ГОУ СПО «Тираспольский техникум информатики и права»

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**

**по профессиональному модулю**

**ПМ.03 Участие в интеграции программных модулей**

Наименование профессионального модуля

**2.09.02.03 «Программирование в компьютерных системах»**

код, наименование профессии/специальности

Тирасполь, 2022

|  |  |
| --- | --- |
| ОДОБРЕН  Цикловой методической  комиссией\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *наименование комиссии*  Протокол № \_\_\_\_  от «\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_\_ г. | Разработан в соответствии с ГОС по специальности 2.09.02.03 Программирование в компьютерных системах |
| Председатель цикловой методической комиссии  \_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_ | Заместитель директора по методической работе (заведующий методическим отделом, методист)  \_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_ |

Заместитель директора учебно-производственной работе

\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_

Составители: \_ Е.Н. Парван , преподаватель информационных дисциплин, ГОУ СПО «ТТИиП»

Ф.И.О., должность, категория, организация профессионального образования

Е.Г. Белоус, преподаватель информационных дисциплин, первой кв. категории, ГОУ СПО «ТТИиП»

Ф.И.О., должность, категория, организация профессионального образования

Т.С. Новакова, преподаватель информационных дисциплин, ГОУ СПО «ТТИиП»

Ф.И.О., должность, категория, организация профессионального образования

Согласовано: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ф.И.О., должность, наименование предприятия (организац

**ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ**

**по профессиональному модулю**

**ПМ.03 Участие в интеграции программных модулей**

**по специальности**

**2.09.02.03 «Программирование в компьютерных системах»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Приобретенный практический опыт, освоенные умения, усвоенные знания** | **Результаты**  **обучения ПК, ОК** | **Основные показатели**  **оценки результата** | **Наименование тем раздела, МДК, темы, подтемы** | **Уровень освоения** | **Наименование**  **контрольно-оценочного средства** | |
| **Текущий контроль** | **Промежуточная**  **Аттестация**  **(согласно рабочему учебному плану)** |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| **МДК.03.01 Технология разработки программного обеспечения** | | | | | | |
| **иметь практический опыт:**   * участия в выработке требований к программному обеспечению; * участия в проектировании программного обеспечения с использованием специализированных программных пакетов;   **уметь***:*   * владеть основными методологиями процессов разработки программного обеспечения; * использовать методы для получения кода с заданной функциональностью и степенью качества;   **знать***:*   * модели процесса разработки программного обеспечения; * основные принципы процесса разработки программного обеспечения; * основные подходы к интегрированию программных модулей; * основные методы и средства эффективной разработки; * основы верификации и аттестации программного обеспечения; * концепции и реализации программных процессов; * принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного обеспечения; * методы организации работы в коллективах разработчиков программного обеспечения; * основные положения метрологии программных продуктов, принципы построения, проектирования и использования средств для измерений характеристик и параметров программ, программных систем и комплексов; * стандарты качества программного обеспечения; * методы и средства разработки программной документации. | ПК3.1 Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения. | * точность понимания поставленной задачи; * обоснование требований к программному обеспечению; * качество рекомендаций по формализации предметной области с учетом ограничений; * оценивать уровень сложности компонент ПО; * точность и грамотность понимания проектной программной документации на уровне взаимодействия компонент ПО. | **Тема 1.**  **Определение технологии**  **конструирования программного обеспечения** Тема 1.1 Классический жизненный цикл Макетирование. Стратегии конструирования ПО: Инкрементная модель, Быстрая разработка приложений.  Тема 1.2 Стратегии конструирования ПО:  Спиральная модель. Компонентно-ориентированная модель  Тема 1.3. Тяжеловесные и облегченные процессы ХР-процесс  Тема 1.4. Модели качества процессов конструирования | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  Тест по разделу  **Практические занятия №1,2,3**  Разработка, описание и анализ информации **Самостоятельная работа обучающихся**.  Составление сводной таблицы по характеристикам программных продуктов. Характеристика программ общего, проблемно-ориентированного и специального назначения | Экзаменационные  билеты  для устного  (письменного)  экзамена по МДК |
| ПК3.2 Выполнять интеграцию модулей в программную систему. | * изложение основных характеристик программной системы; * использование методов для получения кода с заданной функциональностью и степенью качества; * определение этапов проектирования программных систем и их архитектуры; * изложение подходов к интегрированию программных модулей; * использование методов и средств эффективной разработки; | **Тема 2. Руководство программным проектом** Тема 2.1 Процесс руководства проектом Планирование проектных задач  Тема 2.2 Размерно-ориентированные метрики. Функционально-ориентированные метрики.Тема 2.3 Выполнение оценки в ходе руководства проектом Выполнение оценки проекта на основе LOC- и FP-метрик Конструктивная модель стоимости Предварительная оценка программного проекта Анализ чувствительности программного проекта | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  Тест по разделу  **Практические занятия №4-6**  Разработка технического задания к программному обеспечению **Самостоятельная работа обучающихся**.  Составление технического задания на создание сайта |
| ПК3.3 Выполнять отладку программного продукта с использованием специализированных программных средств. | * изложение основных характеристик программной системы; * использование методов для получения кода с заданной функциональностью и степенью качества; * скорость и качество отладки программного кода; * изложение основных положений метрологии программных продуктов; * использование методов и средств эффективной разработки; | **Тема 3. Классические**  **методы анализа**  Тема 3.1. Структурный анализ. Методы анализа, ориентированные на структуры данных  Тема 3.2. Метод анализа Джексона.  Методика Джексона | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практические занятия №7-9**  Разработка требований к программному обеспечению |
| ПК3.4 Осуществлять разработку тестовых наборов и тестовых сценариев. | * изложение основных методик тестирования программных компонент и системы в целом; * организацию процесса тестирования; * составление тестовых заданий; * выбор алгоритма тестирования; * скорость и качество тестирования программной системы и отдельных компонент | **Раздел 4. Основы проектирования программных систем**  Тема 4.1 Особенности процесса синтеза программных систем Особенности этапа проектирования  Тема 4.2 Структурирование системы Моделирование управления Декомпозиция подсистем на модули  Тема 4.3 Модульность Информационная закрытость Связность модуля Сцепление модулей  Тема 4.4 Сложность программной системы Характеристики иерархической структуры программной системы | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практические занятия №10**  Методология функционального моделирования  **Самостоятельная работа обучающихся**.  Составление кроссворда на тему «Этапы проектирования программных систем»  Подготовка реферата. |
| ПК3.5 Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования. | * изложение основных характеристик программной системы; * владение основными методологиями процессов разработки программного обеспечения; * изложение основ верификации и аттестации программного обеспечения» * использование стандартов кодирования; * использование методов для получения кода с заданной функциональностью и степенью качества; * изложение основных положений метрологии программных продуктов; * использование методов и средств эффективной разработки; | **Тема 5. Классические методы проектирования**  Тема 5.1 Метод структурного проектирования.  Тема 5.2 Метод проектирования Джексона | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу. Тест по разделу  **Практические занятия №11-14**  Функциональная модель SADT  Моделирование IDF3  Диаграммы потока данных DFD  **Самостоятельная работа обучающихся**.  Подготовка схемы, сравнение методов структурного проектирования  Подготовка сообщение доклада, презентации |
|  |  | **Тема 6.**  **Структурное тестирование программного обеспечения**  Тема 6.1 Основные понятия и принципы тестирования ПО Особенности тестирования «белого ящика».  Тема 6.2 Способ тестирования базового пути. Способы тестирования условий  Тема 6.3 Способ тестирования потоков данных. Тестирование циклов | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №15**  Тестирование «Белого ящика»  **Самостоятельная работа обучающихся**.  Подготовка Реферата, решение задач |
|  | ПК 3.6 Разрабатывать технологическую документацию. | * качество рекомендаций по формализации поставленной задачи; * качество и скорость чтения технической документации; * выработка рекомендаций по использованию стандартов оформления документации; * точность и грамотность оформления программной технологической документации. | **Тема 7. Функциональное тестирование программного обеспечения**  Тема 7.1 Особенности тестирования «черного ящика».  Способ разбиения по эквивалентности.  Тема 7.2 Способ анализа граничных значений  Тема 7.3 Способ диаграмм причин-следствий | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №16**  Тестирование «Черного ящика» |
| ОК 1 Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес |  | **Тема 8. Организация**  **процесса тестирования программного обеспечения**  Тема 8.1 Методика тестирования программных систем  Тестирование элементов.  Тема 8.2 Тестирование интеграции. Тестирование правильности  Системное тестирование. Искусство отладки | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №17**  NUnit тестирование |
| ОК 2 Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество | * демонстрация интереса к будущей профессии | **Тема 9. Основы объектно- ориентированного представления программных систем**  Тема 9. 1. Принципы объектно- ориентированного представления программных систем.  Тема 9. 2. Объекты.  Тема 9. 3. Классы | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу |
| ОК 3 Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность | * выбор и применение методов и способов решения профессиональных задач в области программной инженерии; | **Тема 10. Базис языка визуального моделирования**  Тема 10.1.Унифицированный язык моделирования  Тема 10.2. Предметы в UML. Отношения в UML  Тема 10.3.Диаграммы в UML. Механизмы расширения в UML | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №22-23**  Методология объектно-ориентированного программирования  **Самостоятельная работа обучающихся**.  Подготовка Реферата, решение задач |
| ОК 4 Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития | * решение стандартных и нестандартных профессиональных задач в области интеграции компонент программного обеспечения компьютерных систем; | **Тема 11. Статические модели объектно-ориентированных программных систем**  Тема 11.1. Вершины в диаграммах классов Отношения в диаграммах классов  Тема 11.2. Деревья наследования Примеры диаграмм классов | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №21**  Диаграмма классов |
| ОК 5 Использовать информационно-коммуникационные технологии в профессиональной деятельности | * эффективный поиск необходимой информации; * использование различных источников, включая электронные; | **Тема 12. Динамические модели объектно- ориентированных программных систем**  Тема 12.1. Моделирование поведения программной системы Диаграммы схем состояний Диаграммы деятельности  Тема 12.2. Диаграммы взаимодействия Диаграммы сотрудничества Диаграммы последовательности  Тема 12.3. Диаграммы UseCase Спецификация элементов UseCase Пример диаграммы UseCase Построение модели требований Кооперации и паттерны Бизнес-модели | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №22, 23**  Временная диаграмма  Концептуальные диаграммы **Самостоятельная работа обучающихся**.  Подготовка сообщения |
|  | ОК 6 Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями | * использование методов и средств разработки компонент программной системы | **Тема 13. Модели реализации**  **объектно- ориентированных программных систем**  Тема 13. 1. Компонентные диаграммы Использование компонентных диаграмм. Основы компонентной объектной модели. | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №24, 25**  Диаграммы размещения **Самостоятельная работа обучающихся**.  Подготовка реферата |
|  | ОК 7 Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий | * взаимодействие с обучающимися, преподавателями и мастерами в ходе обучения | **Тема 14. Метрики объектно-**  **ориентированных**  **программных**  **систем**  Тема 14.1. Метрические особенности объектно-ориентированных программных систем  Тема 14.2. Эволюция мер связи для объектно-ориентированных программных систем  Тема 14.3. Набор метрик Чидамбера и Кемерера. Использование метрик Чидамбера-Кемерера  Метрики Лоренца и Кидда.  Тема 14.4. Набор метрик Фернандо Абреу  Метрики для объектно-ориентированного тестирования | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Самостоятельная работа обучающихся**.  Подготовка рефератов |
|  | ОК8 Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации | * самоанализ и коррекция результатов собственной работы по завершению цикла практических работ, направленных на формирование командного стиля разработки | **Тема 15. Унифицированный**  **процесс разработки**  **объектно-ориентированных ПС**  Тема 15.1. Эволюционно-инкрементная организация жизненного цикла разработки.  Тема 15.2. Управление риском Этапы унифицированного процесса разработки.  Тема 15.3. Пример объектно-ориентированной разработки. Разработка в стиле экстремального программирования | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №26**  Разработка в стиле экстремального программирования  **Самостоятельная работа обучающихся**.  Подготовка инструкции по эксплуатации программного обеспечения |
|  | ОК 9 Ориентироваться в условиях частой смены технологий в профессиональной деятельности | * организация самостоятельных занятий при изучении профессионального модуля * анализ инноваций в области разработки модулей программного обеспечения и их интеграции в компьютерную систему, изучение периодических изданий | **Тема 16. Объектно-ориентированное тестирование**  Тема 16.1. Расширение области применения объектно-ориентированного тестирования Изменение методики при объектно-  ориентированном тестировании  Тема 16.2. Проектирование объектно-ориентированных тестовых вариантов. Способы тестирования содержания класса  Тема 16.3. Способы тестирования взаимодействия классов Предваряющее тестирование при экстремальной разработке | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №27, 28**  Проектирование ОО тестовых вариантов  Тестирования содержания класса **Самостоятельная работа обучающихся**.  Подготовка рефератов,сравнении CASE-средств: ARIS и RationalRose |
|  | ОК 10. Исполнять воинскую обязанность, в том числе с применением полученных профессиональных знаний (для юношей) | -перечисляет основные виды вооружения военной техники и специального снаряжения, состоящих на вооружении (оснащении) воинских подразделений | **Тема 17. Автоматизация конструирования визуальной модели программной системы**  Тема 17.1. Общая характеристика CASE- системы RationalRose  Тема 17.2. Создание диаграммы UseCase.  Создание диаграммы последовательности. Создание диаграммы классов  Тема 17.3. Создание компонентной диаграммы. Генерация программного кода | 1,2,3 | Вопросы для устного  (письменного) опроса  по разделу  **Практическое занятие №29-37**  Создание диаграммы USE CASE  Создание диаграммы последовательности  Создание диаграммы классов  Создание компонентной диаграммы  Генерация программного кода  **Самостоятельная работа обучающихся**.  Подготовка технической документации |

