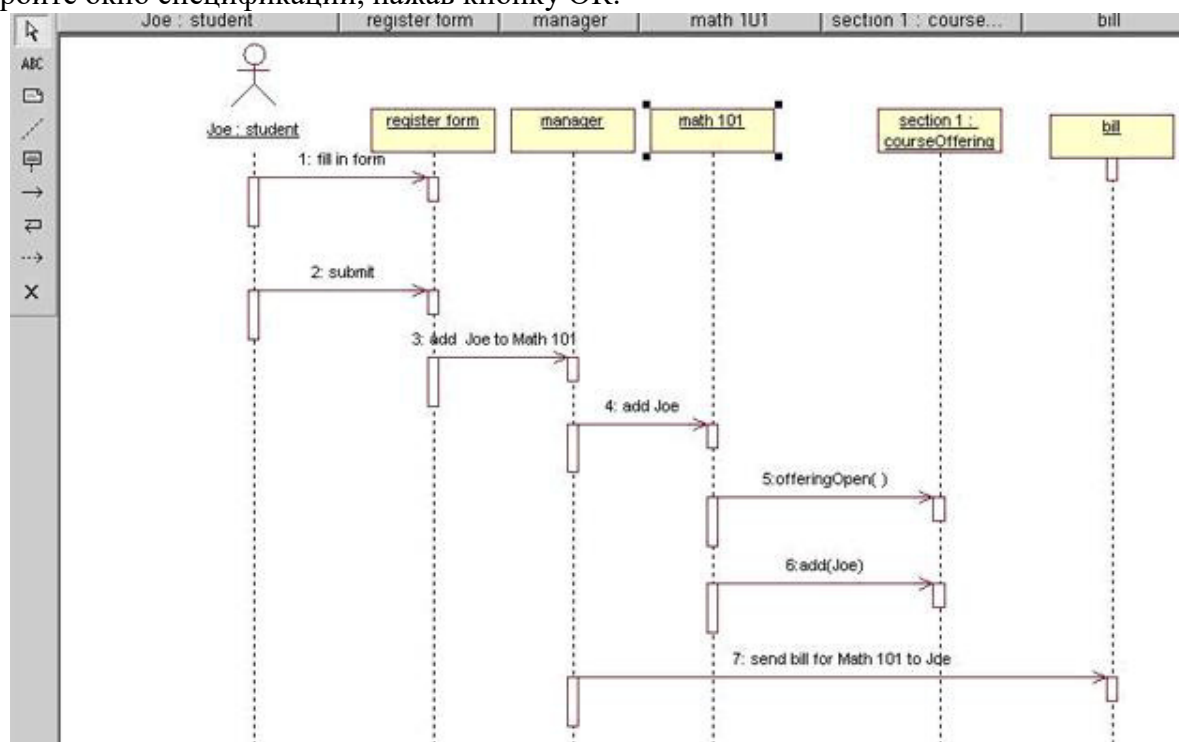


Рис. 17. Отображение операции на сообщение

- Получите имя сообщения обратным действием — отталкиваясь от имени операции (реализуется последовательность: отдельно создается новая операция класса, а затем она отображается на существующее сообщение) (рис. 17):

- щелкните правой кнопкой по классу в браузере;
- выберите в появившемся контекстном меню команду New: Operation. Появляется рамка с надписью opname;
- наберите имя новой операции класса — offeringOpen вместо надписи opname;
- щелкните правой кнопкой по сообщению "accepting students?" на диаграмме последовательности. В результате станет видимым контекстное меню;
- выберите в меню операцию offeringOpen() — сообщение переименовывается (на нем отображается операция класса).
- Настройте описания классов на конкретный язык программирования:
 - откройте диалоговое окно командой Tools: Options;
 - перейдите на вкладку Notation (рис. 18);
 - выберите название языка Ada 95 из раскрывающегося списка Default Language.

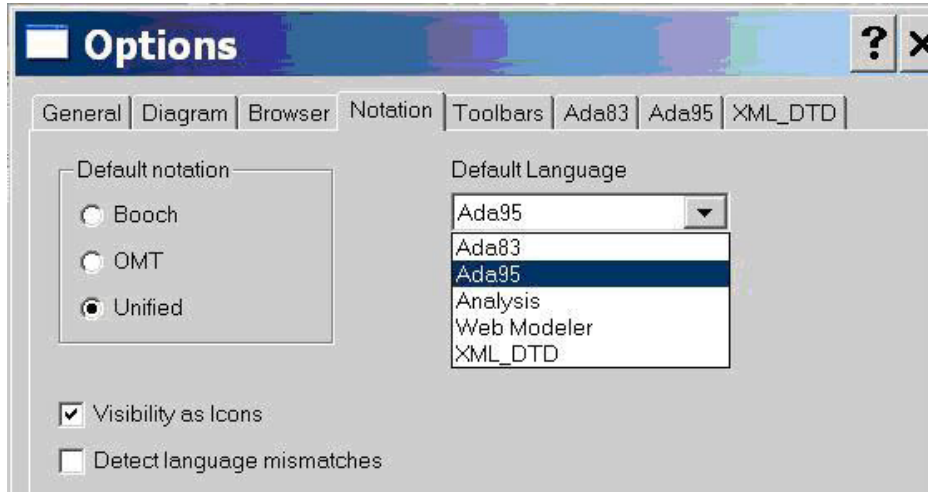


Рис. 18. Выбор языка программирования

- Определите типы данных, типы возвращаемых значений:
 - выполните двукратный щелчок по значку класса CourseOffering в окне браузера или диаграмме классов. В результате станет видимым окно спецификации класса;
 - выберите страницу Attributes (свойства) (рис. 19);
 - щелкните по полю Type. В результате станет видимым раскрывающийся список;
 - введите требуемый тип данных (Integer);
 - закройте окно спецификации, нажав кнопку ОК.

ПРИМЕЧАНИЕ:

Типы данных для свойств можно устанавливать, используя спецификацию Attribute или вводя их в строчке отображения свойства на диаграмме классов (формат attribute: data type).

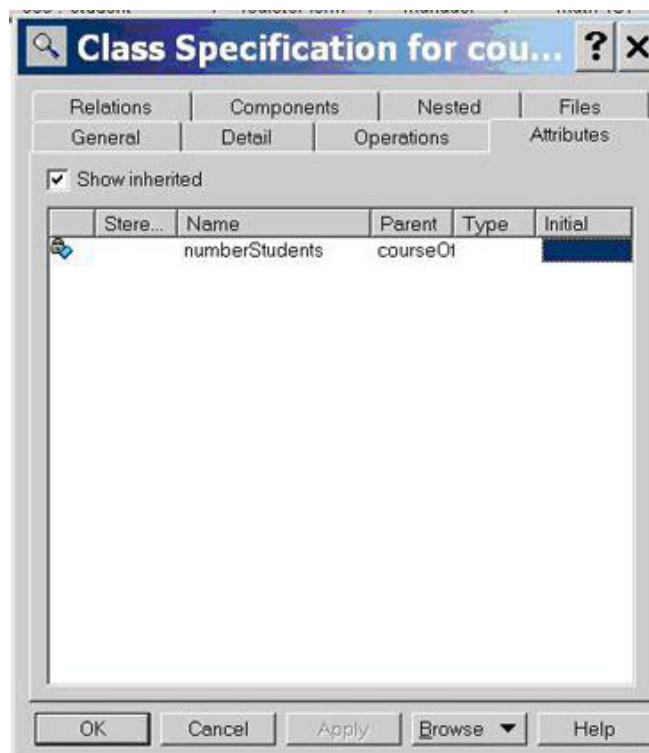


Рис. 19. Определение атрибутов

- Задайте тип возвращаемого результата для операции offeringOpen (рис. 20):
 - выполните двукратный щелчок по значку класса CourseOffering в окне браузера или диаграмме классов. В результате станет видимым окно спецификации класса;
 - выберите страницу Operations;

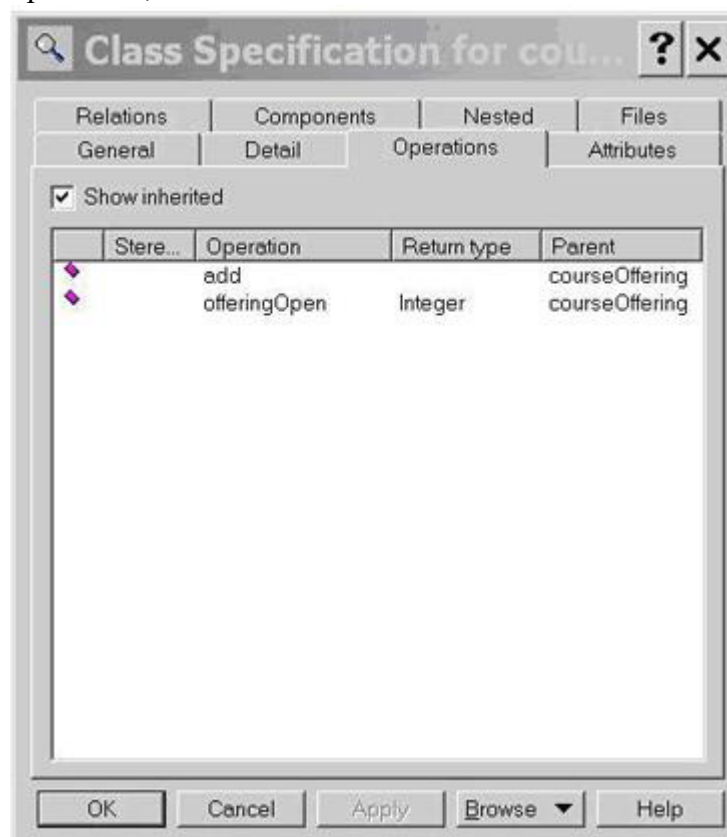


Рис. 20. Задание типа возвращаемого результата

- щелкните по полю Return type. В результате станет видимым раскрывающийся список;
- введите требуемый возвращаемый тип (Integer);
- закройте окно спецификации, нажав кнопку OK.

Примечания:

Аргументы операции устанавливаются с помощью диалогового окна *Operation Specification*. Для перехода к этому окну нужно на вкладке (странице) *Operations* щелкнуть правой кнопкой по имени операции и в появившемся контекстном меню выбрать команду *Specification*. Далее в появившемся диалоговом окне следует перейти на вкладку *Detail*. Аргументы операции и возвращаемый тип можно также установить, вводя их в строчке отображения операции на диаграмме классов (формат *operation (argument name: data type): return type*).

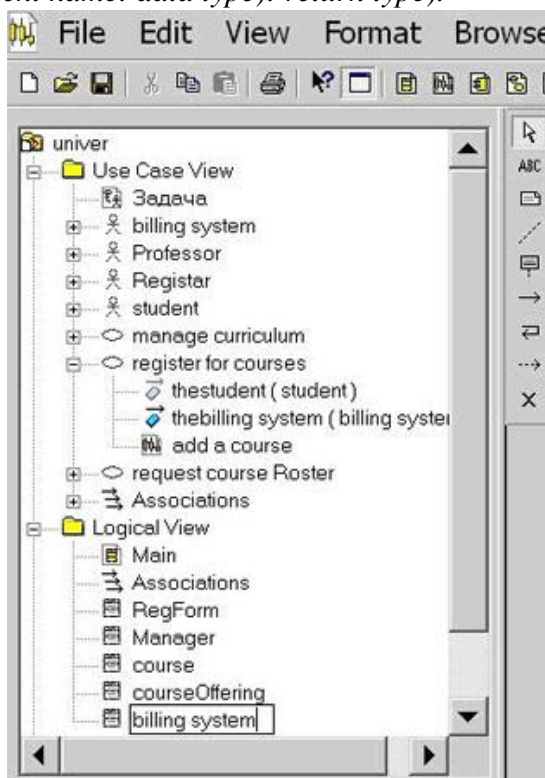


Рис. 21. Отображение классов в браузере

Для генерации кодов классов модели нужно создать компонентную диаграмму. Для определения компонентов исходного кода используют компонентное представление (Component View) (рис. 22). Среда Rational Rose автоматически создает главную компонентную диаграмму.

- Щелкните в окне браузера по значку + слева от пакета Component View.
- Откройте главную компонентную диаграмму, выполнив двойной щелчок по значку Main.

В общем случае каждому классу должны соответствовать два компонента — компонент спецификации и компонент реализации. Для каждого компонента будет создан свой файл. Например, в языке C++ классу соответствуют два файла-компонента: h-файл (файл спецификации) и cpp-файл (файл реализации).

- Создайте один компонент для представления файла спецификации по классу CourseOffering (расширение .ads и стереотип Package Specification) и один компонент для представления файла реализации по классу CourseOffering (расширение .adb и стереотип Package Body) (Рис. 22).