ГОУ СПО «Тираспольский техникум информатики и права»

**КОНТРОЛЬНО - ОЦЕНОЧНЫЕ СРЕДСТВА**

**для текущего контроля знаний, умений обучающихся**

**по междисциплинарному курсу**

**МДК.03.02 «Инструментальные средства разработки программного обеспечения»**

**по специальности**

**2.09.02.03 «Программирование в компьютерных системах»**

код, наименование профессии/специальности

Тирасполь, 2022

**2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ, УМЕНИЙ ОБУЧАЮЩИХСЯ**

**2.1. Критерии оценки работы обучающегося на практической работе**

**Устный ответ**

Оценка знаний предполагает дифференцированный подход к обучающемуся, учет его индивидуальных способностей, степень усвоения и систематизации основных понятий и категорий по дисциплине. Кроме того, оценивается не только глубина знаний поставленных вопросов, но и умение использовать в ответе практический материал.

Оценивается культура речи, владение навыками ораторского искусства.

Критерии оценивания: последовательность, полнота, логичность изложения, анализ различных точек зрения, самостоятельное обобщение материала, использование профессиональных терминов, культура речи, навыки ораторского искусства. Изложение материала без фактических ошибок.

Оценка «отлично» ставится в случае, когда материал излагается исчерпывающе, последовательно, грамотно и логически стройно, при этом раскрываются не только основные понятия, но и анализируются точки зрения различных авторов. Обучающийся не затрудняется с ответом, соблюдает культуру речи.

Оценка «хорошо» ставится, если обучающийся твердо знает материал, грамотно и по существу излагает его, знает практическую базу, но при ответе на вопрос допускает несущественные погрешности.

Оценка «удовлетворительно» ставится, если обучающийся освоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении материала, затрудняется с ответами, показывает отсутствие должной связи между анализом, аргументацией и выводами.

Оценка «неудовлетворительно» ставится, если обучающийся не отвечает на поставленные вопросы.

**Исследовательский проект (реферат)**

Исследовательский проект – проект, структура которого приближена к формату научного исследования и содержит доказательство актуальности избранной темы, определение научной проблемы, предмета и объекта исследования, целей и задач, методов, источников, историографии, обобщение результатов, выводы.

Результаты выполнения исследовательского проекта оформляется в виде реферата.

Критерии оценивания - поскольку структура исследовательского проекта максимально приближена к формату научного исследования, то при выставлении учитывается доказательство актуальности темы исследования, определение научной проблемы, объекта и предмета исследования, целей и задач, источников, методов исследования, выдвижение гипотезы, обобщение результатов и формулирование выводов, обозначение перспектив дальнейшего исследования.

Оценка «отлично» ставится в случае, когда обучающийся демонстрирует полное понимание проблемы, все требования, предъявляемые к заданию выполнены.

Оценка «хорошо» ставится, если обучающийся демонстрирует значительное понимание проблемы, все требования, предъявляемые к заданию выполнены.

Оценка «удовлетворительно» ставится, если обучающийся, демонстрирует частичное понимание проблемы, большинство требований, предъявляемых к заданию, выполнены

Оценка «неудовлетворительно» ставится, если обучающийся демонстрирует непонимание проблемы, многие требования, предъявляемые к заданию, не выполнены.

**Информационный проект (сообщение, доклад с презентацией)**

Информационный проект – проект, направленный на стимулирование учебно-познавательной деятельности студента с выраженной эвристической направленностью (поиск, отбор и систематизация информации об объекте, оформление ее для презентации).

Информационный проект отличается от исследовательского проекта, поскольку представляет собой такую форму учебно-познавательной деятельности, которая отличается ярко выраженной эвристической направленностью.

Критерии оценивания - при выставлении оценки учитывается самостоятельный поиск, отбор и систематизация информации, раскрытие вопроса (проблемы), ознакомление студенческой аудитории с этой информацией (представление информации), ее анализ и обобщение, оформление, полные ответы на вопросы аудитории с примерами.

Оценка «отлично» ставится в случае, когда обучающийся полностью раскрывает вопрос (проблему), представляет информацию систематизировано, последовательно, логично, взаимосвязано, использует более 5 профессиональных терминов, широко использует информационные технологии, ошибки в информации отсутствуют, дает полные ответы на вопросы аудитории с примерами.

Оценка «хорошо» ставится, если обучающийся раскрывает вопрос (проблему), представляет информацию систематизировано, последовательно, логично, взаимосвязано, использует более 2 профессиональных терминов, достаточно использует информационные технологии, допускает не более 2 ошибок в изложении материала, дает полные или частично полные ответы на вопросы аудитории.

Оценка «удовлетворительно» ставится, если обучающийся, раскрывает вопрос (проблему) не полностью, представляет информацию не систематизировано и не совсем последовательно, использует 1-2 профессиональных термина, использует информационные технологии, допускает 3-4 ошибки в изложении материала, отвечает только на элементарные вопросы аудитории без пояснений.

Оценка «неудовлетворительно» ставится, если вопрос не раскрыт, представленная информация логически не связана, не используются профессиональные термины, допускает более 4 ошибок в изложении материала, не отвечает на вопросы аудитории.

**Правила защиты практической работы**

Практическая работа заканчивается ее защитой. Во время защиты преподаватель выясняет степень усвоения материала и полученных навыков. Основная цель защиты – демонстрация студентом полученных при выполнении практической работы умений, знаний, компетенций. При защите практической работе студент должен:

- кратко сформулировать цель работы и порядок ее проведения;

- четко пояснять назначение, цель и результат каждого этапа разработки программы;

- показать навыки работы с интегрированной средой программирования;

- отвечать на вопросы преподавателя, связанные с теоретическими моментами практической работы;

- продемонстрировать работающую программу с заранее подготовленными тестовыми наборами;

- выполнять анализ полученных программой результатов;

- отвечать на вопросы преподавателя по разработанному программному коду;

- сделать вывод о проделанной работе.

1.5. Критерии оценивания практической работы

По результатам защиты практической работы, происходит ее оценивание. Основными критерии для оценивания практической работы являются:

- объем выполненной работы в процентах;

- правильность функционирования программы по разработанным сценариям;

- количество успешных ответов у преподавателя на защите;

- качество оформления программного кода.

Оценку «Отлично» студент получает, если:

-практическая работа выполнена в полном объеме;

-свободно ориентируется в программном коде;

-отвечает на все вопросы преподавателя.

Оценку «Хорошо» студент получает, если:

-практическая работа в основном выполнена, за исключением незначительных недоработок;

-отвечает почти на все вопросы преподавателя.

Оценку «Удовлетворительно» студент получает, если:

-практическая работа выполнена не в полном объеме и решает только часть поставленной задачи;

-многие вопросы у студента вызывают сложности при ответе.

Оценку «Не удовлетворительно» студент получает, если:

-отсутствует результирующий программный продукт;

-не ориентируется в поставленной задаче и способах ее решения;

-отсутствует заинтересованность в получении умений и знаний.

**Тестирование**

Является одним из средств контроля знаний обучающихся по дисциплине

Критерии оценивания – правильный ответ на вопрос

Оценка «отлично» ставится в случае, если правильно выполнено 90-100% заданий.

Оценка «хорошо» ставится, если правильно выполнено 70-89% заданий.

Оценка «удовлетворительно» ставится в случае, если правильно выполнено 50-69% заданий.

Оценка «неудовлетворительно» ставится, если правильно выполнено менее 50% заданий.

Контрольная работа

Оценивается не только глубина знаний поставленных вопросов, но и умение изложить письменно.

Критерии оценивания: последовательность, полнота, логичность изложения, анализ различных точек зрения, самостоятельное обобщение материала. Изложение материала без фактических ошибок.

Оценка «отлично» ставится в случае, когда соблюдены все критерии.

Оценка «хорошо» ставится, если обучающийся твердо знает материал, грамотно и по существу излагает его, знает практическую базу, но допускает несущественные погрешности.

Оценка «удовлетворительно» ставится, если обучающийся освоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении материала, затрудняется с ответами, показывает отсутствие должной связи между анализом, аргументацией и выводами.

Оценка «неудовлетворительно» ставится, если обучающийся не отвечает на поставленные вопросы.

**2.2. Содержание контрольно-оценочных средств для текущего контроля знаний, умений обучающихся**

**Раздел 1. Метод моделирования как инструмент разработки программных систем.**

Тема 1.1 Принципы имитационного моделирования. История вопроса.

Тема 1.2 Этапы создания имитационной модели

**Вопросы для устного опроса:**

**Тест по разделу «Метод моделирования как инструмент разработки программных систем»**

|  |
| --- |
| **Практическое занятие № 1.**  **Изучение интерфейса и возможностей графического конструктора.** |

**Тема 1. Определение технологии конструирования программного обеспечения**

Тема 1.1. Классический жизненный цикл Макетирование. Стратегии конструирования ПО: Инкрементная модель, Быстрая разработка приложений

Тема 1.2. Стратегии конструирования ПО:

Спиральная модель. Компонентно-ориентированная модель

Тема 1.3. Тяжеловесные и облегченные процессы ХР-процесс

Тема 1.4. Модели качества процессов конструирования

**Вопросы для устного опроса:**

1. Дайте определение технологии конструирования программного обеспечения.
2. Какие этапы классического жизненного цикла вы знаете?
3. Охарактеризуйте содержание этапов классического жизненного цикла.
4. Объясните достоинства и недостатки классического жизненного цикла.
5. Чем отличается классический жизненный цикл от макетирования?
6. Какие существуют формы макетирования?
7. Чем отличаются друг от друга стратегии конструирования ПО?
8. Укажите сходства и различия классического жизненного цикла и инкрементной модели.
9. Объясните достоинства и недостатки инкрементной модели.
10. Чем отличается модель быстрой разработки приложений от инкрементной модели?
11. Объясните достоинства и недостатки модели быстрой разработки приложений.
12. Укажите сходства и различия спиральной модели и классического жизненного цикла.
13. В чем состоит главная особенность спиральной модели?
14. Чем отличается компонентно-ориентированная модель от спиральной модели и классического жизненного цикла?
15. Перечислите достоинства и недостатки компонентно-ориентированной модели.
16. Чем отличаются тяжеловесные процессы от облегченных процессов?
17. Чем отличаются тяжеловесные процессы от прогнозирующих процессов?
18. Чем отличаются подвижные процессы от облегченных процессов?
19. Перечислите достоинства и недостатки тяжеловесных процессов.
20. Перечислите достоинства и недостатки облегченных процессов.
21. Приведите примеры тяжеловесных процессов.
22. Приведите примеры облегченных процессов.

Практическое занятие№1-3. Разработка, описание и анализ информации

**Тема 2. Руководство программным проектом**

Тема 2.1. Процесс руководства проектом Планирование проектных задач

Тема 2.2. Размерно-ориентированные метрики. Функционально-ориентированные метрики.

Тема 2.3. Выполнение оценки в ходе руководства проектом Выполнение оценки проекта на основе LOC- и FP-метрик Конструктивная модель стоимости Предварительная оценка программного проекта Анализ чувствительности программного проекта

**Вопросы для устного опроса:**

1. Что такое мера?
2. Что такое метрика?
3. Что такое выполнение оценки программного проекта?
4. Что такое анализ риска?
5. Что такое трассировка и контроль?
6. Какие размерно-ориентированные метрики вы знаете?
7. Для чего используют размерно-ориентированные метрики?
8. Определите достоинства и недостатки размерно-ориентированных метрик.
9. Определите достоинства и недостатки функционально-ориентированных метрик.
10. Можно ли перейти от FP-оценок к LOC-оценкам?
11. Охарактеризуйте шаги оценки проекта на основе LOC- и FP-метрик. Чем отличается наиболее точный подход от наименее точного?
12. Что такое конструктивная модель стоимости? Для чего она применяется?
13. В чем состоит назначение модели композиции? На каких оценках она базируется?
14. В чем состоит назначение модели раннего этапа проектирования?
15. Охарактеризуйте основное уравнение модели раннего этапа проектирования.
16. Как оцениваются масштабные факторы?

Практическое занятие№4-6. Разработка технического задания к программному обеспечению

### Техническое задание на разработку информационной системы

1. ВВЕДЕНИЕ

Наименование программы «Организация дополнительного образования»

Наименование темы – «Разработка информационной системы организации дополнительного образования»

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Основанием для проведения работ является задание на дипломное проектирование.

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

Разработка информационной системы «Организация дополнительного образования» предназначена для повышения эффективности распространения информационных средств и технологий.

Выполняемые функции:

* 1. . Учет данных об организациях дополнительного образования.
  2. . Учет данных об услугах, которые предоставляют организации.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам ИС «Организация дополнительного образования».

Разработка ИС «Организация дополнительного образования» должно обеспечивать выполнение следующих функций:

4.1.1. При выполнении функции 3.1. система должна поддерживать учет данных об организациях дополнительного образования и предоставлять возможность их редактирования.

4.1.2. При выполнении функции 3.2. система должна поддерживать учет данных об услугах, которые предоставляют организации дополнительного образования, и предоставлять возможность их редактирования.

4.2. Требования к составу и параметрам технических средств.

4.2.1. ИС разрабатывается для IBM PC совместимого персонального компьютера. Компьютер должен отвечать минимальным требованиям установленной на нем операционной системой Windows 7, а также программой MSWord.

4.3. Требования к информационной и программной совместимости

4.3.1. Информационная система разрабатывается с использованием среды разработки VisualStudio 2010 и СУБД SQLServer.

4.3.2 ЭВМ должна работать под управлением операционной системы Windows 7 и выше.

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Работа должна включать в себя следующую документацию:

1. «Техническое задание» (ГОСТ 19.201-78)

2. «Методика испытаний» (ГОСТ 19.301-79)

3. «Руководство пользователя» (ГОСТ 19.505-79)

4. «Руководство программиста» (ГОСТ 19.504-79)

6. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии разработки, этапы и содержание работ указаны в таблице 2.1.

Таблица 2.1

**Стадии и этапы разработки**

|  |  |  |
| --- | --- | --- |
| **Стадии разработки** | **Этапы работ** | **Содержание работ** |
| Техническое задание | Обоснование необходимости разработки модуля | Сбор исходных материалов.  Выбор и обоснование критериев эффективности и качества разрабатываемой программы. Обоснование необходимости проведения научно–исследовательских работ. |
| Научно – исследовательские работы | Определение структуры входных и выходных данных. Определение требований к техническим требованиям. Обоснование принципиальной возможности решения поставленной задачи. Определение требований к программе.  Определение стадий, этапов и сроков разработки программы и документации на нее. Выбор языка программирования. Определение необходимости проведения научно – исследовательских работ на последующих стадиях. Согласование и утверждение технического задания. |

Таблица 2.1 (продолжение)

|  |  |  |
| --- | --- | --- |
|  | Разработка и  утверждение  технического  задания |  |
| Рабочий проект | Разработка модуля | Программирование и отладка программы. |
| Разработка программной документации | Разработка программных документов согласно ГОСТ 19.101-77 |
| Испытание модуля | Разработка, согласование и утверждение программы и методики испытаний.  Проведение предварительных испытаний.  Корректировка программы и программной документации по результатам испытаний. |
| Внедрение | Подготовка и передача модуля | Подготовка и сдача программы и программной документации. Оформление и утверждение акта о передаче программы на сопровождение.  Передача программы в фонд алгоритмов и программ. |

7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Контроль и приемка работы осуществляются согласно документу «Методика испытаний» комиссией в составе руководителя практики (ФИО) и разработчика (ФИО).

**Тема 3. Классические методы анализа**

Тема 3.1. Структурный анализ. Методы анализа, ориентированные на структуры данных

Тема 3.2. Метод анализа Джексона. Методика Джексона

**Вопросы для устного опроса:**

1. Какие задачи решает аппарат анализа?
2. Что такое диаграмма потоков данных?
3. Чем отличается диаграмма потоков данных от блок-схемы алгоритма?
4. Какие элементы диаграммы потоков данных вы знаете?
5. Как формируется иерархия диаграмм потоков данных?
6. Какую задачу решает диаграмма потоков данных высшего (нулевого) уровня? Почему ее называют контекстной моделью?
7. Чем нагружены вершины диаграммы потоков данных?
8. Чем нагружены дуги диаграммы потоков данных?
9. Как организован словарь требований?
10. Какие средства расширения вы знаете?
11. Каковы особенности диаграммы управляющих потоков?
12. Поясните понятие активатора процесса.
13. Поясните понятие условия данных.
14. Поясните понятие управляющей спецификации.
15. Поясните понятие окна управляющей спецификации.
16. Как организована спецификация процесса?
17. Поясните организацию диаграммы переходов-состояний.
18. Какие задачи решают методы анализа, ориентированные на структуры данных?
19. Какие методы анализа, ориентированные на структуры данных, вы знаете?
20. Какие шаги выполняет метод Джексона на этапе анализа?
21. Какие типы структурных диаграмм Джексона вы знаете?
22. Как организовано в методе Джексона обнаружение объектов?
23. Что такое структура объектов Джексона?
24. Как создается структура объектов Джексона?
25. Поясните диаграмму системной спецификации Джексона.
26. Какова задача структурного текста Джексона?

Практическое занятие №7-9. Разработка требований к программному обеспечению

**Тема 4. Основы проектирования программных систем**

Тема 4.1. Особенности процесса синтеза программных систем Особенности этапа проектирования

Тема 4.2. Структурирование системы Моделирование управления Декомпозиция подсистем на модули

Тема 4.3. Модульность Информационная закрытость Связность модуля Сцепление модулей

Тема 4.4. Сложность программной системы Характеристики иерархической структуры программной системы

**Вопросы для устного опроса:**

1. Какова цель синтеза программной системы? Перечислите этапы синтеза.
2. Дайте определение разработки данных, разработки архитектуры и процедурной разработки.
3. Какие особенности имеет этап проектирования?
4. Решение каких задач обеспечивает предварительное проектирование?
5. Какие модели системного структурирования вы знаете?
6. Какие типы моделей управления вы знаете?
7. Какие существуют разновидности моделей централизованного управления?
8. Поясните разновидности моделей событийного управления.
9. Поясните понятия модуля и модульности. Зачем используют модули?
10. В чем состоит принцип информационной закрытости? Какие достоинства он имеет?
11. Что такое связность модуля?
12. Какие существуют типы связности?
13. Дайте характеристику функциональной связности.
14. Дайте характеристику информационной связности.
15. Охарактеризуйте коммуникативную связность.
16. Охарактеризуйте процедурную связность.
17. Дайте характеристику временной связности.
18. Дайте характеристику логической связности.
19. Охарактеризуйте связность по совпадению.
20. Что такое сцепление модуля?
21. Какие существуют типы сцепления?
22. Что значит «улучшать сцепление»?
23. Какие подходы к оценке сложности системы вы знаете?
24. Что определяет иерархическая структура программной системы?