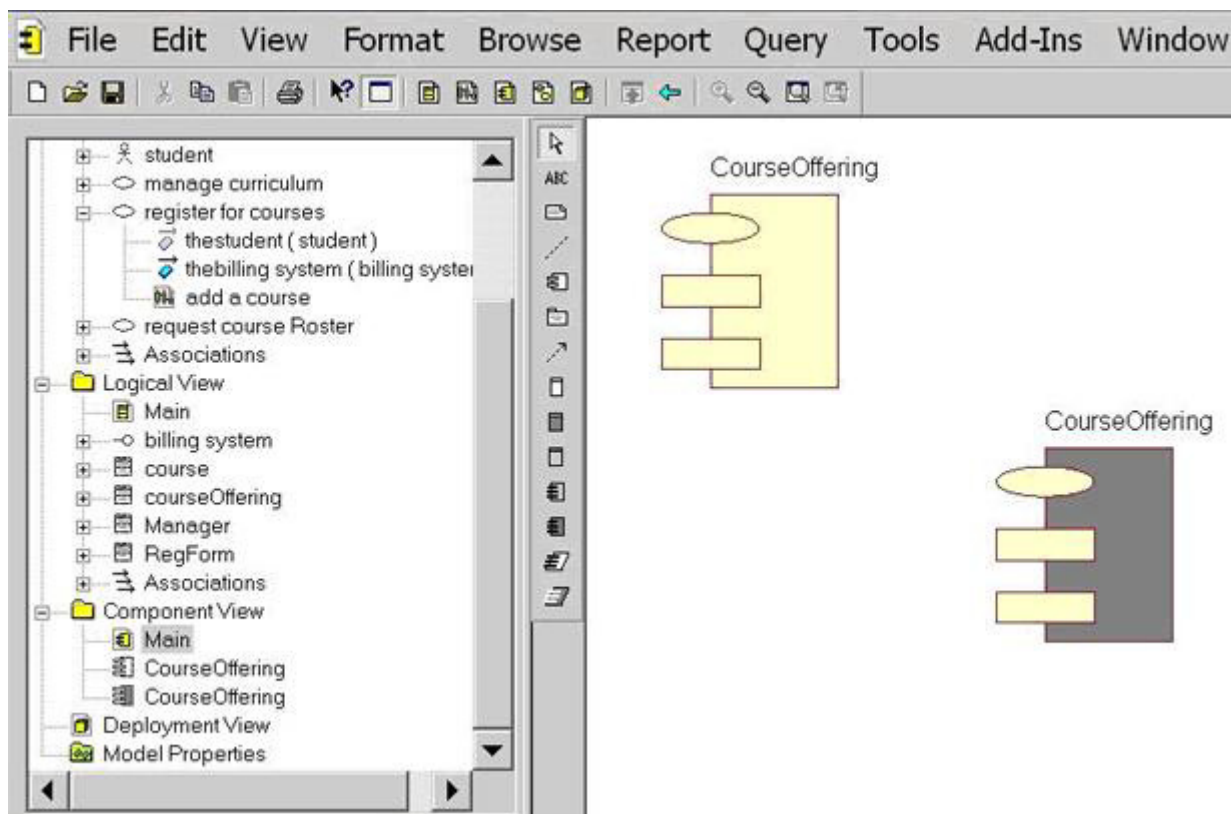


Рис. 22. Диаграмма компонентов

- щелкните на панели инструментов по значку спецификации пакета Package Specification;
- щелкните в нужном месте диаграммы для добавления компонента в диаграмму;
- введите его имя — CourseOffering, пока новый компонент остается выделенным;
- повторите предыдущие шаги с использованием значка тела пакета Package Body;
- щелкните на панели инструментов по значку отношения зависимости;
- щелкните по компоненту, представляющему .adb-файл (тело пакета), и перетащите стрелку на компонент, представляющий .ads-файл (спецификация пакета).
- Назначьте классы модели созданным компонентам (рис. 23).
- выполните двукратный щелчок по значку компонента CourseOffering, представляющего .ads-файл (спецификацию пакета), в окне браузера или компонентной диаграмме. В результате станет видимым окно спецификации компонента (рис. 23);
- выберите страницу (вкладку) Realizes. На ней представлен список классов модели;
- щелкните правой кнопкой по классу CourseOffering. В результате станет видимым контекстное меню;
- выберите команду Assign;

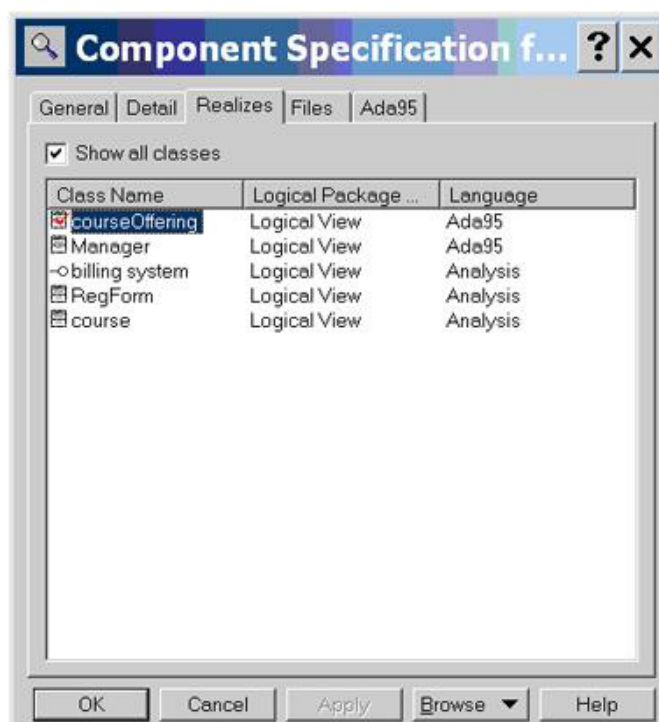


Рис. 23. Назначение классов компоненту

- закройте окно спецификации, нажав кнопку ОК;
- выполните аналогичные действия для тела пакета, представляющего.adb-файл.

Генерация кода:

- На компонентной диаграмме выделите оба компонента CourseOffering.
- Выберите команду Tools/Ada95/Code Generation из главного меню (рис. 24).
- Ознакомьтесь с появившимся окном итогов генерации кода Code Generation Status (рис. 25). Все ошибки заносятся в log -окно.
- Для завершения процесса генерации кода нажмите кнопку Close.

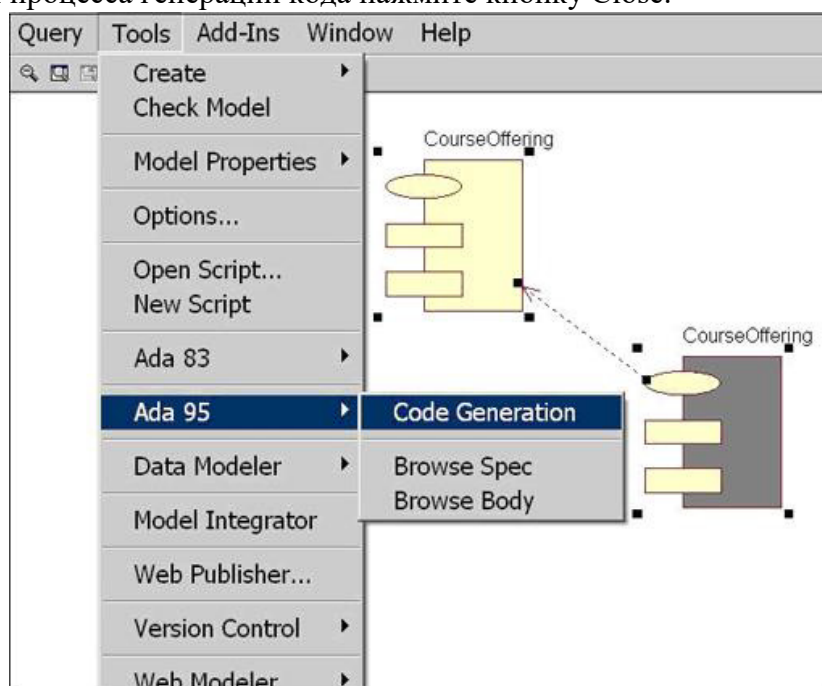


Рис. 24. Генерация кода

В процессе генерации Rational Rose отображает логическое описание класса в кар как программного кода — в коде появляются языковые описания имени класса, свойств класса и заголовки методов. Для описания тела каждого метода формируется программная заготовка.



Рис. 25. Окно итогов генерации кода

Появляются и программные связи классов.

Программист будет дополнять этот код, работая в конкретной среде программирования, имеющей мост связи с системой Rational Rose. После каждого существенного дополнения программист с помощью возвратного проектирования, основанного на использовании моста связи, будет модифицировать диаграммы классов, вводя в них изменения, соответствующие результатам программирования.

- Просмотрите код, сгенерированный средой Rational Rose.

Раздел 2. Оценочные материалы для оценки сформированности компетенций

Осваиваемая компетенция: ОК-1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

Индикаторы достижения компетенций

Знание: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте;

алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности

Умение: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;

составить план действия; определить необходимые ресурсы;

владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите контексты деятельности с соответствующими подходами к выбору способов решения задач.

Контексты	Подходы/способы решения
А. Разработка нового продукта в стартапе	1. Быстрый прототипирование и минимально жизнеспособный продукт (MVP) с частыми итерациями
В. Поддержка существующего инфраструктурного проекта	2. Формализация требований и строгий регламент процессов, документирование архитектуры
С. Внедрение строгих регуляторных требований	3. Применение проверенных шаблонов и стандартов, документирование решений и контроль версий
Д. Масштабирование продукта в многостраничной архитектуре	4. Поэтапное увеличение масштабируемости, выбор технологий с поддержкой горизонтального масштабирования

Ответ: A1B3C2D4

Соотнесите роли участников проекта с типичными функциями выбора средств разработки.

Контексты	Функции
А. Архитектор решений	1. Оценка совместимости инструментов с существующей инфраструктурой

B. Руководитель проекта	2. Определение требований к процессам сборки и развёртывания
C. DevOps-инженер	3. Выбор инструментов для тестирования и обеспечения качества
D. QA-инженер	4. Оценка соответствия требованиям заказчика и регуляторным требованиям

Ответ: A1B2C4D3

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность выбора инструментальных средств для нового проекта:

1. Определение требований к функциональности и нефункциональным характеристикам
2. Оценка совместимости с уже существующими стеками
3. Прототипирование и пилотное внедрение
4. Финальный выбор и внедрение

Ответ: 1234

Установите последовательность внедрения нового набора инструментальных средств:

1. Анализ рисков и план перехода
2. Обучение команды и настройка окружения
3. Пилотный запуск и сбор обратной связи
4. Масштабирование и продакшн-внедрение

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики полезны при выборе инструментальных средств разработки?

- A) Оценка стоимости владения и поддержки
- B) Игнорирование совместимости с существующими процессами
- C) Анализ сообщества и уровня поддержки инструментов
- D) Пренебрежение безопасностью конфигураций
- E) Планирование обучения и адекватная документация

Ответ: ACE

Какие методы минимизации рисков при выборе инструментов подходят для крупных проектов?

- A) Пилотирование на небольших командах
- B) Полный переход без этапного внедрения
- C) Трассировка требований к функциональности и совместимости
- D) Верификация через независимые аудиторы и ревью
- E) Игнорирование отзывов пользователей

Ответ: ACD

4. Прочитайте текст и дайте ответ:

Опишите подход к выбору инструментальных средств для нового проекта.

Ответ:

Функциональные:

Соответствие потребностям проекта: поддержка требуемых языков/платформ, наличие нужных функциональных модулей (сборка, тестирование, деплой, мониторинг).

Поддержка стандартов и совместимость с существующими процессами (Git, CI/CD, контейнеризация), наличие API/интерфейсов для интеграции.

Возможность масштабирования и адаптации под будущие требования.

Нефункциональные:

Цена владения и поддержка (лицензии, обновления, срок поддержки).

Надежность и репутация поставщика, наличие активного сообщества и документации.

Безопасность и соответствие регуляторным требованиям (если применимо).

Производительность и устойчивость к нагрузкам, совместимость с текущей инфраструктурой (серверные конфигурации, облако, контейнеры).

Простота внедрения и обучения команды, наличие обучающих материалов и примеров.

Оценка совместимости с существующими процессами и инфраструктурой

Провожу карту совместимости: какие части инструментов вписываются в наш стек (VCS, CI/CD, тестовые фреймворки, оркестрация, мониторинг).

Выполняю пилотный внедрение на небольшой подзадаче: оцениваю интеграцию через пайплайны, обмен артефактами, формат логирования, совместимость версий.

Анализирую риски миграции: потенциальные прерывания, необходимость миграции данных, требования к обучению сотрудников.

Проверяю инфраструктурные требования: поддержка наших ОС/платформ, требования к аппаратному обеспечению и сетевой безопасности.

Документация принятых решений и передача команде

Веду единую документацию по выбору инструментов: обоснование выбора, сравнительная таблица альтернатив, критерии оценки, затронутые риски, план внедрения.

Создаю артефакт «решение по инструментарию» с версионностью и датой утверждения.

Связываю решение с требованиями и архитектурой проекта (отображаю связь с модулями, пайплайнами и контрактами между командами).

Передаю команде через: обновления в Confluence/Notion, ревью в регистре изменений, уведомления в чатах и линк на интеграционные тесты/пилоты.

Обеспечиваю обучение и сопровождение: план обучения, расписание воркшопов, инструкции по настройке, чек-листы для стартового использования.

Опишите процесс внедрения и управления изменениями после выбора инструментов.

Ответ:

Планирование обучения и поддержка пользователей

Разрабатываю план обучения: вводные курсы по основам инструмента, продвинутые сессии по конкретным кейсам (CI/CD, тестирование, безопасность), обучающие материалы и документацию.

Назначаю наставников/модераторов внутри команд, создаю внутреннюю базу знаний с примерами типовых задач.

Организирую «плавающие окна обучения» в зависимости от графиков спринтов и релизов; обеспечиваю доступ к тренажёрам и песочницам.

Обеспечиваю поддержку через чат-каналы, SLA на вопросы и регулярные горячие линии для критических сценариев.

Мониторинг эффективности и сбор обратной связи

Определяю KPI: время внедрения, доля автоматизированных пайплайнов, частота успешных сборок, среднее время восстановления после изменений, количество инцидентов, связанные с инструментом.

Настраиваю дашборды в CI/CD/платформах мониторинга, регулярно собираю обратную связь от пользователей (опросы, ретроспективы, интервью).

Провожу периодические обзоры после релизов: что сработало, какие проблемы возникли, какие улучшения требуются.

Устойчивость к устареванию и обновлениям

Внедряю план версионирования инструментов, регистр изменений и совместимости между версиями.

Регулярно оцениваю риск устаревания и планирую обновления на окнах минимального влияния на бизнес (плановые релизы, тестовые прогонки).