Практическое занятие№ 10. Методология функционального моделирования

**Основы работы в BPwin 4.1**

С помощью этой лабораторной работы Вы сможете:

* познакомиться с основными элементами рабочего интерфейса BPwin 4.1;
* освоить технологию создания новой модели;
* научиться редактировать построенную диаграмму.

**Теоретические сведения**

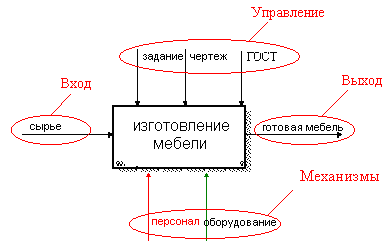
🕮 Для упрощения процесса построения модели бизнес-процессов в IDEF0, можно использовать CASE-средство BPwin, которое позволяет наглядно представить функциональную декомпозицию системы.

**AllFusion Process Modeler 4.1 (BPwin)** является инструментальным средством, полностью поддерживающим стандарт IDEF0, который был принят в начале 90-х годов в США на основе SADT.

Основная идея методологии SADT - построение **древовидной функциональной модели** предприятия.

Сначала функциональность предприятия описывается в целом, без подробностей. Такое описание называется ***контекстной диаграммой*** *(рис. 1.1.)*.

Взаимодействие с окружающим миром описывается в терминах **входа** (данные или объекты, потребляемые или изменяемые функцией), **выхода** (основной результат деятельности функции, конечный продукт), **управления** (стратегии и процедуры, которыми руководствуется функция) и **механизмов** (необходимые ресурсы).



*Рис. 1.1. Пример диаграммы IDEF0*

Кроме того, при создании контекстной диаграммы формулируются *цель моделирования, область* (описания того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и *точка зрения* (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственных за работу моделируемой системы в целом.

Общая функция разбивается на крупные подфункции. Этот процесс называется ***функциональной декомпозицией***.

Затем каждая подфункция декомпозируется на более мелкие - и так далее до достижения необходимой детализации описания. Таким образом, формируется диаграмма IDEF0.

**Этапы построения модели**

1. Определение основного бизнес-процесса.
2. Построение контекстной диаграммы.
3. Построение диаграммы процесса верхнего уровня.
4. Функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм.

**Практическое задание**

**Создание диаграммы А-0**

Построение модели бизнес-процесса рассмотрим на примере работы мебельной фабрики. Во время проведения обследования предприятия были выявлены её целевые задачи, функциональные деятельности каждого из подразделений предприятия и функциональные взаимодействия между ними; информационные потоки внутри подразделений и между ними; внешние по отношению к предприятию объекты и внешние информационные воздействия, а так же нормативно-справочная документация, данные по имеющимся на предприятии средствам и системам автоматизации.

**Целевые функции мебельной фабрики:**

* переработка сырья;
* изготовление деталей для мебели;
* сборка изделия;
* контроль качества.

**Нормативные документы мебельной фабрики:**

* чертежи (деталей, сборочный);
* нормы по переработке сырья;
* стандарты качества;
* производственные инструкции;
* инструкции по технике безопасности.

**Подразделения предприятия:**

* цех по обработке сырья и бракованных изделий;
* цех по изготовлению деталей;
* сборочный цех;
* отдел проверки качества изделия.

Основным сырьем для изготовления мебели является *дерево*.

Определим основной бизнес-процесс, используя выявленные целевые функции. Так как основное предназначение мебельной фабрики состоит в том, чтобы изготавливать мебель, значит, основным бизнес-процессом является ИЗГОТОВЛЕНИЕ МЕБЕЛИ.

Создадим контекстную диаграмму. В данном случае она представляет собой самое общее описание системы. Напомню, что в модели может быть только одна контекстная диаграмма.

Для построения контекстной диаграммы нам необходимо определить входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата; выходную информацию - готовый результат; управление, которое влияет на процесс, но не преобразуется процессом.; механизмы, которые выполняют процесс.

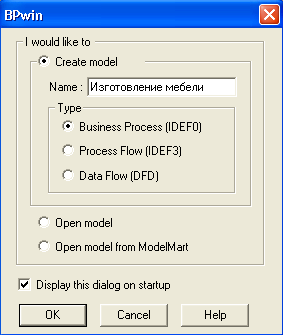
Для контекстного процесса ИЗГОТОВЛЕНИЕ МЕБЕЛИ определим необходимую информацию:

* ВХОД - сырьё;
* УПРАВЛЕНИЕ – чертежи, производственные инструкции, инструкции по технике безопасности (нормативные документы);
* МЕХАНИЗМЫ – персонал, производственное оборудование;
* ВЫХОД – готовая мебель.

 **Задание 1. Запуск программы и открытие проекта**.

**Создание новой модели.**

* 1. Запустите **BPwin**.
  2. Выберите режим работы: создать новую модель (**Create model**).
  3. В поле **Name** введите имя модели: **Изготовление мебели**.
  4. Из группы **Type** выберите тип диаграммы **Business Process** (**IDEF0**) (рис. 1.2.).

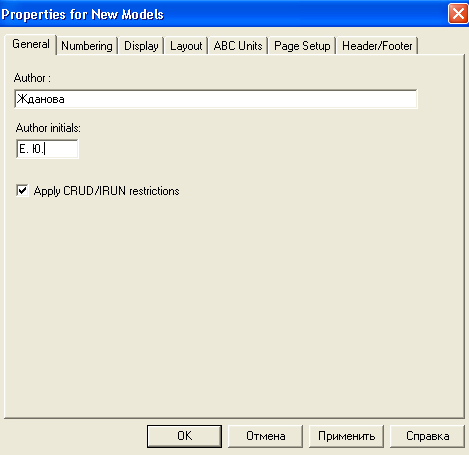
****

*Рис. 1.2. Диалоговое окно создания модели*

* 1. Нажмите OK.

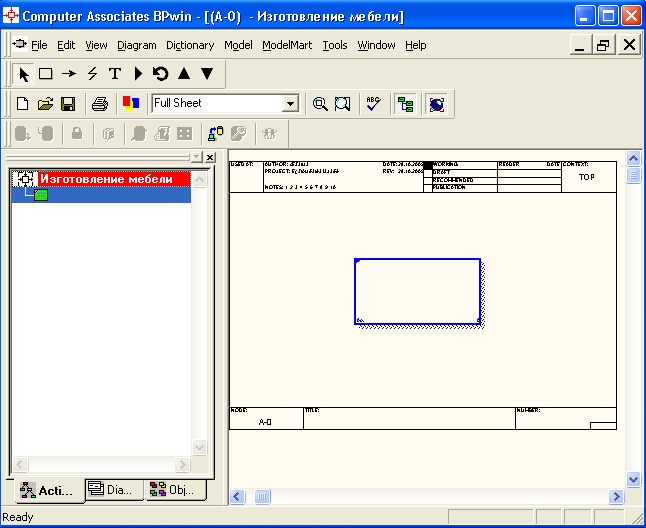
После щелчка на кнопке ОК появляется диалог **Properties for New Models** (рис. 1.3.).

* 1. Укажите свою фамилию и инициалы.

****

*Рис. 1.3. Диалоговое окно Properties for New Models*

После этого вы увидите стандартное окно BPwin (рис. 1.4.):



*Рис. 1.4. Стандартное окно BPwin*

**Основные инструменты BPwin**

Все основные действия с диаграммами, такие как создание, редактирование и т.д., можно выполнить с помощью главного меню, либо контекстно-зависимого меню (меню, появляющееся при нажатии правой кнопке мыши). Принципы работы с меню являются стандартными для среды Windows: объект сначала делается активным, затем над ним осуществляются необходимые действия.

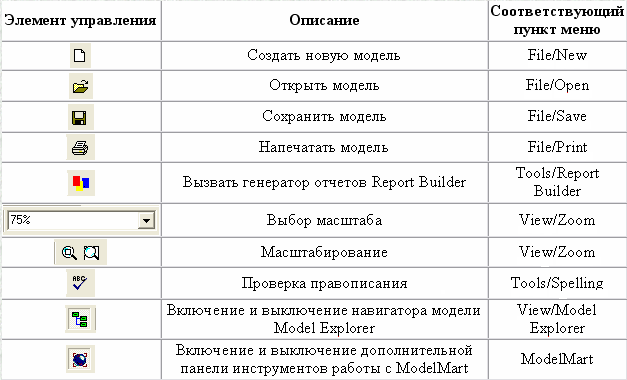
На основной панели инструментов расположены элементы управления, в основном знакомые по другим Windows-интерфейсам (рис. 1.5.):



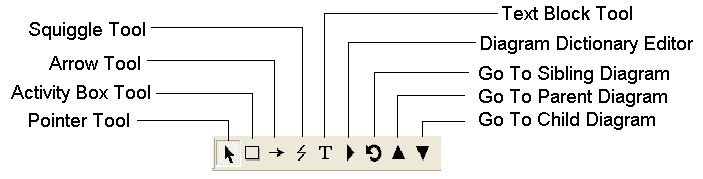
*Рис. 1.5. Элементы управления BPwin*

Функциональность панели инструментов доступна из основного меню BPwin (табл. 1).

**Таблица 1. 1.**  Элементы управления основной панели инструментов BPwin



На основной панели инструментов (либо в любом желаемом месте экрана) расположены инструменты редактора BPwin для IDEF0-диаграмм (рис. 1.6.).



*Рис. 1.6. Панель инструментов редактора BPwin*

**Таблица 1.2.** Инструменты редактора BPwin

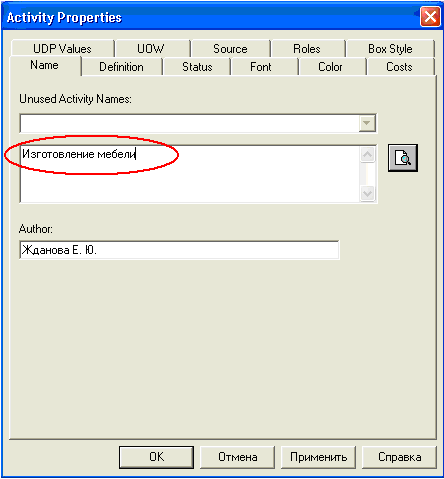


 **Задание 2. Работа с блоком**.

**Построение контекстной диаграммы процесса ИЗГОТОВЛЕНИЕ МЕБЕЛИ.**

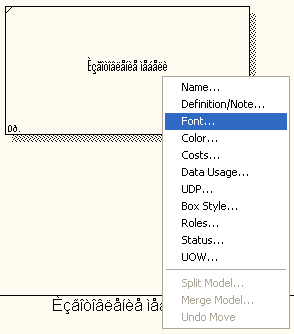
Для ввода имени блока необходимо:

* 1. Щелкнуть правой клавишей мыши по блоку.
  2. Выбрать команду **Name**.
  3. В диалоговом окне ввести название «Изготовление мебели» (рис.1.7.).



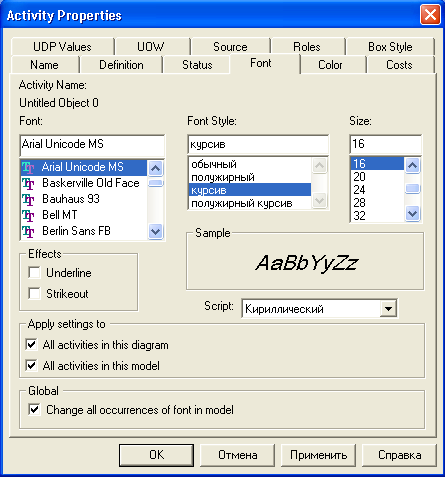
*Рис. 1.7. Диалоговое окно BPwin*

* 1. Для того чтобы текст стал понятен, в контекстном меню выберите пункт **Font** (рис. 1.8.):



*Рис. 1.8. Контекстно-зависимое меню*

* 1. В диалоговом окне **Activity Properties** в нижней части вкладки **Fоnt** установите флажки в опциях **Арр1у setting to**, позволяющих изменить шрифт для всех работ на текущей диаграмме, в модели, и в группе **Global**, позволяющей изменить шрифт одновременно для всех объектов модели, в опции **Script** выберите **«кириллический».**
  2. Установите шрифт **Arial Unicode MS**, курсив, 16 пт (рис. 1.9.).

****

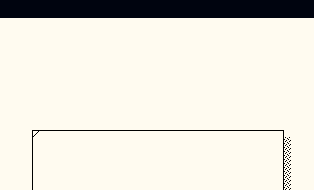
*Рис. 1.9. Вкладка Font диалога Activity Properties*

2После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

 **Задание 3. Построение дуг.**

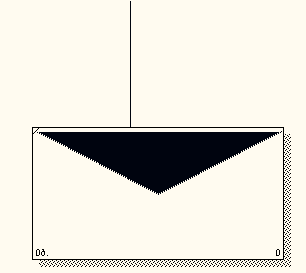
Для построения дуги **управления** необходимо:

* 1. Выбрать на панели инструментов кнопку .
  2. Подвести курсор к верхнему краю окна построения диаграммы до появления черной полосы и произвести щелчок левой кнопкой мыши по этой полосе (рис. 1.10.).



*Рис. 1.10. Начало построения дуги*

* 1. Подвести курсор мыши к верхней стороне блока до образования темного треугольника и щелкнуть левой кнопкой мыши (рис. 1.11.).



*Рис. 1.11. Построение дуги управления*

Построение дуг **входа** и **механизмов** производится аналогичным образом.

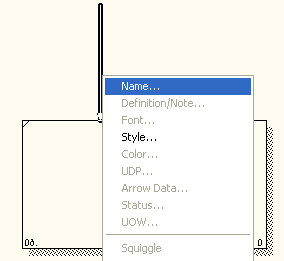
* 1. Постройте дуги **входа** и **механизмов**.

Для построения дуги **выхода** выполняются те же действия, но в противоположном порядке: от правой стороны блока к правой стороне окна построения диаграмм.

* 1. Постройте дугу **выхода.**

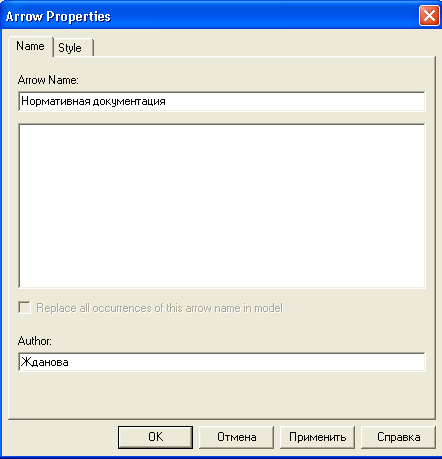
** Задание 4. Идентификация дуги управления.**

* 1. Выберите на панели редактирования кнопку .
  2. Щелкните правой кнопкой мыши по дуге.
  3. Выберите команду **Name** (рис. 1.12.).



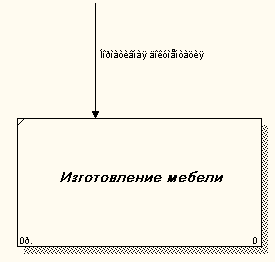
*Рис. 1.12. Контекстно-зависимое меню*

* 1. В диалоговом окне введите название дуги: «Нормативная документация» (рис. 1.13.).



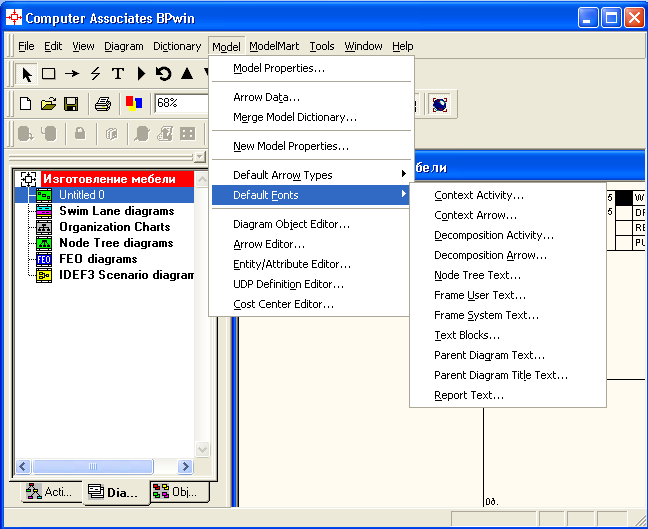
*Рис. 1.13. Диалоговое окно Arrow Properties*

* 1. Для того чтобы текст для дуги стал понятен (рис 1.14.), выберите меню **Моdel - Default Fonts.**

**

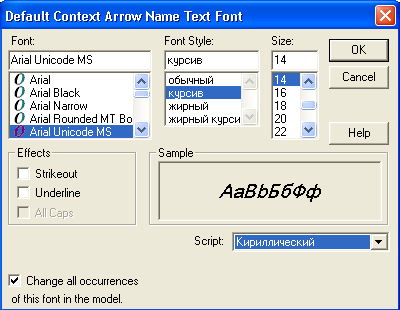
*Рис. 1.14. Фрагмент диаграммы*

ВРwin позволяет установить шрифт по умолчанию для объектов определенного типа (например, дуг) на диаграммах и в отчетах. После выбора меню **Моdel - Default Fonts** появляется каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов (рис. 1.15.):



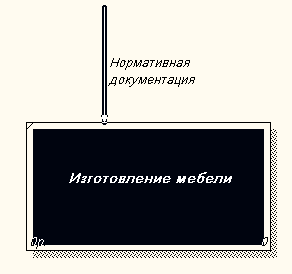
*Рис. 1.15. Диалог Моdel - Default Fonts*

1. **Context Activity** - работа на контекстной диаграмме;
2. **Context Arrow**- стрелки на контекстной диаграмме;
3. **Decomposition Activity** - работы на диаграмме декомпозиции;
4. **Decomposition Arrow** - стрелки на диаграмме декомпозиции;
5. **Node Tree Техt** - текст на диаграмме дерева узлов;
6. **Frame User Техt** - текст, вносимый пользователем в каркасе диаграмм;
7. **Frame System Техt** - системный текст в каркасе диаграмм;
8. **Техt Вlocks** - текстовые блоки;
9. **Parent Diagram Техt** - текст родительской диаграммы;
10. **Parent Diagram Title Техt** - текст заголовка родительской диаграммы;
11. **Report Техt** - текст отчетов.
    1. В диалоговом окне **Default Context Arrow Name Text Font** в нижней части установите флажок в опции **Change all occurrences**, позволяющей изменить шрифт для названий всех дуг на текущей диаграмме, в опции **Script** выберите **«кириллический».**
    2. Установите шрифт **Arial Unicode MS**, курсив, 14 пт. (рис. 1.16.).



*Рис. 1.16. Диалоговое окно Default Context Arrow Name Text Font*

Вот что должно у вас получиться (рис. 1.17.).



*Рис. 1.17. Фрагмент диаграммы*

 **Задание 5. Работа с блоком**.

Самостоятельно постройте дуги:

* ВХОДА: «Сырьё»;
* МЕХАНИЗМА: «Персонал», «Производственное оборудование»;
* ВЫХОДА: «Готовая мебель».

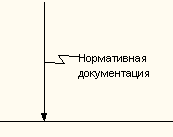
2После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

 **Задание 6. Установление тильды.**

Название дуги является независимым объектом, который можно перемещать относительно дуги. Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом тильды.

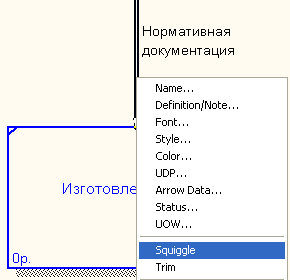
Чтобы установить тильду следует:

* 1. На панели инструментов нажать кнопку ;
  2. Щелкнуть левой кнопкой мыши по тексту, а затем по дуге (рис. 1.18.);



*Рис. 1.18. Установление тильды*

* 1. Можно также использовать команду контекстно-зависимого меню **Squiggle** (рис. 1.19.).



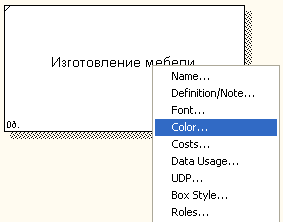
*Рис. 1.19. Контекстно-зависимое меню*

Дуга представляет собой совокупность отельных графических объектов: прямые участки, изогнутые участки, изображение наконечника. Отдельные элементы можно передвигать независимо друг от друга, меняя форму дуги, также дугу можно перемещать как единый неделимый элемент.

* 1. Установите тильду к остальным дугам и их названиям.