Обеспечиваю совместимость артефактов и данных между версиями, сохраняю обратную совместимость там, где возможно.

Веду регулярные аудиты безопасности и зависимости, управляю обновлениями через централизованный реестр зависимостей и политики обновления.

5. Прочитайте текст и дополните его:

При выборе инструментальных средств важно учитывать _____, чтобы обеспечить долгосрочную устойчивость проекта.

Ответ: масштабируемость/поддерживаемость/совместимость

Процесс принятия решений по инструментам должен включать _____ анализ рисков и альтернатив.

Ответ: структурированный/формализованный/детальный

Осваиваемая компетенция: ОК-2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности

Индикаторы достижения компетенций

Знание: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации

Умение: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите типы источников с их основными особенностями.

Контексты	Особенности
А. Документация по инструментам (официальная документация)	1. Предоставляют официальный, точный и актуальный синтаксис и примеры использования инструментов

В. Сообщества разработчиков и форумы	2. Могут содержать неоднозначности и разнообразные практики; полезны для альтернативных подходов
С. Внутренние регистры изменений и артефакты проекта	3. Обеспечивают трассируемость решений и историю изменений в проекте
Д. Научные статьи и отраслевые обзоры	4. Проталкивают новые идеи, технологии и обобщённые принципы, требуют критического анализа

Ответ: A1B2C3D4

Соотнесите роли участников проекта с типами источников информации, которыми они обычно пользуются.

Контексты	Источники
А. Архитектор решений	1. Технические спецификации и контракты
В. Роль QA/тестировщик	2. Тестовые протоколы, отчёты о дефектах
С. Инженер DevOps	3. Операционные инструкции по сборке и развёртыванию
Д. Менеджер проекта	4. Обобщающие обзоры и бизнес-обоснования

Ответ: A1B2C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность поиска и анализа информации по инструментальным средствам проекта:

1. Определение потребностей проекта и критериев отбора источников
2. Поиск и сбор информации из титула (документации), форумов и регистров изменений
3. Верификация найденной информации через альтернативные источники и практические примеры
4. Документация выводов и формирование рекомендаций

Ответ: 1234

Установите последовательность подготовки и принятия решения по выбору инструментальных средств на основе полученной информации:

1. Оценка достоверности и применимости информации
2. Формирование альтернатив и сравнение по критериям
3. Принятие решения и документирование обоснований
4. Передача результатов команде и настройка коммуникации

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие источники информации стоит учитывать при формировании выбора инструментальных средств?

- А) Официальная документация по инструментам

- В) Рекомендации конкурентов без проверки контекста
 - С) Внутренние метрики и прошлый опыт команды
 - Д) Обзоры в блогах без ссылок на источники
 - Е) Тестовые примеры и демо-версии инструментов
- Ответ: ACE

Какие методы анализа информации полезны при работе с несколькими инструментами в проекте?.

- А) Контекстный сравнительный анализ (сильные/слабые стороны, риск-оценка)
 - В) Формализация требований без проверки инструментов
 - С) Верификация через пилотный прототип и минимальный экземпляр
 - Д) Игнорирование отзывов пользователей и команды
 - Е) Документация выводов с версиями и источниками
- Ответ: ACE

4. Прочитайте текст и дайте ответ:

Опишите процесс поиска и анализа информации для обоснования выбора инструментальных средств в рамках проекта.

Ответ:

Типы источников и оценка достоверности

Официальная документация инструментов: технические руководства, API-справочники, руководства по интеграции. Достоверность высокая, проверяемость источников по версии продукта, датам обновления и версии документации.

Внутренние артефакты проекта: требования, контракты, технические решения, результаты пилотов. Достоверность определяется степенью согласования в команде, наличием подписей и версий.

Рекомендованные внешние источники: обзоры в отраслевых изданиях, блоги экспертов, статьи со ссылками на примеры использования. Достоверность оцениваю по авторитетности источника, дате публикации, сопоставимости с требованиями проекта.

Сообщество и форумы: комментарии пользователей, ответы в репозиториях, вопросы/решения. Достоверность варьируется; использую как сигналы для дополнительных проверок, не как основную опору.

Практический пилот и демо-версии: собственные тесты и демонстрации на небольшой части проекта. Достоверность высокая, поскольку основана на реальном окружении проекта.

Структурирование найденной информации

Файлы и структура знаний:

Создаю единый репозиторий знаний (Wiki/Notion/Confluence) с разделами: требования, сравнение инструментов, архитектурные решения, риски, планы внедрения.

Использую единый шаблон для заметок: источник, версия/дата, сущность информации, связь с требованиями, риск/выгода.

Таблицы и диаграммы:

Таблицы сравнения инструментов: критерии (функционал, совместимость, лицензии, поддержка, безопасность, документы), оценки по шкале 1–5, краткие обоснования.

Диаграммы трассируемости: связь между требованиями, выбранными инструментами, артефактами проекта и тестами.

Мэппинг совместимости: текущий стек технологий, требования к интеграции, зоны риска миграции.

Процесс документирования:

Каждый этап анализа фиксируется в отчётах: цель, методика, результаты, альтернативы, риск-уровень, рекомендации.

Присваиваю версии документации и фиксирую даты утверждения: кто утвердил, какие изменения.

Передача результатов команде и коммуникация

Форматы передачи:

Краткие синхронизированные резюме для менеджмента (ключевые выводы, риски, рекомендации).

Подробные отчёты для специалистов (инструментологи, архитекторам) с обоснованием выбора и планом внедрения.

Каналы коммуникации:

Регулярные стендапы/синхронизации по проекту, где обсуждаются результаты анализа и принятое решение.

Обновления в регистре изменений, уведомления в чатах, публикации в общем репозитории знаний.

Включение представителей команд в процесс ревью: архитекторы, DevOps, QA, безопасность.

Последующая коммуникация:

План внедрения: сроки, этапы, ответственные, критерии приемки.

Обратная связь: сбор комментариев от команд, доработка критериев и повторная верификация выбранных средств.

Опишите подход к документированию и поддержке актуальности информации по инструментальным средствам в проекте.

Ответ:

Форматы документов и хранение

Форматы: структурированные руководства по инструментам, сравнительные таблицы, чек-листы внедрения, протоколы решений, отчёты пилотов, регистры изменений.

Хранение: единый репозиторий знаний (Confluence/Notion/Markdown в Git), структурированный по разделам проекта и инструментам; версии документов фиксируются в системе контроля версий и регистре изменений.

Метаданные: дата обновления, автор, версия инструмента, связанный артефакт (ТЗ, архитектурные решения, тесты).

Обновления и версионирование

Процедуры обновления: устанавливаю регулярные сессии обозрения актуальности информации (ежеквартально/при релизе), фиксирую изменения в регистре версий и привязываю к релизу проекта.

Версионирование: каждому документу присваивается номер версии, фиксируются предшествующие версии и изменения, ведётся история правок.

Контроль изменений: согласование изменений с ответственными командами (архитектура, разработка, QA, безопасность), регуляторные требования учитываются при обновлениях.

Доступ для всей команды

Доступность: обеспечиваю единый доступ к актуальным данным через общедоступный репозиторий знаний, регулирую доступ на уровне ролей (чтение/редактирование).

Прозрачность: публикую уведомления об обновлениях и пометки «актуально/устарело» в комментариях к документам.

Верификация актуальности: периодически провожу аудиты содержания, запрашиваю подтверждения у ответственных за компоненты инструментов.

5. Прочитайте текст и дополните его:

Для эффективной оценки информации нужно применять _____ методы анализа (пороговый анализ, сравнительный анализ, риск-оценку).

Ответ: структурированные/мультиметрические/мультитурные

При выборе инструментальных средств важно учитывать _____, чтобы обеспечить прозрачность и обоснованность решений.

Ответ: источники/достоверность/критерии отбора

Осваиваемая компетенция: ОК–3 Планировать и реализовывать собственное профессиональное и личностное развитие

Индикаторы достижения компетенций

Знание: содержание актуальной нормативно-правовой документации; современная научная и профессиональная терминология; возможные траектории профессионального развития и самообразования

Умение: определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите виды деятельности по развитию с соответствующими целями.

Контексты	Цели
А. Профессиональное обучение (курсы, сертификации)	1. Расширение профессионального набора навыков и сертификатов
В. Самостоятельная практика и изучение новых инструментов	2. Освоение новых технологий и инструментов на практике
С. Участие в сообществах и обмен опытом	3. Расширение сети контактов и обмен знаниями
Д. Планирование карьерного пути и целей	4. Четкое направление и этапы карьерного роста

Ответ: A1B2C3D4

Соотнесите участника команды с типичной ролью в личном развитии и обучении.

Контексты	Функции
А. Супервизор/наставник	1. Поддержка и руководство в выборе курсов и путей развития
В. Лидер команды	2. Поощрение и обеспечение возможностей для обучения в команде
С. Сотрудник, отвечающий за самообразование	3. Самостоятельная организация обучения и отслеживание прогресса
Д. HR/координатор обучения	4. Организация корпоративных обучающих программ и сертификаций

Ответ: A1B2C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность планирования личностного и профессионального развития:

1. Самооценка текущих компетенций и определение потребностей
2. Выбор инструментов и форматов обучения
3. Составление плана развития (цели, сроки, ресурсы)
4. Реализация плана и мониторинг прогресса

Ответ: 1234

Установите последовательность реализации индивидуального плана развития в контексте проекта:

1. Установка целей в рамках проекта и личной карьеры
2. Поиск курсов/курируемых материалов и назначение времени на обучение
3. Применение новых знаний на практике в проекте
4. Оценка результатов и корректировка плана
5. Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики эффективны для планирования и реализации профессионального развития?

- A) Регулярная саморефлексия и обновление плана развития
- B) Игнорирование обратной связи от коллег
- C) Участие в внешних курсах и сертификациях по мере возможности
- D) Игнорирование времени на обучение в рабочем расписании
- E) Документирование достигнутых результатов и новых навыков

Ответ: ACE

Какие подходы к развитию личной эффективности полезны в контексте работы с инструментальными средствами разработки?

- A) Техники тайм-менеджмента и приоритизации задач
- B) Регулярные ретроспективы по собственному обучению
- C) Одновременная попытка освоить все новые инструменты сразу
- D) Постановка измеримых целей и отслеживание прогресса
- E) Игнорирование слабых зон и фокус на сильных сторонах

Ответ: ABD

4. Прочитайте текст и дайте ответ:

Опишите подход к формированию личного плана профессионального развития в области инструментальных средств разработки ПО.

Ответ:

Оценка компетенций и выявление пробелов:

проводжу самооценку по ключевым навыкам (JVM/CLI/CI/CD/настройка окружения, тестирование, безопасность) и сравниваю с требованиями текущих проектов;

запрашиваю обратную связь у коллег и руководителей, анализирую прошлые задачи и места, где возникают задержки или повторяющиеся проблемы;
формирую список пробелов с приоритетами и ориентиром на ближайший релиз.
Выбор направлений обучения:
формирую набор курсов/книг/практических задач по каждому пробелу, учитывая релевантность к стеку и сроки проекта;
чередую теорию и практику (чтение материалов + мини-проекты/задания в рамках реального проекта);
устанавливаю конкретные цели и критерии приемки обучения (например, освоение инструмента в течение 4 недель, демонстрация решения в ревью).
Интеграция в рабочий распорядок и оценка эффективности:
включаю обучение в спринты: выделяю фиксированные окна для обучения и экспериментов;
после каждого этапа обучения выполняю небольшой спринтовый эксперимент и оцениваю влияние на производительность и качество;
веду краткий отчет об эффекте (скорость выполнения задач, качество кода, количество ошибок) и корректирую план при необходимости.

Опишите стратегию взаимодействия с руководством и коллегами для поддержки вашего профессионального роста.

Ответ:
Обоснование обучения:
подчеркиваю влияние на бизнес-цели: сокращение времени вывода функционала, повышение качества за счет современных инструментов, уменьшение рисков;
привожу данные по текущим проблемам и ожидаемым выгодам (метрики производительности, исправления ошибок, стабильность пайплайнов);
предлагаю конкретный план занятий и ориентировочные бюджеты/время.
Планирование времени и ресурсов:
договариваюсь с руководителем о составе времени в спринтах и доступе к обучающим материалам/средствам;
фиксирую согласование бюджета, лицензий, времени на обучение и тестовые среды;
обеспечиваю поддержку коллег через наставничество и командные обзоры.
Передача знаний и применение на практике:
делаю мини-брейнштормы/презентации для команды, создаю краткие руководства и чек-листы;
внедряю изученные практики в процессы (конфигурацию CI/CD, тестирование, рефакторинг);
отражаю эффект в реальных проектах: ускорение задач, улучшение качества кода, снижение числа дефектов, регулярные демонстрации результатов.

5. Прочитайте текст и дополните его:

Для эффективного планирования развития важно устанавливать _____, чтобы направлять обучение и оценивать прогресс.

Ответ: конкретные цели/SMART-цели/измеримые цели

Регулярная _____ обратная связь от руководителя и коллег поддерживает корректировку плана развития.

Ответ: постоянная/регулярная/конструктивная

Осваиваемая компетенция: ОК-4 Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами

Индикаторы достижения компетенций

Знание: психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности

Умение: организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите виды деятельности по развитию с соответствующими целями.

Контексты	Цели
А. Профессиональное обучение (курсы, сертификации)	1. Расширение профессионального набора навыков и сертификатов
В. Самостоятельная практика и изучение новых инструментов	2. Освоение новых технологий и инструментов на практике
С. Участие в сообществах и обмен опытом	3. Расширение сети контактов и обмен знаниями
Д. Планирование карьерного пути и целей	4. Четкое направление и этапы карьерного роста

Ответ: A1B2C3D4

Соотнесите участника команды с типичной ролью в личном развитии и обучении.