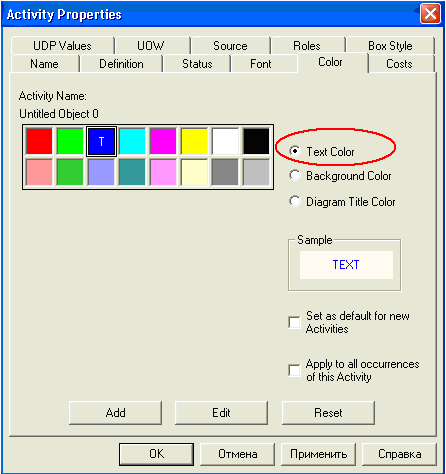
 **Задание 7. Изменение цвета текста, фона блока, цвета и стиля дуг.**

* 1. Для изменения цвета текста выполните команду контекстно-зависимого меню **Color** (рис. 1.20.):



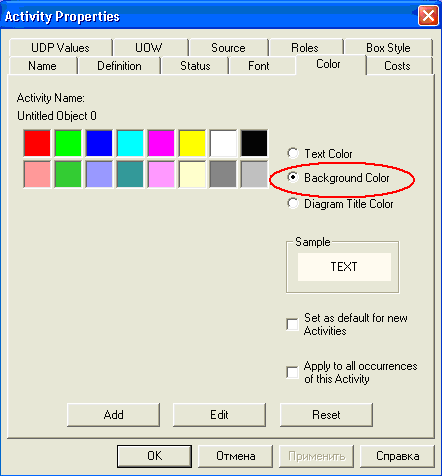
*Рис. 1.20. Контекстно-зависимое меню*

* 1. Выберите цвет и нажмите кнопку  (рис. 1.21.).



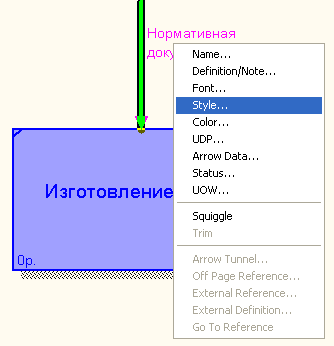
*Рис. 1.21. Диалоговое окно выбора цвета текста и стрелок*

* 1. Для изменения фона блока выберите **Background Color** и цвет (рис. 1.22.):

**

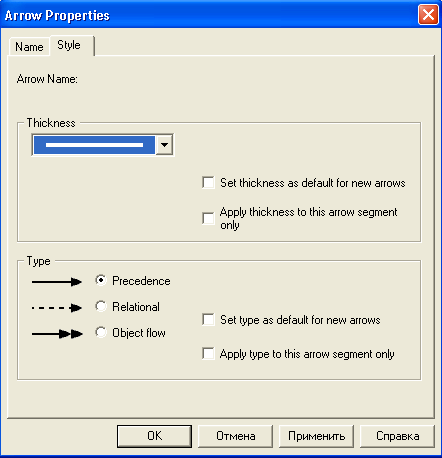
*Рис. 1.22. Вкладка Color диалога Activity Properties*

* 1. Для изменения стиля дуги выберите в контекстно-зависимом меню команду **Style** (рис. 1.23.):



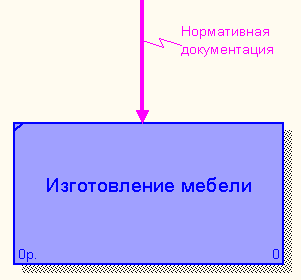
*Рис. 1.23. Контекстно-зависимое меню*

* 1. В диалоговом окне укажите тип и стиль дуги, нажмите на кнопку ОК (рис. 1.24.).



*Рис. 1.24. Диалоговое окно Arrow Properties*

Вот что должно у вас получиться (рис. 1.25.).



*Рис. 1.25. Пример форматирования элементов диаграммы*

**Удаление блоков, дуг или текста.**

Для удаления блока и дуги или текста необходимо их активизировать щелчком левой кнопки мыши и нажать клавишу Delete, а затем подтвердить намерения по поводу удаления.

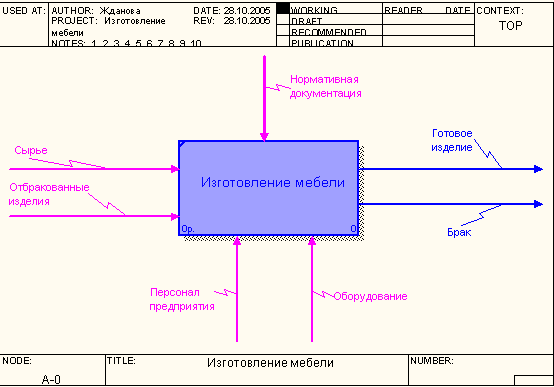
 **Задание 8. Форматирование диаграммы**.

* 1. Самостоятельно произведите форматирование всех элементов диаграммы, опираясь на данные, приведенные в таблице 1.3.

**Таблица 1.3.** Таблицаредактирования

|  |  |  |
| --- | --- | --- |
| **Объект** | **Действие** | **Порядок выполнения** |
| Блок | Изменение размера | Для изменения высоты перетащить мышью верхнюю или нижнюю границу блока, аналогично меняется размер по горизонтали |
| Текст | Стандартные способы форматирования | Правой клавишей мышки щелкнуть по тексту, выбрать пункт контекстно-зависимого меню **Font Editor.** |
| Дуга | Изменение стиля, цвета, размера | Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт: **Style Editor, Color Editor** или **Trim** |
| Удаление блоков, дуг или текста: активизировать щелчком левой кнопки мышки необходимый объект и нажать **Delete**, а затем подтвердить запрос на удаление. | | |

После выполнения задания у вас должна получиться следующая контекстная диаграмма (рис. 1.26.):



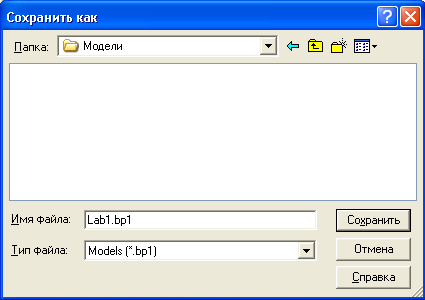
*Рис. 1.26. Контекстная диаграмма процесса «Изготовление мебели»*

*2После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.*

 **Задание 9. Сохранение полученной диаграммы.**

Сохраните полученную диаграмму.

* 1. Создайте папку, назовите её своей фамилией и в неё сохраняйте свои работы.
  2. В меню **File** выберите команду **Save as**.
  3. Укажите путь к своей папке и имя файла **Lab1.bp1** (рис. 1.27.).



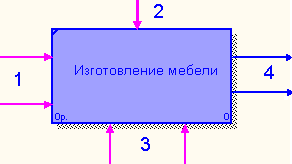
*Рис. 1.27. Диалоговое окно сохранения документа.*

* 1. Нажмите на кнопку СОХРАНИТЬ.

**Контрольные вопросы**

Ответьте на контрольные вопросы:

1. Перечислите основные возможности BPwin.
2. Охарактеризуйте основные элементы рабочего интерфейса BPwin.
3. Какую методологию поддерживает BPwin?
4. Укажите назначение каждой из дуг изображенных на рисунке.



1. Назовите основные этапы построения модели.
2. Какой процесс можно назвать функциональной декомпозицией?
3. Перечислите элементы контекстной диаграммы.
4. При помощи какого инструмента строятся дуги на диаграмме?

* **Тестовое задание**

1. Что представляет собой модель бизнес-процессов:

1. комплекс диаграмм, каждая из которых описывает отдельный бизнес-процесс;
2. иерархию диаграмм, каждая из которых описывает отдельный бизнес-процесс;
3. случайный набор диаграмм, каждая из которых описывает отдельный бизнес-процесс;
4. все ответы правильные;
5. правильного ответа нет.

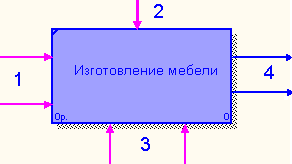
2. Что входит в состав диаграмм:

1. блоки, описывающие подпроцессы (функции системы), и дуги, связывающие блоки вместе и изображающие взаимодействия и взаимосвязи между блоками;
2. блоки, описывающие подпроцессы (функции системы);
3. дуги, связывающие блоки вместе и изображающие взаимодействия и взаимосвязи между блоками;
4. нумерация диаграммы;
5. правильные ответы b), d).

3. Укажите этап, в котором происходит построение диаграммы процесса верхнего уровня:

1. рисование дуги управления;
2. определение основного бизнес-процесса;
3. построение контекстной диаграммы;
4. функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм;
5. все ответы правильные.

4. Укажите, для чего предназначена левая сторона блока (1):



1. для **управления** (правила, стратегии, стандарты);
2. для **механизмов** (ресурсы, которые выполняют процесс);
3. для **выходов** (материал или информация, получаемые в результате выполнения процесса);
4. для **входов** (материал или информация, которые используются или преобразуются для получения результата);
5. правильные ответы c), d).

5. Какая команда из контекстно-зависимого меню позволит изменить стиль дуги:

1. Font Editor;
2. Color Editor;
3. Trim;
4. Style Edition;
5. Background Color;
6. Squiggle.

6. Для чего предназначен инструмент :

1. используется для создания тильды (squiggle), которая соединяет дугу с ее названием;
2. используется для выбора и определения позиции объектов, добавленных в диаграмму;
3. используется для создания текстовых блоков;
4. используется для выбора цвета текста и фона;
5. правильного ответа нет.

**Тема 5. Классические методы проектирования**

Тема 5.1. Метод структурного проектирования

Тема 5.2. Метод проектирования Джексона

**Вопросы для устного опроса:**

1. В чем состоит суть метода структурного проектирования?
2. Какие различают типы информационных потоков?
3. Какие задачи решают главный контроллер, контроллер входящего потока, контроллер выходящего потока и контроллер центра преобразования?
4. Поясните шаги метода структурного проектирования.
5. Какие задачи уточнения иерархической структуры программной системы вы знаете?
6. Какие шаги предусматривает метод Джексона на этапе проектирования?
7. В чем состоит суть развития диаграммы системной спецификации Джексона?
8. В чем состоит учет системного времени (в методе Джексона)?

Практическое занятие№11 Функциональная модель SADT

«Функциональная модель SADT. Методология IDEF0.

Декомпозиция контекстной диаграммы»

С помощью этой работы Вы сможете:

* научиться производить декомпозицию контекстной диаграммы;
* освоить правила построения дуг и тоннелирования стрелок.

**Теоретические сведения**

Сущность структурного подхода к разработке ИС заключается в декомпозиции (разбиении) системы на автоматизируемые функции, которые в свою очередь делятся на подфункции, на задачи и так далее. Процесс декомпозиции продолжается вплоть до определения конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны.

В основе структурного метода лежит несколько общих принципов:

– разбиение системы на множество независимых задач, доступных для понимания и решения;

– иерархическое упорядочивание, т.е. организация составных частей проблемы в древовидные структуры с добавлением новых деталей на каждом уровне.

К основным принципам относятся:

– абстрагирование, т.е. выделение существенных аспектов системы и отвлечение от несущественных;

– формализация, т.е. общее методологическое решение проблемы;

– непротиворечивость, состоящая в обосновании и согласовании элементов системы;

– иерархическая структуризация данных.