Контексты	Функции
1. Супервизор/наставник	1. Поддержка и руководство в выборе курсов и путей развития
2. Лидер команды	2. Поощрение и обеспечение возможностей для обучения в команде
3. Сотрудник, отвечающий за самообучение	3. Самостоятельная организация обучения и отслеживание прогресса
4. HR/координатор обучения	4. Организация корпоративных обучающих программ и сертификаций

Ответ: A1B2C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность планирования личностного и профессионального развития:

1. Самооценка текущих компетенций и определение потребностей
2. Выбор инструментов и форматов обучения
3. Составление плана развития (цели, сроки, ресурсы)
4. Реализация плана и мониторинг прогресса

Ответ: 1234

Установите последовательность реализации индивидуального плана развития в контексте проекта:

1. Установка целей в рамках проекта и личной карьеры
2. Поиск курсов/курируемых материалов и назначение времени на обучение
3. Применение новых знаний на практике в проекте
4. Оценка результатов и корректировка плана

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики улучшают взаимодействие в команде и с заказчиками?

- A) Открытая и уважительная коммуникация
- B) Игнорирование сроков и рисков
- C) Прозрачность статусов и принятия решений
- D) Регулярное получение обратной связи и её учет
- E) Молчаливое согласование требований без обсуждений

Ответ: ACD

Какие техники повышения эффективности взаимодействия с заказчиками и стейкхолдерами применимы в рамках ИСДПО?

- A) Регулярные демонстрации прототипов и результатов
- B) Избежание документирования решений
- C) Установление четких критериев приемки
- D) Игнорирование рисков и ограничений проекта
- E) Включение заказчика в процессы уточнения требований

Ответ: ACE

4. Прочитайте текст и дайте ответ:

Опишите подход к организации эффективного взаимодействия в команде и с заказчиками в рамках проекта по ИСДПО.

Ответ:

Правила коммуникации и роли

Определяю понятную структуру коммуникации: кто отвечает за какие каналы (Slack/Teams для оперативной связи, почта для формальных уведомлений, регистр изменений для документации).

Распределяю роли в команде: product owner/менеджер продукта, архитектор/технический лидер, DevOps, разработчики, тестировщики, аналитик требований. Для каждой роли фиксирую ответственность за взаимодействие с заказчиком, обновления статусов и принятие решений.

Устанавливаю правила взаимодействия: регулярные синхронизации (ежедневные стендапы), регламент ответов на запросы (SLA на ответы в течение X часов), требования к оформлению артефактов (чёткие формулировки, версии, ссылки на артефакты).

Прозрачность ролей и процессов документируется в одном источнике (регистра изменений/вики), доступном для всей команды и заказчика.

Сбор и использование обратной связи

Встраиваю механизм регулярной обратной связи: короткие опросы после спринтов, открытые сессии «-lessons learned», каналы для анонимных замечаний, а также формальные review-окна.

Обработку обратной связи делю на шаги: сбор данных, классификация по темам (требования, качество, процесс, коммуникация), приоритизация и формирование плана действий.

Обеспечиваю видимость статуса исправлений и улучшений: обновление регистров изменений, демонстрации на стендапах или демонстрациях, связь изменений с конкретными фичами/задачами.

Прозрачность статусов и решений

Веду единую систему трекинга задач и артефактов (jira/похожий инструмент) с понятной моделью статусов и историей изменений.

Регулярно показываю текущий статус проектов заказчику и внутри команды через дашборды и короткие презентации.

Документирую принятые решения (обоснование, альтернативы, риск-оценка) и связываю их с требованиями и тестами. Обновляю соответствующие документационные артефакты и версии в репозитории.

Опишите стратегию урегулирования конфликтов и управления ожиданиями заказчика в контексте разработки ПО.

Ответ:

Выявление и вынос на обсуждение конфликтных ситуаций

Активно слушаю стороны, фиксирую факты и факторы, вызывающие конфликт (политика регуляторов, сроки, качество, технические ограничения).

Систематизирую конфликтные вопросы в журнале конфликтов: участники, влияние на сроки, риски, альтернативы решения.

Организую открытые обсуждения с участием заинтересованных сторон (когда возможно — совместно с заказчиком) с целью достижения общего понимания проблемы.

Процедуры решения конфликтов и документирования решений

Привожу креативные и формализованные варианты решения проблемы, оцениваю риски и влияние на проект.

Выбор решения фиксирую в реестре решений с обоснованием, альтернативами и ответственными за внедрение.

Внедряю план действий, сроки и критерии приемки. Провожу последующую проверку на устойчивость решения через тесты/демонстрации.

Документация включает: контекст конфликта, принятые решения, обсуждаемые альтернативы, риски, план коммуникации и изменения в требованиях.

Информирование заказчика о рисках, изменениях и компромиссах

Регулярно информирую о восприятии рисков, возможных изменениях и компромиссах через запланированные встречи, письма-обновления и регистр изменений.

Предоставляю визуальные материалы (дашборды, диаграммы зависимостей, предварительные сценарии внедрения) для ясности.

Устанавливаю ожидания по времени на обсуждения и согласование изменений, фиксирую решения и сроки в официальных документах.

Как организовать эффективный процесс код-ревью в распределённой команде разработчиков, учитывая разные часовые пояса участников?

Ответ: Для эффективного код-ревью в распределённой команде следует:

1. Установить четкие стандарты и чек-листы проверки
2. Использовать асинхронное ревью через платформы типа GitHub/GitLab
3. Назначать временные окна для активного обсуждения
4. Внедрить систему ротации ревьюверов
5. Записывать видеоразборы сложных моментов
6. Использовать автоматизированные инструменты проверки стиля кода
7. Проводить еженедельные синхронные обсуждения ключевых изменений

Какие стратегии помогут плавно внедрить новые инструменты разработки в уже сложившуюся команду?

Ответ: Эффективные стратегии внедрения:

1. Постепенное введение с обучением на реальных задачах
2. Назначение "чемпионов" инструмента в команде
3. Создание песочницы для экспериментов
4. Интеграция с существующим workflow
5. Сбор обратной связи и адаптация под нужды команды
6. Документирование лучших практик использования
7. Регулярные демонстрации преимуществ нового инструмента

5. Прочитайте текст и дополните его:

Для эффективного взаимодействия в команде важно поддерживать _____, чтобы снизить недопонимания и повысить продуктивность.

Ответ: открытость/прозрачность/доверие

Чтобы обеспечить эффективную коммуникацию с заказчиком, необходимо устанавливать _____ каналы и форматы отчетности.

Ответ: регулярные/предсказуемые/четко структурированные

Осваиваемая компетенция: ОК-5 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

Индикаторы достижения компетенций

Знание: особенности социального и культурного контекста; правила оформления документов и построения устных сообщений

Умение: грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите типы коммуникаций с целями в рамках проекта.

Контексты	Цели
А. Формальная переписка с госорганами	1. Обеспечение ясности требований и согласование изменений
В. Внутренняя коммуникация в команде	2. Поддержание прозрачности статусов и документирования решений
С. Коммуникация с заказчиками	3. Соблюдение формальностей,

	стандартов и регуляторных требований
D. Общение с регуляторами и аудиторами	4. Тестирование восприятия информации и сбор обратной связи

Ответ: A3B2C1D4

Соотнесите каналы коммуникации с их характеристиками.

Контексты	Характеристики
A. Официальные письма	1. Удобны для оперативной координации и быстрых вопросов
B. Внутренний чат/мессенджер	2. Подходят для формального формулирования требований и регуляторной документации
C. Видеоконференции	3. Позволяют лицом к лицу обсудить сложные вопросы и повысить доверие
D. Публичные отчеты/презентации	4. Предназначены для широкой аудитории и демонстрации результатов

Ответ: A2B1C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность подготовки и проведения устной коммуникации с заказчиком по техническим решениям:

1. Подготовка материалов (пояснения, резюме, диаграммы)
2. Выбор формата встречи и расписания
3. Проведение встречи и сбор вопросов
4. Фиксация итогов и передача материалов

Ответ: 1234

Установите последовательность подготовки и подачи письменного отчета заказчику:

1. Сбор данных и формулировка выводов
2. Оформление документа в официальном стиле
3. Ревизия терминами и соответствие регуляторным требованиям
4. Рассылка и публикация отчета

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие принципы коммуникации важны для эффективной устной и письменной коммуникации в государственном стиле?

- A) Точность формулировок и отсутствие двусмысленности
- B) Эмоциональная окрашенность и неформальный стиль
- C) Соблюдение регуляторных требований и стандартов оформления
- D) Игнорирование вопросов заказчика
- E) Ясная структура документа и логическая последовательность

Ответ: ACE

Какие методы проверки понятности и полноты информации в рамках госязыка можно применить?

- А) Предварительные чтения коллег-редакторов
- В) Чек-листы терминологии и регламентов
- С) Публикация без обратной связи
- Д) Тестирование на реальной аудитории (регуляторы/заказчики)
- Е) Использование двусмысленных формулировок для гибкости

Ответ: ABD

4. Прочитайте текст и дайте ответ:

Опишите подход к подготовке и проведению устной коммуникации с заказчиком на государственном языке РФ в условиях ограничений культурного контекста.

Ответ:

Адаптация стиля и форматов под аудиторию

Для госорганов: используйте формальный стиль, осторожную формулировку, точные термины, ссылки на регуляторные требования и стандарты оформления. Предпочтение имеют структурированные презентации (цель, сроки, риски, решение) и четко зафиксированные договоренности.

Для клиентов/заказчиков: сохраняйте деловой, но понятный стиль, делайте акцент на бизнес-ценности, риски и KPI. Включайте краткие резюме по итогам встречи, конкретные действия и сроки.

Для коллег внутри команды: можно использовать более гибкий стиль, акценты на технических деталях, схемах архитектуры, допущениях и возможностях внедрения; применяйте гибкую структуру презентаций (пороговые решения, альтернативы, риск-оценка).

Общий подход: заранее обсуждайте формат встречи, готовьте материалы в виде коротких резюме и детальных приложений, используйте визуальные средства (схемы, диаграммы) для повышения понимания.

Регулирование языка и оформление документов

Соблюдайте официальный стиль при документах для госорганов и регуляторов: формулировки, терминология, структура документации, ссылки на нормативные документы.

Обеспечьте единый лексикон и глоссарий для всей команды, чтобы снизить риск двусмысленности.

Используйте ясные заголовки, нумерацию разделов, таблицы и диаграммы, чтобы улучшить восприятие и прослеживаемость.

Проверяйте язык на соответствие требованиям: корректность грамматики, отсутствие двусмысленных формулировок, единообразие терминов.

Сбор и использование обратной связи после встречи

Сбор: применяйте короткие опросы или чек-листы после встречи, записывайте вопросы и замечания заказчика, фиксируйте акты встречи.

Анализ: категоризируйте обратную связь по темам (требования, сроки, качество коммуникации, ясность материалов), оценивайте важность и влияние на план проекта.

Действие: формируйте план действий по каждому пункту обратной связи с ответственными лицами и сроками выполнения; возвращайтесь к заказчику с подтверждением принятых мер и ожидаемых результатов.

Контроль: обновляйте документы и регистры решений, держите регуляторов/заказчика в курсе прогресса через регламентированные форматы отчетности.

Опишите подход к подготовке письменной документации на государственном языке РФ, учитывая культурные различия и нормы делового общения.

Ответ:

Элементы документа и их содержание

Титульный лист: наименование документа, дата, версия, инициатор.

Аннотация/резюме: кратко о цели, основных выводах и рекомендациях.

Введение: контекст, объекты и предмет документа.

Основная часть: требования/данные, анализ, обоснования решений, альтернативы.

Заключение: выводы и рекомендации, план действий.

Приложения: диаграммы, таблицы, ссылки на регуляторные документы.

Ссылочная часть и перечень используемой терминологии (гlossарий).

Регистры изменений: версия, дата обновления, автор, суть изменений.

Подписи и утверждения ответственных лиц.

Проверка языка и соответствие требованиям регуляторов

Применяйте официальный стиль, избегайте двусмысленных формулировок, используйте точную терминологию и нормативные ссылки.

Выполняйте вычитку и редактуру: орфография, пунктуация, стиль, соответствие регламентам оформления документов.

Привлекайте носителей языка/редакторов для проверки терминологии и соответствия отраслевым требованиям.

Проводите соответствующую экспертизу: сверяйте содержание с регуляторными требованиями и внутренними процедурами компании.

Передача материалов и контроль доступа

Используйте единое место хранения документов (регистры изменений, база знаний или репозиторий документов) с управлением доступом по ролям.

Обеспечьте версии и контроль изменений: кто внёс изменения, когда, какие именно изменения.

Организуйте контроль доступа: назначьте роли (чтение/редактирование), регламентируйте передачу материалов заказчику и внутри команды.

Обеспечьте возможность обратной связи и запросов на доступ к архивным версиям документов.

Какие основные критерии вы бы использовали при выборе системы контроля версий для крупного IT-проекта? Приведите обоснование вашего выбора.

Ответ: Для крупного проекта следует выбирать систему контроля версий, учитывая:

1. Поддержку распределенной работы (например, Git)
2. Возможности ветвления и слияния кода
3. Интеграцию с CI/CD системами
4. Масштабируемость и производительность
5. Наличие инструментов для code review

Git является оптимальным выбором, так как удовлетворяет всем этим требованиям и является отраслевым стандартом.

Опишите преимущества использования интегрированных сред разработки (IDE) по сравнению с простыми текстовыми редакторами.

Ответ: Основные преимущества IDE:

1. Встроенные инструменты отладки
2. Автодополнение кода и рефакторинг
3. Интеграция с системами сборки
4. Поддержка проектной структуры
5. Встроенные средства тестирования

Эти возможности значительно ускоряют разработку и уменьшают количество ошибок.

5. Прочитайте текст и дополните его:

Для эффективной коммуникации на государственном языке важно поддерживать _____, чтобы обеспечить ясность и уважение к культурному контексту.

Ответ: официальный стиль/официально-деловой/формальный стиль

При взаимодействии с регуляторами и госорганами необходимо соблюдать _____ форматы и стандарты оформления документов.

Ответ: унифицированные/регламентные/официальные

Осваиваемая компетенция: ОК-6 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения

Индикаторы достижения компетенций

Знание: сущность гражданско-патриотической позиции, общечеловеческих ценностей; значимость профессиональной деятельности по специальности

Умение: описывать значимость своей специальности

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите аспекты гражданской ответственности с соответствующими действиями в рамках проекта.

Контексты	Действия
А. Соблюдение закона и регуляторных требований	1. Подпись контрактов без скрытых условий и злоупотреблений
В. Прозрачность и открытость в отношениях с заказчиком	2. Публикация отчётности по расходам и времени в общедоступной форме
С. Этическое поведение и противодействие коррупции	3. Отказ от непроверенных бонусов и взяток, сообщение о конфликтах интересов
Д. Ответственное использование ресурсов и времени	4. Соответствие требованиям закона, прозрачное ведение документации

Ответ: A4B2C3D1

Соотнесите виды ценностей с примерами поведения.

Контексты	Примеры
А. Социальная ответственность	1. Прозрачная обработка персональных данных клиентов
В. Честность и справедливость	2. Соблюдение регламентов и стандартов разработки
С. Профессиональная компетентность	3. Учет интересов общества и устойчивое развитие проекта
Д. Национальная и безопасность данных (защита информации)	4. Честное распределение задач и вознаграждений

Ответ: A3B4C2D1

Соотнесите принципы выбора инструментов с их характеристиками:

Принцип	Характеристика
А) Патриотический подход	1) Приоритет российских аналогов
В) Прозрачность	2) Открытые конкурсные процедуры
С) Техническая замкнутость	3) Использование только зарубежных решений
Д) Антикоррупционность	4) Четкие критерии оценки предложений

Ответ: A1B2D4

Соотнесите этапы внедрения с их социальной значимостью:

Этап	Значение
А) Обучение сотрудников	1) Повышение цифровой грамотности
В) Открытая документация	2) Доступность для всех пользователей
С) Скрытое тестирование	3) Недоверие к системе
Д) Мониторинг результатов	4) Улучшение качества услуг

Ответ: A1B2D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность внедрения антикоррупционных практик в проекте ИСДПО:

1. Обучение команды антикоррупционной политике
2. Разработка регламентов по закупкам и принятию решений
3. Внедрение механизмов контроля и аудита
4. Мониторинг соблюдения и коррекция политики

Ответ: 1234

Установите последовательность формирования общественно ответственной культуры в команде:

1. Определение ценностей и принципов поведения
2. Введение практик прозрачности и обмена знаниями
3. Регулярные обсуждения этических кейсов

4. Оценка вклада и корректировка политики

Ответ: 1234

Установите правильную последовательность внедрения отечественного ПО в госучреждении:

1. Проведение сравнительного анализа с зарубежными аналогами
2. Организация обучения сотрудников
3. Получение сертификата ФСТЭК
4. Пилотное тестирование в одном департаменте
5. Полномасштабное развертывание

Ответ: 13245

Расположите этапы выбора инструментов разработки с учетом антикоррупционных мер:

1. Публикация открытого технического задания
2. Проведение независимой экспертизы предложений
3. Анализ российских аналогов
4. Подписание контракта с прозрачными KPI
5. Организация тендера

Ответ: 31524

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики способствуют демонстрации гражданско-патриотической позиции в рамках проекта?

- A) Соблюдение законов и регуляторных требований
- B) Игнорирование регламентов ради ускорения процесса
- C) Прозрачность в расходах и времени
- D) Уклонение от ответственности за ошибки
- E) Этичное поведение и уважение к коллегам

Ответ: ACE

Какие подходы помогают предотвращать конфликт интересов в процессе разработки ПО?

- A) Объявление и фиксация конфликтов интересов
- B) Информирование руководства и участников проекта
- C) Автоматическое игнорирование любых замечаний
- D) Разделение ролей и ротация обязанностей
- E) Прозрачное документирование принятых решений

Ответ: ABDE

Какие характеристики инструментов разработки соответствуют антикоррупционным стандартам?

- A. Открытая лицензионная политика
- B. Прозрачные механизмы ценообразования
- C. Наличие "серых" схем приобретения
- D. Возможность независимого аудита кода
- E. Закрытые условия сотрудничества с поставщиками

Ответы: ABD

Какие действия при выборе средств разработки демонстрируют патриотическую позицию?

- A. Предпочтение российских аналогов
- B. Использование исключительно зарубежных решений
- C. Участие в тестировании отечественных IDE
- D. Поддержка локализации на русский язык
- E. Отказ от изучения российских стандартов

Ответ: ACD

4. Прочитайте текст и дайте ответ:

Как применяю принципы антикоррупционного поведения в проектах ИСДПО

Ответ:

Меры против конфликтов интересов и соблюдения регламентов:

открыто объявляю возможные конфликты интересов, веду реестр и получаю формальное согласование.

соблюдаю регламенты закупок, требований к документации и утверждений, применяю принципы разделения функций.

исключаю личную выгоду при выборе инструментов и подрядчиков, привлекаю независимых рецензентов.

Прозрачность закупок и распределения ресурсов:

фиксирую критерии отбора, веду прозрачный тендер/обзор вариантов, публикую обоснования решений.

документирую распределение задач и времени, публикую отчеты о расходах и распределении ресурсов.

Реакция на сомнения в части этики и законности:

активно выслушиваю, собираю факты, приглашаю независимых сотрудников к разбору.

предоставляю чёткие ответы, демонстрирую соблюдение регламентов и регистрирую принятые решения.

Как формирую культуру честности и ответственности в команде

Ответ:

Способы поощрения открытой коммуникации и обмена информацией:

регулярно проводю открытые обсуждения, создаю безопасное пространство для вопросов и критики.

внедряю краткие стендапы, ретроспективы и публичные апдейты по прогрессу и проблемам.

Обучение этике и правовым требованиям:

организую краткие тренинги по антикоррупционной политике, регуляторным требованиям и кодексу поведения.

предоставляю чек-листы и руководства по этике, назначаю наставников для поддержки сотрудников.

Фиксация и обсуждение этических кейсов:

веду журнал кейсов с описанием ситуации, принятых решений и уроков.

регулярно обсуждаю кейсы на ретроспективах и внедряю выводы в процессы и регламенты.

Какие критерии выбора инструментов разработки должны учитываться для соблюдения антикоррупционных принципов в государственных ИТ-проектах?

Ответ:

1. Наличие сертификации ФСТЭК/Минцифры для российского ПО
2. Прозрачность лицензионной политики и ценообразования
3. Возможность независимой верификации исходного кода
4. Соответствие требованиям импортозамещения
5. Наличие документации на русском языке

Как разработчик может проявить патриотическую позицию при выборе инструментов для образовательного проекта?

Ответ:

1. Предпочтение отечественных сред разработки (PascalABC.NET, КуМир)
2. Использование открытых стандартов вместо проприетарных форматов
3. Участие в локализации интерфейсов инструментов
4. Интеграция с российскими образовательными платформами
5. Проведение мастер-классов по работе с отечественным ПО

5. Прочитайте текст и дополните его:

Для эффективного противодействия коррупции в проекте важно соблюдать _____, чтобы обеспечить прозрачность и доверие.

Ответ: регламент/принципы/политика

Внедрение _____ процедур по принятию решений снижает риск конфликтов интересов.

Ответ: прозрачных/формализованных/механизмов

Осваиваемая компетенция: ОК-7 Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях

Индикаторы достижения компетенций

Знание: правила экологической безопасности при ведении профессиональной деятельности; основные ресурсы, задействованные в профессиональной деятельности; пути обеспечения ресурсосбережения

Умение: соблюдать нормы экологической безопасности; определять направления ресурсосбережения в рамках профессиональной деятельности по специальности

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите практики разработки с их экологическими эффектами.

Контексты	Эффекты
А. Энергосберегающее программирование (оптимизация Alg/алгоритмов)	1. Снижение энергопотребления серверов и устройств
В. Контейнеризация и эффективное использование ресурсов	2. Оптимизация использования вычислительных ресурсов
С. Внедрение CI/CD с минимальным потреблением энергии	3. Снижение первичных выбросов за счет меньшей потребности в мощности

D. Тестирование в облаке/локальных средах с учетом энергоэффективности	4. Уменьшение времени тестирования и потребности в вычислительных ресурсах
--	--

Ответ: A2B1C3D4

Соотнесите роли участников проекта с ответственностями в области экологичности и ЧС.

Контексты	Ответственности
A. Архитектор решений	1. Оценка энергоэффективности пайплайна и инфраструктуры
B. Инженер DevOps	2. Реализация планов по устойчивому тестированию и контролю рисков
C. Тестировщик/QA	3. Внесение экологических требований в ТЗ и регламенты
D. Менеджер проекта	4. Координация действий по безаварийной эксплуатации

Ответ: A3B1C2D4

Установите соответствие между функциями IDE и энергоэффективностью:

Функции IDE	Энергоэффективность
1. Автодополнение кода	A) Сокращение времени процессорных вычислений
2. Инкрементальная компиляция	B) Выявление энергозатратных участков кода
3. Встроенный профилировщик	C) Уменьшение времени написания кода
4. Подсветка синтаксиса	D) Снижение нагрузки на глаза разработчика
5. Интеграция с ChatGPT	

Ответ: 1C2A3B4D

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность подготовки к внедрению экологичных практик в проекте:

1. Оценка текущего энергопотребления и выбросов
2. Выбор инструментов и методик снижения энергопотребления
3. Внедрение изменений в пайплайн и инфраструктуру
4. Мониторинг эффектов и корректировка

Ответ: 1234

Установите последовательность реагирования на чрезвычайную ситуацию в проекте (например, аварийное выключение ресурсов):

1. Инициация оповещений и приведения плана в действие
2. Определение приоритетов восстановления и ролей
3. Выполнение рестарта/восстановления компонентов
4. Восстановление нормального функционирования и анализ причин

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики способствуют экологичной и безопасной разработке ПО?

- A) Использование энергосберегающих настроек CI/CD и виртуализации
- B) Развертывания без учета энергопотребления в продакшене
- C) Оптимизация вычислительных задач и сроков тестирования
- D) Игнорирование регламентов по хранению данных
- E) Планирование снижения углеродного следа инфраструктурных решений

Ответ: ACE

Какие меры помогают снижать риски в чрезвычайных ситуациях в ИСДПО проектах?

- A) Наличие заранее расписанных процедур инцидент-менеджмента
- B) Регулярные учения и тесты восстановления
- C) Игнорирование уведомлений о сбоях
- D) Документирование уроков и обновление регламентов
- E) Резервирование критических компонентов и данных

Ответ: ABDE

4. Прочитайте текст и дайте ответ:

Опишите подход к внедрению экологичных и безопасных практик в рамках проекта по ИСДПО

Ответ:

Меры по снижению энергопотребления и отходов:

применение энергосберегающих настроек в CI/CD, виртуализации и контейнеризации;

оптимизация вычислительных задач, отказ от ненужных процессов, плавное масштабирование;

минимизация хранения и переноса данных, использование ретенции логов;

использование экологичных облачных площадок и повторного использования инфраструктуры.

Оценка эффекта и метрики:

метрики энергопотребления (кВт-ч на сборку/прайм-процесс), время выполнения тестов, коэффициент загрузки серверов;

показатели облачных затрат и выбросов CO₂-эквивалента;

частота регрессионных ошибок после изменений и экономия ресурсов.

Согласование с заказчиком и командой:

фиксирую цели и ожидаемые бизнес-выгоды, предлагаю план внедрения с этапами и бюджетом;

регулярно обсуждаю прогресс на ретроспективах и в статус-совещаниях; оформляю решения в регистре изменений и в документах проекта.

Опишите стратегию подготовки и проведения учений по ликвидации ЧС и восстановлению после инцидентов в ИСДПО

Ответ:

Сценарии и тренировки:

разрабатываю типовые сценарии ЧС: отказ инфраструктуры, утечка данных, сбой пайплайна; проводлю тренировки с ролями и фиксированными временными рамками;

регулярные учения, включая эмуляцию уведомлений, запуск плана и тестирование коммуникаций.

Фиксация опыта и улучшения:

веду журнал инцидентов и уроков (учитываю причины, влияние, применённые решения);

обновляю регламенты, чек-листы и планы учений; внедряю корректировки в пайплайны и архитектуру.

Информирование команды и заказчика:

подводим итоги учений через краткие доклады и письма-обновления;

публикуем результаты, планы улучшений и ответственные лица; обеспечиваем доступ к отчетам и материалам учений.

Какие характеристики инструментов разработки следует учитывать для снижения их экологического воздействия?

Ответ:

1. Энергоэффективность работы IDE
2. Возможность работы на маломощных устройствах
3. Поддержка удаленной разработки
4. Интеграция с облачными сервисами
5. Наличие инструментов анализа производительности

Как организовать инструментальную среду разработки, чтобы минимизировать потребление ресурсов?

Ответ:

1. Использование легковесных редакторов кода
2. Настройка виртуальных окружений
3. Применение контейнеризации
4. Оптимизация настроек сборки
5. Использование shared dev-серверов

5. Прочитайте текст и дополните его:

Для устойчивого развития проекта важно учитывать _____, чтобы снизить воздействие на окружающую среду.

Ответ: энергосбережение/утилизация/ресурсосбережение

При планировании ЧС необходимо иметь _____ план и процедуры для быстрого восстановления.