1. **Метод функционального моделирования SADT**

На основе метода SADT, предложенного Д. Россом, разработана методология IDEF0 (Icam DEFinition), которая является основной частью программы ICAM (Интеграция компьютерных и промышленных технологий), проводимой по инициативе ВВС США. Методология IDEF0 является наиболее признанным эффективным средством анализа, конструирования и отображения бизнес-процессов, применяемым также и широко за пределами США.

Метод SADT применяется при моделировании широкого круга систем, для которых определяются требования и функции, после чего проводится их реализация.

Методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели предметной области, которая отображает функциональную структуру, производимые функции и действия, а также связи между ними.

Результат применения метода SADT - модель, которая состоит из диаграмм, фрагментов текстов и глоссария со ссылками друг на друга. Все функции и интерфейсы представляются диаграммами в виде, соответственно, блоков и дуг. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, в то время как информация, которая подвергается обработке (исходные данные), указывается с левой стороны блока, а результаты работы функции (выход, результат) - с правой стороны. Механизм, осуществляющий операцию (человек или автоматизированная система), задается дугой, входящей в блок снизу (см. Рис. 1).

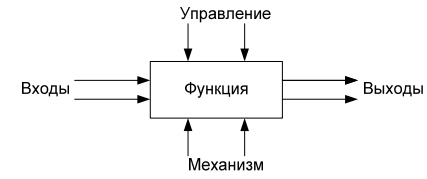


Рис. 1.1. Структура модели

Описание системы с помощью SADT называется моделью. Субъектом моделирования служит сама система. Однако моделируемая система никогда не существует изолированно: она всегда связана с окружающей средой. По этой причине в методологии SADT подчеркивается необходимость точного определения границ системы, т.е. модель устанавливает точно, что является и что не является субъектом моделирования, описывая то, что входит в систему, и, подразумевая то, что лежит за ее пределами. SADT-модель должна иметь единственный субъект.

С определением модели тесно связана позиция (называемая точкой зрения), с которой наблюдается система и создается ее модель. "Точку зрения" лучше всего представлять себе как место (роль, должность) человека или объекта в рассматриваемой системе, на которое надо «встать», чтобы увидеть систему в действии и необходимой полноте. У конкретной модели может быть только одна точка зрения.

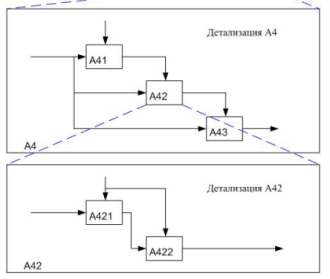
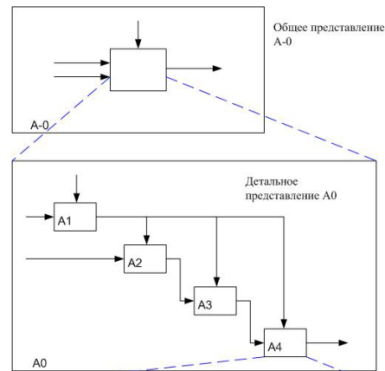


Рис. 1.2. Структура SADT-модели. Иерархия и декомпозиция диаграмм

Обычно вопросы для SADT-модели формулируются на самом раннем этапе проектирования, при этом основная суть этих вопросов должна быть выражена в одной-двух фразах, которые становятся целью модели.

После того как определены субъект, цель и точка зрения модели, начинается первая интеграция процесса моделирования по методологии SADT. Субъект определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации о субъекте и форму ее представления. Цель становится критерием окончания моделирования. Конечным результатом этого процесса является набор тщательно взаимоувязанных описаний, начиная с описания самого верхнего уровня системы и заканчивая подробным описанием ее деталей или отдельных операций.

Каждое из таких тщательно взаимосогласованных описаний называется диаграммой и имеет определенный уровень детализации. SADT-модель объединяет и организует диаграммы в иерархические структуры, в которых диаграммы наверху модели менее детализированы, чем диаграммы нижних уровней. Другими словами, модель SADT можно представить в виде древовидной структуры диаграмм, где верхняя диаграмма является наиболее общей, а самые нижние – максимально детализированы (см. Рис. 2).

Каждый блок на диаграмме имеет свой номер. Блок любой диаграммы может быть детализирован диаграммой нижнего уровня, которая, в свою очередь, также может детализироваться с помощью необходимого числа диаграмм. Таким образом, формируется иерархия диаграмм. Для того чтобы указать положение любой диаграммы или блока в иерархии, им присваивают уникальные обозначения. Например, А41 (A сокр. от Activity) является диаграммой, которая детализирует блок 1 на диаграмме А4. Аналогично, А4 детализирует блок 4 на диаграмме А0, которая является самой верхней (родительской) диаграммой модели.

Некоторые дуги имеют начало в одном из блоков диаграммы и завершение в другом, у других же начало может исходить от границ диаграммы – дуги управления, механизма, дуги входа и выхода, перенесенные с родительской (верхнего уровня) диаграммы. Таким образом, источник или получатель этих пограничных дуг может быть обнаружен только на родительской диаграмме.

Также следует сказать о так называемых «туннельных дугах». Туннельные дуги означают, что данные, выраженные этими дугами не рассматриваются на следующем уровне детализации (как бы проходят «насквозь»). Если «туннель» расположен в месте соединения дуги с блоком « », то данные этой дуги не обязательны на следующем уровне детализации. Если же «туннель» находится на противоположном конце дуги « » - это значит, что данные дуги не описываются на родительской диаграмме. Граничные дуги должны продолжаться (дублироваться) на родительской диаграмме, делая ее полной и непротиворечивой (см. Рис. 3).

Для упрощения понимания приведенных диаграмм, следует расшифровать применяемую в IDEF систему обозначений, позволяющую аналитику точно идентифицировать и проверять по дугам связи между диаграммами. Эта схема кодирования дуг - "ICOM" - получила название по первым буквам английских эквивалентов слов вход (Input), управление (Control), выход (Output), механизм (Mechanism).

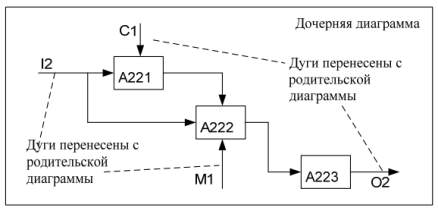
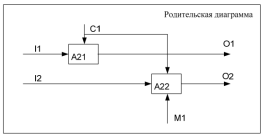


Рис. 1.3. Соответствие дуг родительской и дочерних диаграмм

**2.1. Декомпозиция**

🕮 После создания контекстной диаграммы, которая представляет собой описание контекста моделируемой системы, проводится **функциональная декомпозиция** – система разбивается на подсистемы и каждая подсистема описывается в том же синтаксисе, что и система в целом. Затем каждая подсистема разбивается на более мелкие и так до достижения нужного уровня подробности. В результате такого разбиения, каждый фрагмент системы изображается на отдельной диаграмме декомпозиции. *Диаграмма декомпозиции предназначена для детализации работы.*

При декомпозиции процесса все стрелки, входящие или исходящие из него, должны быть перенесены на диаграмму нижнего уровня и использованы при ее построении. При этом запрещены всякие новые стрелки, выходящие за пределы новой диаграммы, кроме специальных, так называемых "**тоннелированных**" стрелок.

**Создание диаграммы А0**

*Диаграмма верхнего уровня* создается путем декомпозиции основной функции контекстной диаграммы. На диаграмме декомпозиции функции нумеруются автоматически слева направо. Номер функции показывается в правом нижнем углу. В левом верхнем исчезает небольшая диагональная черта, которая показывает, что данная функция была декомпозирована.

**Практическое задание**

 **Задание 10. Детализация процесса «Изготовление мебели».**

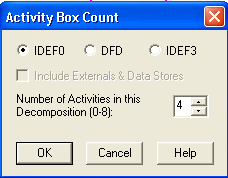
Откройте файл **Lab1.bp1**, сохраненный на предыдущем уроке.

Следующим шагом является детализация контекстного процесса с помощью диаграммы верхнего уровня. Эта диаграмма содержит в себе четыре процесса:

1. Процесс 1.1 – ПЕРЕРАБОТКА СЫРЬЯ.
2. Процесс 1.2 – ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.
3. Процесс 1.3 – СБОРКА ИЗДЕЛИЯ.
4. Процесс 1.4 – КОНТРОЛЬ КАЧЕСТВА.

Произведите детализацию процесса «Изготовление мебели», задав нужное количество новых блоков. Для этого:

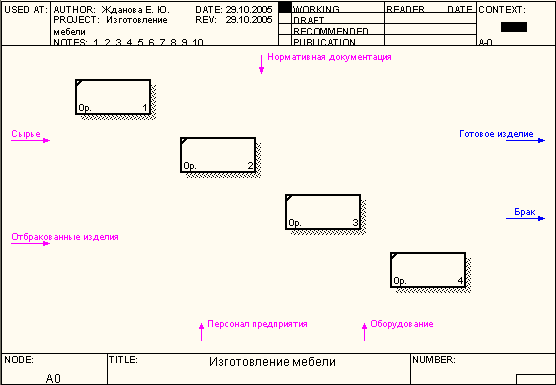
1. Щелкните по блоку «Изготовление мебели» и выберите инструмент .
2. В диалоговом окне введите число, на которое будет произведена декомпозиция – 4.
3. Укажите тип диаграммы **IDEF0** (рис. 2.1.) и нажмите OK.



*Рис. 2.1. Диалоговое окно декомпозиции блока*

1. Укажите названия новых блоков («Переработка сырья», «Изготовление деталей», «Сборка изделия», «Контроль качества»).

При декомпозиции функции входящие в нее и исходящие из нее дуги автоматически появляются на диаграмме декомпозиции (миграция дуг), но при этом не касаются блоков. Такие стрелки называются ***несвязанными*** и воспринимаются в BPwin как синтаксическая ошибка (см. рис.2.2.).



*Рис. 2.2.*  *Декомпозиция верхнего уровня*

Определим входные и выходные потоки для новых процессов.

**Процесс 1.1. ПЕРЕРАБОТКА СЫРЬЯ:**

1. Вход – СЫРЬЁ.
2. Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ.
3. Выход – ЗАГОТОВКИ.

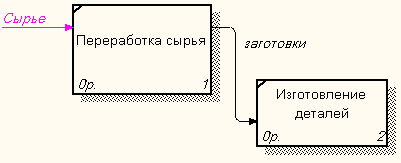
Произведем процесс связывания мигрирующих дуг:

1. Выберите инструмент  рисования дуг.
2. Щелкните мышью по наконечнику входного потока СЫРЬЁ.
3. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.

Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ построим немного позже.

Для построения выходного потока ЗАГОТОВКИ выполните действия:

1. Выберите инструмент  рисования дуг.
2. Щелкните левой кнопкой мышки по выходной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
3. Затем щелкните по входной стороне блока ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.
4. Выберите инструмент  текст, в контекстном меню – команду **Name,** укажите название дугиЗАГОТОВКИ.
5. Проверьте себя (рис. 2.3.).