Ответ: покрывающий/формализованный/детализированный

Осваиваемая компетенция: ОК-8 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

Индикаторы достижения компетенций

Знание: роль физической культуры в общекультурном, профессиональном и социальном развитии человека; основы здорового образа жизни; условия профессиональной деятельности и зоны риска физического здоровья для специальности; средства профилактики перенапряжения

Умение: использовать физкультурно-оздоровительную деятельность для укрепления здоровья, достижения жизненных и профессиональных целей; применять рациональные приемы двигательных функций в профессиональной деятельности; пользоваться средствами профилактики перенапряжения характерными для данной специальности

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите виды физической активности с их целями в контексте работы над ИСДПО.

Контексты	Цели
А. Разминка перед рабочими задачами	1. Повышение выносливости и общая физическая активность
В. Регулярные кардио-тренировки	2. Предотвращение болей в спине и улучшение осанки
С. Укрепляющие упражнения для мышц спины/пояса	3. Снижение напряжения и улучшение гибкости
Д. Растяжка и релаксация после работы	4. Поддержание мобильности и снятие мышечной усталости

Ответ: A3B1C2D4

Соотнесите роли участников команды с их участием в программе ОК-8.

Контексты	Участие
А. Руководитель проекта	1. Поощрение перерывов на разминку и участие в инструктажах по технике безопасности
В. Специалист по охране труда	2. Разработка политики здоровья и расписания коротких физкультурных пауз
С. Специалист по фитнесу/Wellness	3. Организация мероприятий и тестирования функциональной пригодности
Д. QA-инженер	4. Ведение тестирования и контроля качества

Ответ: A2B1C3D4

Соотнесите языки программирования с их основным применением.

Язы	Применение
1. Python	A. Data Science и ML
2. JavaScript	B. Веб-разработка (Frontend)
3. SQL	C. Работа с базами данных

к Язы	Применение
C++ 4.	D. Высокопроизводительные вычисления

Ответ: 1A2B3C4D

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность внедрения программы физической активности на рабочем месте:

1. Оценка потребностей сотрудников и текущей физической подготовки
2. Разработка набора упражнений и расписания пауз
3. Внедрение программы и обучение технике выполнения
4. Мониторинг эффективности и коррекции

Ответ: 1234

Этапы проведения периодического контроля здоровья сотрудников в контексте работы над проектами ИСДПО:

1. Организация периодических медицинских осмотров и опросов самочувствия
2. Анализ данных и определение зон риска
3. Введение корректирующих мер (паузы, упражнения, рекомендации)
4. Обсуждение результатов с командой и руководством

Ответ: 1234

Установите последовательность работы с Git:

1. Фиксация изменений (git commit)
2. Клонирование репозитория (git clone)
3. Отправка изменений на сервер (git push)
4. Добавление файлов в индекс (git add)
5. Создание новой ветки (git branch)

Ответ: 25413

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики помогают поддерживать здоровье и работоспособность при работе за компьютером?

- A) Регулярные перерывы на разминку
- B) Непрерывная работа без отдыха для скорости
- C) Правильная организация рабочего места (освещение, монитор на уровне глаз)
- D) Глоток кофе вместо физической активности
- E) Достижение баланса между физической активностью и сном

Ответ: ACE

Какие методы контроля за эффективностью физкультурной программы на рабочем месте целесообразно использовать?

- A) Самоотчеты сотрудников о самочувствии

- В) Измерение времени пауз и количества выполненных упражнений
 - С) Мониторинг продуктивности без учета физического состояния
 - Д) Регистрация жалоб и боль в спине
 - Е) Регулярные медицинские обследования по графику
- Ответ: ABDE

4. Прочитайте текст и дайте ответ:

Опишите подход к внедрению программы физической активности на рабочем месте в рамках проекта по ИСДПО

Ответ:

Виды упражнений и пауз:

краткие разминки в начале и между задачами, растяжка, микро-перерывы на глаза и спину; легкие силовые упражнения (перемена плеч, наклоны, приседания) без оборудования; прогулки на 5–10 минут каждые 2–3 часа.

Оценка эффекта и метрики:

метрики: частота участия, продолжительность активностей, субъективное самочувствие, уровень усталости, показатели продуктивности и ошибок, визуализация на дашборде.

Согласование с руководством и командой:

представлю план с целями, бюджетом и расписанием; проведу пилотный запуск на одной группе; регулярно обсуждать результаты на стендапах и ретроспективах.

Опишите стратегию поддержки здорового образа жизни в команде

Ответ:

Мотивация участия:

создание мотивирующих целей, небольшие стимулы, групповые занятия, корпоративные чаты с напоминаниями и история успеха.

Учет ограничений и противопоказаний:

сбор медицинских ограничений, индивидуальные варианты упражнений, опции замены активностей, возможность выбора уровней сложности.

Интеграция результатов в культуру проекта:

включение физической активности в политику команды, отражение практик в регламентах и onboarding, публикация кейсов и успешных практик на общем информационном ресурсе.

5. Прочитайте текст и дополните его:

Регулярные _____ паузы помогают снизить усталость глаз и мышечное напряжение.

Ответ: краткие/регулярные/тайм-ауты

Для поддержания физической формы на рабочем месте важны _____ привычки (распорядок, сон, питание, Bewegung).

Ответ: здоровые/постоянные/устойчивые

Осваиваемая компетенция: ОК-9 Использовать информационные технологии в профессиональной деятельности

Индикаторы достижения компетенций

Знание: современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности

Умение: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите типы информационных технологий с их применением в профессиональной деятельности.

Контексты	Примеры применения
A. Облачные вычисления	1. Хранение и обработка данных в удалённых дата-центрах
B. Большие данные (Big Data) и аналитика	2. Анализ больших массивов данных и извлечение инсайтов
C. Кибербезопасность	3. Защита информации, управление доступом, мониторинг угроз
D. Информационная визуализация	4. Визуализация сложной информации для принятия решений

Соотнесите роли в команде с соответствующими инструментами и подходами ИТ-обеспечения.

Контексты	Инструменты/подходы
A. ИТ-администратор	1. Управление доступами и мониторинг инцидентов
B. Аналитик данных	2. Построение дашбордов и бизнес-аналитика
C. Специалист по кибербезопасности	3. Обеспечение резервного копирования и восстановления
D. Руководитель проекта	4. Управление проектами и требования к ИТ-инфраструктуре

Ответ: A3B2C1D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность внедрения облачной аналитики в проект:

1. Сбор требований и выбор подходящей платформы
2. Развертывание окружения и настройка доступа
3. Импорт данных и создание первых аналитических пайплайнов
4. Визуализация результатов и интеграция в бизнес-процессы

Ответ: 1234

Установите последовательность обеспечения информационной безопасности при внедрении новой ИТ-системы:

1. Оценка рисков и требований к безопасности
2. Разработка политики доступа и учётных записей

3. Внедрение мер защиты и проведения тестирований
 4. Мониторинг, аудит и обновления
- Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики эффективны при применении информационных технологий в работе?

- A) Регулярное обновление ПО и патчей
 - B) Игнорирование резервного копирования
 - C) Документация процессов и стандартов
 - D) Игнорирование сроков и доступности систем
 - E) Повышение осведомлённости о безопасности
- Ответ: ACE

Какие методы повышения эффективности использования ИТ в команде стоит применять?

- A) Автоматизация повторяющихся задач
 - B) Единый набор инструментов и стандартов
 - C) Полная централизация без делегирования
 - D) Непроверяемые эксперименты с инфраструктурой
 - E) Контроль версий и аудит изменений
- Ответ: ABE

4. Прочитайте текст и дайте ответ:

Опишите подход к выбору и внедрению информационных технологий в проекте

Ответ:

Оценка требований и выбор технологий: собираю бизнес-цели, функциональные и нефункциональные требования, требования к безопасности и совместимости.

сравниваю кандидаты по критериям: функциональность, нагрузка, масштабируемость, стоимость владения, поддержка и совместимость с текущим стеком.

выбираю минимально жизнеспособное решение (MVP) с планом расширения и ясно прописанными условиями перехода.

Планирование внедрения и обучение: формирую дорожную карту внедрения, сроки, ответственных, требования к обучению и доступ к тренажерам/песочнице.

подготавливаю обучающие материалы, включающие руководства, чек-листы и примеры задач.

запускаю пилотную настройку, собираю обратную связь и корректирую план.

Измерение эффективности и адаптация подходов: устанавливаю KPI (время развёртывания, время простоя, производительность, удовлетворённость пользователей, уровень ошибок).

регулярно анализирую данные, провожу ретроспективы по внедрению и вношу коррективы в стек и процессы.

Опишите стратегию обеспечения устойчивости инфраструктуры и работы с данными через ИТ-инструменты

Ответ:

Управление данными, доступом и резервным копированием: применяю принцип наименьших прав доступа, регулярно обновляю учетные записи и политики доступа.

реализую резервное копирование с периодичностью и тестами восстановления, хранение копий в отдельных регионах/хранилищах.

обеспечиваю шифрование данных в состоянии покоя и при передаче, а также аудит доступа.

Соответствие регуляторам и стандартам: веду регламентируемый набор политик и процедур, привязанных к регуляторам (регламент обновления, контроль изменений, журналы аудита).

регулярно провожу внутренние аудиты, обзоры соответствия и обновляю документацию.

Реагирование на инциденты и восстановление сервиса: внедряю план реагирования на инциденты, четко распределяю роли и сроки реагирования.

выполняю восстановление по сценарию DRP/BCP, тестирую восстановление и документирую уроки.

информирую команду и руководство о статусе, причинах и мерах восстановления, обновляю регламенты по итогам инцидентов.

5. Прочитайте текст и дополните его:

Для успешной реализации проектов на основе ИТ важно обеспечивать _____, чтобы гарантировать предсказуемость процессов.

Ответ: регламентированность/структурированность/автоматизация

Применение _____ подходов к управлению данными и доступом повышает безопасность и надёжность.

Ответ: модульных/централизованных/регламентированных

Осваиваемая компетенция: ОК-10 Пользоваться профессиональной документацией на государственном и иностранном языках

Индикаторы достижения компетенций

Знание: правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности

Умение: понимать общий смысл четко произнесенных высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите типы документации с языками и целями использования.

Контексты	Языки/цели
А. Техническая документация (API, спецификации)	1. Государственный язык, формальный стиль
В. Пользовательская документация	2. Иностранный язык, техническая точность
С. Регуляторная/сертификационная документация	3. Язык проекта, понятность для команды
Д. Внутренняя документация для команды	4. Язык регуляторов, требования по соответствию

Ответ: A2B3C4D1

Соотнесите роли участников с типами документации, которыми они чаще работают.

Контексты	Документация
А. Разработчик	1. Технические спецификации и примеры использования
В. Ревьюер/архитектор	2. Тест-кейсы и регламенты качества
С. Менеджер проекта	3. План проекта и дорожная карта
Д. QA-инженер	4. Руководства по эксплуатации и обновлениям

Ответ: A1B4C3D2

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность поиска и использования документации на иностранном языке для проекта:

1. Определение необходимого объема и языков документации
2. Поиск источников и сбор материалов
3. Перевод и верификация терминологии
4. Внедрение и передача командам

Ответ: 1234

Установите последовательность подготовки и подачи документации на государственном языке для регуляторной проверки:

1. Сбор требований и регуляторных ссылок
2. Подготовка черновиков и защита версий
3. Внутреннее согласование и корректура
4. Публикация и архивирование

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики помогают корректно работать с документацией на разных языках?

- А) Использование единых глоссариев и терминологий
- В) Перевод без проверки носителями языка
- С) Верификация терминов через регуляторов и специалистов

- D) Игнорирование культурных различий в формулировках
 - E) Наличие примеров и контекстов использования
- Ответы: ACE

Какие подходы улучшают качество перевода технической документации?

- A) Машинный перевод без постредакции
 - B) Локализация с участием носителей языка
 - C) Привязка к оригинальным источникам и справочникам
 - D) Игнорирование контекстуальных примечаний
 - E) Периодическая переоценка терминологии
- Ответ: BCE

4. Прочитайте текст и дайте ответ:

Опишите подход к работе с профессиональной документацией на государственном и иностранном языках

Ответ:

Поиск и верификация источников:

формирую список целевых источников (официальная документация, регуляторные требования, справочники терминологии, надежные отраслевые публикации).

проверяю актуальность версии, дату обновления и соответствие контексту проекта; сверяю термины с регламентами.

Единообразие терминологии:

создаю единый глоссарий и glossary по проекту; внедряю контроль версий терминов.

пользуюсь стилевыми руководствами и словарями для разных языков; проводим вычитку носителями языка.

Оформление и распространение:

структурирую документы в едином репозитории (регистры изменений, база знаний); фиксирую версии и авторство.

рассылаю заказчикам и команде через утвержденные каналы, сопровождаю сопроводительными заметками и краткими сводками.

Опишите стратегию обеспечения качества перевода и соответствия документации требованиям регуляторов

Ответ:

Проверка языка и терминологии:

выполняю многократную вычитку, привлекаю носителей/редакторов, сверяю термины с глоссарием и регуляторными источниками.

применяю контроль качества перевода на предмет стилистики, точности и однозначности формулировок.

Учет регуляторных требований и обновления:

поддерживаю карту регуляторных требований и следую регламентам обновлений; фиксирую изменения в реестре версий.

регулярно проводим аудит соответствия и обновляем документацию в соответствии с требованиями.

Документация перевода и аудит:

веду журнал переводов с отметками о версиях, источниках и ответственных.

фиксирую принятые изменения, аргументацию и проверки качества; организую независимый аудит по расписанию.

Опишите процесс code review. Какие аспекты кода должны проверяться в первую очередь?

Ответ: Аспекты для проверки:

Соответствие требованиям (функциональность)

Соблюдение code style и стандартов

Качество архитектуры и дизайна

Эффективность алгоритмов

Обработка ошибок и edge cases

Тестируемость кода

Безопасность (SQL-инъекции и т.д.)

Комментарии и документация

Процесс:

Создание pull/merge request

Автоматические проверки (CI)

Ручной review коллегами

Обсуждение и внесение правок

Одобрение и слияние

5. Прочитайте текст и дополните его:

Для эффективной работы с документацией на иностранных языках важно соблюдать _____, чтобы обеспечить точность перевода.

Ответ: глоссарий/терминология/контекст

При подготовке регуляторной документации на государственном языке необходимо поддерживать _____ стиль и формат.

Ответ: официальный/формальный/регламентированный

Осваиваемая компетенция: ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.

Основные принципы процесса разработки программного обеспечения.

Основные подходы к интегрированию программных модулей.

Виды и варианты интеграционных решений.

Современные технологии и инструменты интеграции.

Основные протоколы доступа к данным.

Методы и способы идентификации сбоев и ошибок при интеграции приложений.

Методы отладочных классов.

Стандарты качества программной документации.

Основы организации инспектирования и верификации.

Встроенные и основные специализированные инструменты анализа качества программных продуктов.

Графические средства проектирования архитектуры программных продуктов.

Методы организации работы в команде разработчиков

Умения:

Анализировать проектную и техническую документацию.

Использовать специализированные графические средства построения и анализа архитектуры программных продуктов.

Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.

Определять источники и приемники данных.

Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace).

Оценивать размер минимального набора тестов.

Разрабатывать тестовые пакеты и тестовые сценарии.

Выявлять ошибки в системных компонентах на основе спецификаций.

Практический опыт:

Разрабатывать и оформлять требования к программным модулям по предложенной документации.

Разрабатывать тестовые наборы (пакеты) для программного модуля.

Разрабатывать тестовые сценарии программного средства.

Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите контексты взаимодействия между модулями с соответствующими аспектами требований.

Контексты	Аспекты требований
A. Архитектурная документация	1. Форматы данных и протоколы обмена
B. Спецификации интерфейсов	2. Совместимость и контракт между модулями
C. Регламенты интеграции	3. Стратегии обработки ошибок и надёжности интерфейсов
D. Требования к тестированию	4. План тестирования интеграции

Ответ: A2B1C3D4

Соотнесите роли участников проекта с типами артефактов взаимодействия.

Контексты	Артефакты
A. Архитектор решений	1. Контракты между модулями
B. Инженер по интеграции	2. Тест-кейсы для интерфейсов
C. Разработчик модуля	3. Архитектурные диаграммы взаимодействий
D. Тестировщик интеграций	4. План верификации совместимости

Ответ: A3B1C4D2

Установите соответствие между функциями IDE и их применением:

Функция IDE	Применение при анализе взаимодействий
-------------	---------------------------------------

Функция IDE	Применение при анализе взаимодействий
A) Анализ графа вызовов	1. Выявление зависимостей между модулями
B) Рефакторинг кода	2. Улучшение структуры кода
C) Визуализация UML	3. Документирование архитектуры
D) Автодополнение	4. Ускорение написания кода

Ответ: A1B2C3

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность формирования требований к взаимодействию между модулями:

1. Анализ существующей документации и контрактов
2. Выделение интерфейсов, форматов и ограничений
3. Формализация контрактов/протоколов обмена
4. Верификация и утверждение требований

Ответ: 1234

Установите последовательность верификации требований к взаимодействию между модулями:

1. Создание тест-кейсов на интерфейсы
2. Поэтапная интеграция и тестирование
3. Сбор обратной связи и коррекция контрактов
4. Финальная верификация и передача в производство

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие элементы стоит включать в требования к взаимодействию между модулями?.

- A) Форматы данных и протоколы обмена
- B) Язык программирования модулей
- C) Контракты и ожидания по задержкам
- D) Имя автора кода
- E) Требования к тестированию интерфейсов

Ответ: ACE

Какие методы минимизируют риски при разработке архитектуры взаимодействия?.

- A) Раннее моделирование контрактов
- B) Пакетная поставка без тестирования

- С) Поэтапная интеграция и пилотные запуски
 - Д) Игнорирование регламентов безопасности
 - Е) Документация версий и изменений
- Ответ: ACE

Какие функции современных IDE помогают анализировать взаимодействия между модулями?

- А. Визуализация зависимостей
- В. Автодополнение кода
- С. Анализ графа вызовов
- Д. Подсветка синтаксиса
- Е. Интеграция с Git

Ответ: AC

4. Прочитайте текст и дайте ответ:

Опишите подход к формированию требований к взаимодействию между модулями в проекте ИСДПО

Ответ:

Анализ документации и данных о взаимодействии:
изучаю существующие архитектурные документы, спецификации интерфейсов, регламенты обмена данными и реальный опыт прошлых интеграций;
выделяю точки взаимодействия, форматы сообщений, требования к задержкам и надёжности.

Формализация интерфейсов и контрактов:

определяю форматы данных, протоколы обмена, допустимые варианты ошибок и поведения при их возникновении;

фиксирую контракты между модулями (предусловия/последствия, входы/выходы, обязательные схемы взаимодействия) и привязываю к версиям API.

Трассируемость требований к тестам и внедрению:

связываю каждое требование с тест-кейсами и артефактами реализации;

веду карту трассируемости от требований через тесты к релизу и внедрению в окружение заказчика.

Опишите подход к верификации требований к взаимодействию между модулями

Ответ:

Методы тестирования (инструменты/скорости/покрытие):

комбинирую модульные тесты, интеграционные тесты и контрактное тестирование; применяю автоматизированные тесты в CI/CD; использую инструменты для симуляции нагрузки и ошибок;

оцениваю охват тестами по критическим путям взаимодействия и требованиям к производительности.

Документация результатов и коррекций:

фиксирую результаты тестирования в отчётах, регистрирую найденные дефекты и принятые решения по их устранению;

обновляю контракты/контексты взаимодействия при необходимости и регистрирую изменения в регистре версий.

Управление версиями контрактов и изменений:

версионировать интерфейсы и контракты, храню их в системе управления версиями, фиксирую дату утверждения и ответственных;

регламентирую процедуры изменения контрактов, требования к обзору и согласованию изменений с заинтересованными сторонами.

5. Прочитайте текст и дополните его:

Для надёжной интеграции между модулями важно поддерживать _____, чтобы обеспечить предсказуемость взаимодействий.

Ответ: контракт/совместимость/интерфейс

Формальные _____ между модулями облегчают автоматизацию тестирования и верификацию.

Ответ: контракты/интерфейсы/спецификации

Осваиваемая компетенция: ПК 2.2. Выполнять интеграцию модулей в программное обеспечение

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.

Основные принципы процесса разработки программного обеспечения.

Основные подходы к интегрированию программных модулей.

Основы верификации программного обеспечения.

Современные технологии и инструменты интеграции.

Основные протоколы доступа к данным.

Методы и способы идентификации сбоев и ошибок при интеграции приложений.

Основные методы отладки.

Методы и схемы обработки исключительных ситуаций.

Основные методы и виды тестирования программных продуктов.

Стандарты качества программной документации.

Основы организации инспектирования и верификации.

Приемы работы с инструментальными средствами тестирования и отладки.

Методы организации работы в команде разработчиков

Умения:

Использовать выбранную систему контроля версий.

Использовать методы для получения кода с заданной функциональностью и степенью качества.

Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов.

Использовать различные транспортные протоколы и стандарты форматирования сообщений.

Выполнять тестирование интеграции.

Организовывать постобработку данных.

Создавать классы-исключения на основе базовых классов.

Выполнять ручное и автоматизированное тестирование программного модуля.

Выявлять ошибки в системных компонентах на основе спецификаций.

Использовать приемы работы в системах контроля версий

Практический опыт:

Интегрировать модули в программное обеспечение.

Отлаживать программные модули.

Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите контексты интеграции модулей с их основными аспектами.

Контексты	Аспекты
A. Архитектура и интерфейсы	1. Определение форматов данных и протоколов обмена
B. Контракты между модулями	2. Условия совместного использования ресурсов и взаимодействий
C. Инструменты интеграции	3. Средства автоматизации сборки и развёртывания
D. Тестирование интеграции	4. План и кейсы тестирования интеграции

Ответ: A1B2C3D4

Соотнесите роли участников проекта с типами артефактов интеграции.

Контексты	Артефакты
A. Архитектор решений	1. Контракты между модулями
B. Инженер по интеграции	2. Архитектурные диаграммы взаимодействий
C. Разработчик модуля	3. План интеграции
D. QA-инженер	4. Протоколы тестирования интеграции

Ответ: A2B3C1D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность подготовки к интеграции модулей:

1. Анализ контрактов и интерфейсов
2. Подготовка окружения и инструментов
3. Реализация интерфейсов и контрактов
4. Интеграционное тестирование

Ответ: 1234

Установите последовательность проведения интеграционного цикла:

1. Запуск CI/CD и сбор артефактов
2. Интеграционное тестирование и фиксация дефектов
3. Верификация совместимости и регрессионное тестирование
4. Развертывание в окружение заказчика

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики улучшают качество интеграции модулей?

- A) Документация контрактов и интерфейсов
- B) Раннее тестирование взаимодействий
- C) Игнорирование версионирования API
- D) Единые подходы к мониторингу и логированию
- E) Регулярные ревью контрактов и изменений

Ответ: ABDE

Какие методы контроля интеграции наиболее эффективны?.

- A) Контроль версий контрактов
- B) Пилотные интеграции на небольших наборах
- C) Игнорирование регламентов безопасности
- D) Автоматизированное тестирование интеграций
- E) Мониторинг производительности пайплайна

Ответ: ABDE

Какие функции CI/CD систем важны для интеграции?

- A. Автоматическое разрешение зависимостей
- B. Генерация пользовательской документации
- C. Непрерывная сборка и тестирование
- D. Управление правами доступа
- E. Ведение списка задач

Ответ: AC

4. Прочитайте текст и дайте ответ:

Опишите подход к формированию требований к взаимодействию между модулями в проекте

Ответ:

анализ документов и данных о взаимодействии:

изучаю архитектурные документы, спецификации интерфейсов и регламенты обмена данными; выделяю точки соприкосновения, форматы сообщений, требования к задержкам и надёжности.

формализация интерфейсов и контрактов:

фиксирую форматы данных, протоколы обмена, допустимые ошибки и поведения; документирую предусловия/последствия и привязываю к версиям API.

трассируемость требований к тестам и внедрению:

связываю каждое требование с тест-кейсами и артефактами реализации; веду карту трассируемости от требований к тестам и релизу.

Опишите подход к верификации требований к взаимодействию между модулями.

Ответ:

документирование результатов и коррекций:

фиксирую результаты в отчётах, регистрирую дефекты и принятые решения; обновляю контракты и регистры версий.

управление версиями контрактов и изменений:

веду версионирование интерфейсов и контрактов, регламентирую изменения через обзор с заинтересованными сторонами, фиксирую даты и ответственных.

5. Прочитайте текст и дополните его:

Для надёжной интеграции между модулями важно поддерживать _____, чтобы обеспечить предсказуемость взаимодействий.

Ответ: контракты/интерфейсы/форматы

Формальные _____ между модулями облегчают автоматизацию тестирования и верификацию.

Ответ: контракты/интерфейсы/спецификации

Осваиваемая компетенция: ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.
Основные принципы процесса разработки программного обеспечения.
Основные подходы к интегрированию программных модулей.
Основы верификации и аттестации программного обеспечения.
Методы и способы идентификации сбоев и ошибок при интеграции приложений.
Основные методы отладки.
Методы и схемы обработки исключительных ситуаций.
Приемы работы с инструментальными средствами тестирования и отладки.
Стандарты качества программной документации.
Основы организации инспектирования и верификации.
Встроенные и основные специализированные инструменты анализа качества программных продуктов.
Методы организации работы в команде разработчиков

Умения:

Использовать выбранную систему контроля версий.
Использовать методы для получения кода с заданной функциональностью и степенью качества.
Анализировать проектную и техническую документацию.
Использовать инструментальные средства отладки программных продуктов.
Определять источники и приемники данных.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Использовать приемы работы в системах контроля версий.
Выполнять отладку, используя методы и инструменты условной компиляции.
Выявлять ошибки в системных компонентах на основе спецификаций

Практический опыт:

Отлаживать программные модули.
Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите типы специализированных средств с их основной функцией в отладке.

Контексты	Функции
A. Дебаггер	1. Пошаговый проход по коду и изменение состояния переменных
B. Профайлер	2. Определение узких мест в производительности
C. Инструмент трассировки	3. Отслеживание вызовов функций и путей выполнения
D. Инструмент анализа памяти	4. Выявление утечек и ошибок памяти

Ответ: A1B2C3D4

Соотнесите роли в процессе отладки с типами инструментов, которыми они чаще пользуются.

Контексты	Инструменты
A. Разработчик	1. Дебаггер и точки останова
B. Инженер по качеству/тестирования	2. Профайлеры и мониторы ресурсов
C. Инженер по производительности	3. Инструменты трассировки и репортинга ошибок
D. Релиз-инженер	4. Инструменты анализа памяти

Ответ: A1B3C2D4

Установите соответствие между функциями IDE и их применением при отладке:

Функция IDE	Применение при отладке
A) Точки останова	1. Приостановка выполнения
B) Просмотр переменных	2. Анализ состояния программы
C) Рефакторинг	3. Улучшение структуры кода
D) Автодополнение	4. Ускорение написания кода

Ответ: A1B2C3

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность подготовки к отладке модуля:

1. Повторимая фиксация проблемы и сбор репортов
2. Выбор инструмента для reproduce и анализа
3. Воспроизведение проблемы в тестовой среде
4. Внесение исправления и регресс-тестирование

Ответ: 1234

Установите последовательность цикла отладки в процессе разработки:

1. Анализ лога/снимков состояния
2. Применение исправления и локальная валидация
3. Выполнение регрессионного тестирования
4. Верификация в интеграционной среде и отзыв у команды

Ответ: 1234

Установите последовательность отладки в IDE (на примере Visual Studio Code):

1. Запуск отладчика (Debugger)
2. Установка условных точек останова
3. Просмотр значений переменных (Watch)
4. Анализ потока выполнения (Step Over/Into)
5. Исправление кода и повторный запуск

Ответ: 12435

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие инструменты чаще всего используются для эффективной отладки?

- A) Дебаггер с точками останова и watches
- B) Профайлеры для анализа времени выполнения
- C) Инструменты статического анализа кода
- D) Инструменты для автоматического стирания лога
- E) Инструменты для проверки памяти и утечек

Ответ: ABEC (в контексте целевой практики; можно выбрать любые сочетания, включая C)

Какие практики помогают снизить время повторного воспроизведения проблемы?

- A) Ведение подробных репортов и шагов воспроизведения
- B) Регулярное обновление тестовой среды
- C) Игнорирование прошлых инцидентов
- D) Использование фикстур и моков
- E) Автоматизация регресс-тестирования

Ответ: ABDE

Какие функции IDE помогают при отладке?

- A. Интерактивный отладчик с точками останова
- B. Автодополнение кода
- C. Просмотр стека вызовов (call stack)
- D. Генерация документации
- E. Интеграция с Git

Ответ: AC

4. Прочитайте текст и дайте ответ:

Подход к организации отладки модуля с использованием специализированных инструментов

Ответ:

Выбор инструмента:

учитываю тип проблемы (повторяемость, детальность трассировки, память, производительность);

сопоставляю возможности инструмента с требованиями проекта (скорость, совместимость, обучение команды);

выбираю минимально достаточный набор инструментов и план внедрения.

Формулировка reproduce-шагов:

подробно фиксирую условия воспроизведения (входные данные, окружение, конфигурации);

документирую ожидания и фактические результаты на каждом шаге;

сохраняю скриншоты/лог-файлы и применяю снапшоты состояния.

Фиксация и проверка исправлений:

регистрирую каждое исправление в регистре изменений и связанном артефакте;

повторно запускаю тесты, включая регрессию, подтверждая устранение проблемы;

регистрирую влияние исправления на производительность и стабильность.

Стратегия сотрудничества в команде во время отладки

Ответ:

Деление находок:

делаю краткие отчёты по каждому инциденту: причина, шаги воспроизведения, принятые решения;

обсуждаю выводы в стендапах, регламентированных встречах и через регистр решений.

Координация действий:

распределяю роли (отладчик, тестировщик, DevOps) и сроки выполнения;

применяю прозрачную систему статусов задач и зависимостей в системе трекинга.

Документирование для регламентного хранения:

веду единый журнал инцидентов с контекстом, принимаемыми мерами и ответными лицами;

сохраняю версии артефактов, тест-кейсов и отчетов в общедоступном репозитории для аудита и обучения.

5. Прочитайте текст и дополните его:

Для эффективной отладки важно иметь _____, чтобы воспроизводить ошибки и сравнивать результаты.

Ответ: чёткие шаги/конкретные репорты/контекст

Использование _____ памяти помогает выявлять утечки и повторяющиеся проблемы в приложении.

Ответ: инструменты анализа памяти/модули профилирования/детектор ошибок

Осваиваемая компетенция: ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.
Основные принципы процесса разработки программного обеспечения.
Основные подходы к интегрированию программных модулей.
Основы верификации и аттестации программного обеспечения.
Методы и способы идентификации сбоев и ошибок при интеграции приложений.
Методы и схемы обработки исключительных ситуаций.
Основные методы и виды тестирования программных продуктов.
Приемы работы с инструментальными средствами тестирования и отладки.
Стандарты качества программной документации.
Основы организации инспектирования и верификации.
Встроенные и основные специализированные инструменты анализа качества программных продуктов.
Методы организации работы в команде разработчиков

Умения:

Использовать выбранную систему контроля версий.
Анализировать проектную и техническую документацию.
Выполнять тестирование интеграции.
Организовывать постобработку данных.
Использовать приемы работы в системах контроля версий.
Оценивать размер минимального набора тестов.
Разрабатывать тестовые пакеты и тестовые сценарии.
Выполнять ручное и автоматизированное тестирование программного модуля.
Выявлять ошибки в системных компонентах на основе спецификаций

Практический опыт:

Разрабатывать тестовые наборы (пакеты) для программного модуля.
Разрабатывать тестовые сценарии программного средства.
Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите элементы тестового набора с их ролью в тестировании.

Контексты	Роли
A. Тест-кейсы	1. Определяют объём и график тестирования
B. Тест-планы	2. Описывают последовательность действий для проверки конкретной функции
C. Тестовые сценарии	3. Описывают входные наборы данных и ожидаемые результаты
D. Тестовые данные	4. Используются для подготовки конкретных тестов и наборов данных

Ответ: A3B1C2D4

Соотнесите типы тестирования с их целью.

Контексты	Цели
A. Юнит-тестирование	1. Проверка взаимодействия модулей
B. Интеграционное тестирование	2. Проверка отдельных функций и методов
C. Регрессионное тестирование	3. Проверка сохранения работоспособности после изменений
D. Приемочное тестирование	4. Проверка соответствия требованиям заказчика

Ответ: A2B1C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность разработки тестовых наборов:

1. Определение тестовых требований и рисков
2. Проектирование тестовых наборов и сценариев
3. Подготовка тестовых данных
4. Выполнение тестирования и документирование результатов

Ответ: 1234

Установите последовательность внедрения тестирования в процесс разработки:

1. Формирование плана автоматизации
2. Создание тестовых скриптов и конфигураций CI
3. Интеграция тестирования в пайплайн
4. Мониторинг результатов и оптимизация

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики улучшают качество тестирования тестовых наборов?

- A) Раннее участие тестировщиков в планировании
- B) Повторное использование тестовых данных без контроля версии
- C) Нормализация и документирование входных данных
- D) Регулярная ревизия тест-кейсов и обновление по изменениям требований
- E) Игнорирование фидбэка от разработчиков

Ответ: ACD

Какие подходы повышают устойчивость тестовых сценариев к изменениям в кодовой базе?

- A) Модульное тестирование и изоляция тестов
- B) Жёсткая зашивка тестов под конкретные реализации
- C) Использование параметизированных тестов
- D) Пренебрежение тестовыми данными
- E) Автоматизация запуска тестов в CI

Ответ: ACE

Какие функции CI/CD систем важны для тестирования?

- A. Автоматический запуск тестов
- B. Генерация документации
- C. Отчеты о покрытии кода
- D. Управление пользователями
- E. Мониторинг серверов

Ответ: ACE

4. Прочитайте текст и дайте ответ:

Опишите ваш подход к разработке тестовых наборов и тестовых сценариев для ПО

Ответ:

Формирование требований и выбор типов тестирования:

анализирую требования продукта и нефункциональные требования (производительность, безопасность, совместимость).

выбираю типы тестирования по рискам и критичности функций (юнит, интеграционный, функциональный, регрессионный, приемочный, нагрузочный).

фиксирую критерии приемки и метрики качества для каждого теста.

Проектирование тестовых сценариев и данных:

разрабатываю сценарии на основе реального поведения пользователей и контрактов между модулями.

создаю наборы тестовых данных с учётом пограничных случаев и негативных сценариев; документирую источники данных.

применяю принцип независимости тестов, параметризацию и реплики тестовых условий.

Документация и версионирование тестов:

храню тест-кейсы, данные и результаты в едином репозитории с версионированием.

веду связку тестов с требованиями и артефактами разработки (traceability).

регулярно обновляю документацию после изменений требований и рефакторинга.

Опишите стратегию обеспечения качества тестирования в проекте

Ответ:

Планирование автоматизации и среды тестирования:

оцениваю целесообразность автоматизации для каждого типа теста, выбираю стек инструментов и создаю тестовую среду (контейнеризация/виртуализация).

строю CI/CD пайплайны с запуском тестов на каждом шаге сборки и релиза.

обеспечиваю изолированную тестовую среду, данные и повторяемость окружения.

Оценка охвата и качества тестов:

измеряю покрытие функциональности, регрессионный охват, время выполнения тестов и стабильность результатов.

проводжу регулярные ревью тестов, анализ дефектов и внедряю улучшения.

Управление изменениями тестов при эволюции требований:

применяю контроль версий тестов, регистр изменений и затронутые требования.

автоматизирую обновления тестов при изменениях требований и поддерживаю регламент внесения изменений.

информирую команду о impact и пересматриваю критерии приемки.

5. Прочитайте текст и дополните его:

Для обеспечения повторяемости тестов важно использовать _____, чтобы любые тесты можно было воспроизвести.

Ответ: данные/среды/сценарии

В процессе разработки тестов применяют _____, чтобы гарантировать корректность изменений.

Ответ: регулярные ревью/контроль версий/автоматизированные проверки

Осваиваемая компетенция: ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Индикаторы достижения компетенций

Знания:

Модели процесса разработки программного обеспечения.
Основные принципы процесса разработки программного обеспечения.
Основные подходы к интегрированию программных модулей.
Основы верификации и аттестации программного обеспечения.
Стандарты качества программной документации.
Основы организации инспектирования и верификации.
Встроенные и основные специализированные инструменты анализа качества программных продуктов.
Методы организации работы в команде разработчиков

Умения:

Использовать выбранную систему контроля версий.
Использовать методы для получения кода с заданной функциональностью и степенью качества.
Анализировать проектную и техническую документацию.
Организовывать постобработку данных.
Приемы работы в системах контроля версий.
Выявлять ошибки в системных компонентах на основе спецификаций

Практический опыт:

Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования

1. Прочитайте текст и установите соответствие. Запишите выбранные ответы без пробелов:

Соотнесите элементы инспекции с целью проверки кодирования.

Контексты	Цели
А. Стратегии форматирования кода	1. Улучшение читаемости и поддерживаемости кода
В. Правила именования переменных и функций	2. Обеспечение единообразия в проекте
С. Правила комментирования и документации	3. Обеспечение прозрачности и объяснимости кода
Д. Проверка соблюдения стиля и	4. Выявление несоответствий в

конвенций	стиль кодирования
-----------	-------------------

Ответ: A2B1C3D4

Соотнесите роли в инспекции с их обязанностями.

Контексты	Обязанности
A. Инспектор кода	1. Обеспечивает применение правил кодирования и отслеживает нарушения
B. Редактор стиля	2. Проводит обсуждения замечаний и финализирует стиль проекта
C. Руководитель качества	3. Определяет требования к качеству и планирует мероприятия по улучшению
D. Разработчик	4. Исправляет замечания и согласовывает финальные изменения кода

Ответ: A1B2C3D4

2. Прочитайте текст и установите последовательность. Цифры ответов запишите в правильной последовательности без пробелов:

Установите последовательность проведения инспекции кода на соответствие стандартам:

1. Подготовка чек-листов и критериев соответствия
2. Выполнение инспекции и фиксация нарушений
3. Обсуждение замечаний и формирование плана исправлений
4. Внесение изменений и повторная инспекция

Ответ: 1234

Установите последовательность подготовки и проведения автоматического инспекционного скрининга:

1. Настройка инструментов статического анализа
2. Запуск скрининга на базовом репозитории
3. Анализ результатов и фильтрация ложных срабатываний
4. Внесение корректировок в правила и повторный прогон

Ответ: 1234

3. Прочитайте текст и выберите все правильные ответы. Запишите выбранные ответы без пробела:

Какие практики улучшают качество инспекции кода на соответствие стандартам?.

- A) Сопоставление результатов инспекции с чек-листами
- B) Игнорирование замечаний без обоснования
- C) Регулярные обновления стиля и конвенций
- D) Включение разработчиков в процесс инспекции
- E) Автоматизация проверки стилей и правил

Ответ: ACDE

Какие методы проверки соответствия кодирования наиболее эффективны?

- A) Статический анализ и линтинг

- В) Ручная инспекция без регламентов
 - С) Партнёры-ревью с параллельной проверкой
 - Д) Регрессионное тестирование без учета стиля
 - Е) Контроль версий правил и журнал изменений
- Ответ: ACE

Какие функции CI/CD важны для контроля стандартов кода?

- A. Автоматический запуск анализаторов кода
- B. Развертывание в облако
- C. Блокировка мердж-реквестов при нарушениях
- D. Генерация пользовательской документации
- E. Мониторинг нагрузки на сервера

Ответ: AC

4. Прочитайте текст и дайте ответ:

Напишите подход к организации инспекции кода на соответствие стандартам

Ответ:

Выбор применимых стандартов и правил:

анализ требований проекта и регламентов компании; сопоставление с общепринятыми стилями (код-стайл) и отраслевыми стандартами.

формирование набора правил в виде чек-листов и настройка инструментов на их соответствие.

Документация замечаний и исправлений:

ведение журнала замечаний с описанием проблемы, ссылки на артефакты и сроки устранения; фиксация принятых решений.

связь исправлений с соответствующими тестами и версиями кода; ретроспектива по итогам инспекции.

Обучение команды по стилю:

проведение коротких обучающих сессий и примеров типичных ошибок; обновление руководств и глоссариев.

внедрение периодических ревью стиля в ретроспективах и обучающие демо-сигнализации.

Опишите стратегию внедрения автоматического статического анализа в проект

Ответ:

Инструменты и зачем:

выбрать набор инструментов для статического анализа, линтинга и проверки безопасности, настроить под стек проекта; обосновать выбор по охвату и скорости.

Настройка правил и порогов предупреждений:

определить критичность ошибок, пороги для блокировки сборки, адаптировать правила под язык и фреймворк; документировать параметры.

Интеграция в сборку и ревью кода:

встроить анализ в CI/CD, обеспечить автоматические отчеты и сигналы о нарушениях; связать результаты с процессом кода-ревью и регламентами выпуска.

5. Прочитайте текст и дополните его:

Для эффективной инспекции кода важно иметь _____, чтобы обеспечить единообразие и предсказуемость.

Ответ: чёткие правила/политику/стандарты

Автоматический анализ кода обеспечивает _____ выявление нарушений и снижение рисков.

Ответ: раннее/постоянное/регулярное