*Рис. 2.3. Фрагмент диаграммы*

 **Задание 11. Детализация процесса «Изготовление мебели».**

1. Самостоятельно выполните детализацию процессов:

**Процесс 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ:**

1. Вход – ЗАГОТОВКИ.
2. Выход – ГОТОВЫЕ ДЕТАЛИ.

**Процесс 1.3. СБОРКА ИЗДЕЛИЯ:**

1. Вход – ГОТОВЫЕ ДЕТАЛИ.
2. Выход – СОБРАННОЕ ИЗДЕЛИЕ.

**Процесс 1.4. КОНТРОЛЬ КАЧЕСТВА:**

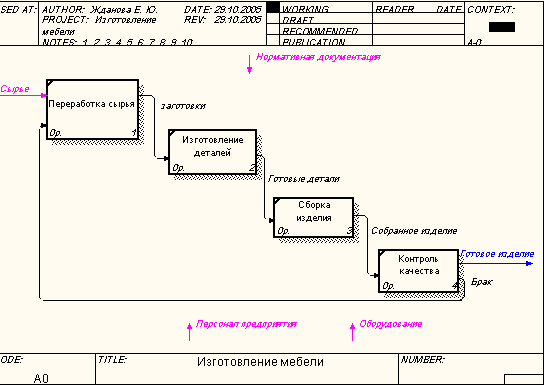
1. Вход – СОБРАННОЕ ИЗДЕЛИЕ.
2. Выход – ГОТОВОЕ ИЗДЕЛИЕ.
3. Выход – БРАК.
4. Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ

*2После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.*

 **Задание 12. Смена направления дуги.**

На Выходе БРАК не выходит за границу модели, а возвращается в процесс ПЕРЕРАБОТКА СЫРЬЯ:

1. Удалите дуги ОТБРАКОВАННЫЕ ИЗДЕЛИЯ и БРАК.
2. Выберите инструмент  рисование дуг.
3. Щелкните левой кнопкой мыши на Выходе блока КОНТРОЛЬ КАЧЕСТВА.
4. Щелкните левой кнопкой на Входе блока ПЕРЕРАБОТКА СЫРЬЯ.
5. Назовите новую дугу – БРАК (рис. 2.4.).

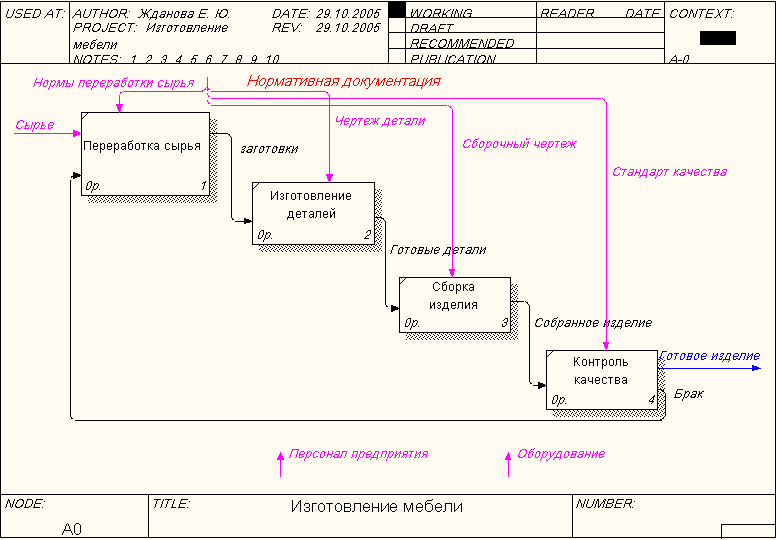


*Рис. 2.4. Процесс декомпозиции*

 **Задание 13. Построение ответвлений дуг.**

Переработка сырья, изготовление деталей, сборка изделия, контроль качества осуществляются согласно Нормативным документам, поэтому у управляющей стрелки НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ появятся ответвления: НОРМЫ ПЕРЕРАБОТКИ СЫРЬЯ, ЧЕРТЕЖ ДЕТАЛИ, СБОРОЧНЫЙ ЧЕРТЕЖ, СТАНДАРТ КАЧЕСТВА.

1. Выберите инструмент  рисование дуг.
2. Щелкните мышью по наконечнику входного потока НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ.
3. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
4. Самостоятельно выполните ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ на блоки ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ, КОНТРОЛЬ КАЧЕСТВА.
5. Проверьте себя(рис. 2.5.).

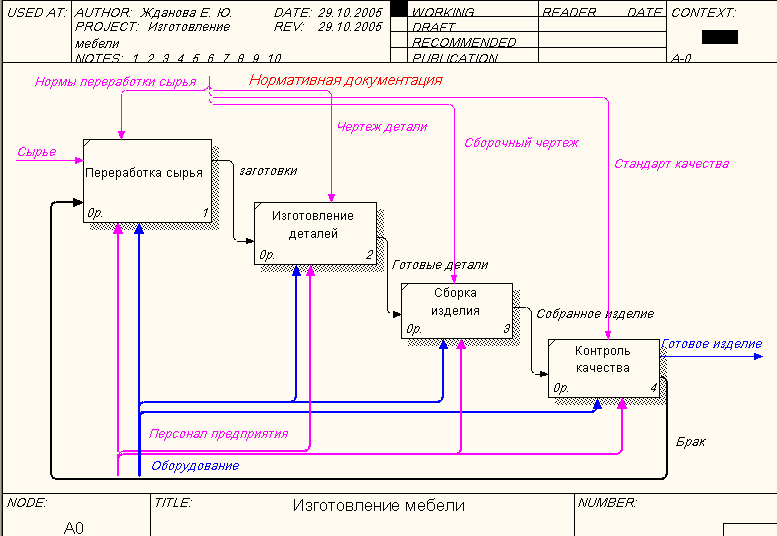


*Рис. 2.5. Ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ*

 **Задание 14. Построение дуг Персонал предприятия, Оборудование.**

Дуги ПЕРСОНАЛ ПРЕДПРИЯТИЯ и ОБОРУДОВАНИЕ для всех процессов будут одинаковые.

1. Самостоятельно соедините каждую дугу с каждым блоком, укажите ее имя.
2. Проверьте себя (рис. 2.6.).



*Рис. 2.6. Построение дуг ПЕРСОНАЛ ПРЕДПРИЯТИЯ и ОБОРУДОВАНИЕ*

*2После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.*

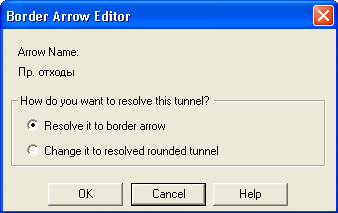
 **Задание 15. «Тоннелирование» стрелок.**

1. В Процессе 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ постройте новую граничную дугу, которой обозначьте Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ.

Вновь внесенные граничные дуги на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня.

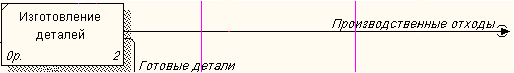
Для их «перетаскивания» наверх нужно:

1. Выбрать инструмент  редактирования.
2. Кликнуть правой кнопкой мыши по квадратным скобкам.
3. Выбрать в контекстном меню пункт **Arrow Tunnel**.
4. В появившемся диалоге **Border Arrow Editor** (рис. 2.7.) щелкнуть по кнопке **Resolve it to border arrow** для миграции стрелки на диаграмму верхнего уровня или по кнопке **Change it to resovled rounded tunnel** для «тоннелирования» дуги.



*Рис. 2.7. Диалог Border Arrow Editor*

Тоннельная дуга изображается с круглыми скобками на конце и не попадет на другую диаграмму (рис. 2.8.). Такое тоннелирование может быть применено для изображения малозначимых стрелок.



*Рис. 2.8. Граничная дуга*

1. Отправьте созданную дугу "Производственные отходы" в тоннель.

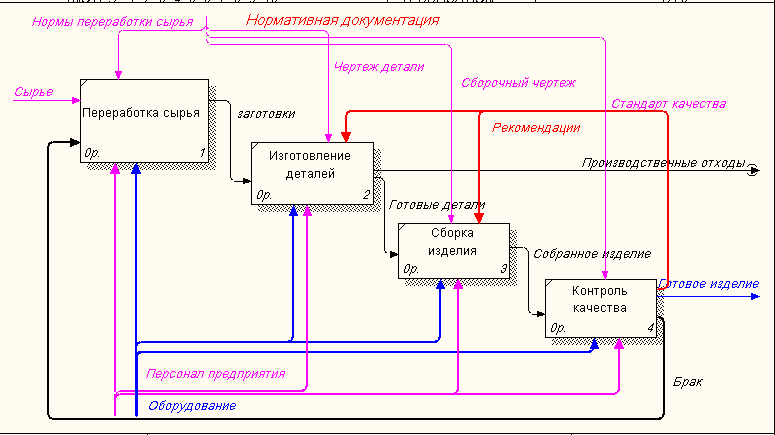
 **Задание 16. Создание обратной связи по управлению.**

Качество изделия может быть повышено путем непосредственного регулирования процессами изготовления деталей и сборки мебели в зависимости от результата (выхода) работы КОНТРОЛЬ КАЧЕСТВА.

Обратная связь по управлению свидетельствует об эффективности бизнес-процесса и создается следующим образом:

1. Выберите инструмент  рисование дуг.
2. Щелкните мышью по выходу КОНТРОЛЬ КАЧЕСТВА.
3. Щелкните по управлению блоков ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ.
4. Выберите инструмент  текст.
5. Назовите обратную связь РЕКОМЕНДАЦИИ.

После выполнения работы у вас должна получиться следующая диаграмма (рис. 2.9.):



Рис*. 2.9. Диаграмма декомпозиции блока ИЗГОТОВЛЕНИЕ МЕБЕЛИ*

 **Задание 17. Сохранение полученной диаграммы.**

1. В меню **File** выберите **Save as.**
2. Укажите путь к своей папке и имя файла **Lab2.bp1**.
3. Нажмите ОК.

**Контрольные вопросы**

1. Как создается диаграмма верхнего уровня?
2. Как на диаграмме отображается декомпозиция?
3. Каким инструментом задается количество блоков для декомпозиции?
4. Какие стрелки BPwin воспринимает как синтаксическую ошибку?
5. Для чего создается обратная связь по управлению?
6. Для чего служит тоннелирование дуг?

**Тестовое задание**

1. **В каком порядке надо располагать блоки на диаграмме детализации:**
   1. в произвольном;
   2. обязательно справа налево;
   3. в соответствии с уровнем сложности последующей детализации;
   4. в соответствии с их доминированием;
   5. все ответы правильные.
2. **Для чего предназначена диаграмма детализации процесса:**
3. для обозначения тоннелированных дуг;
4. для описания контекста моделируемой системы;
5. для детализации работы;
6. все ответы правильные.
7. **Как создается диаграмма верхнего уровня:**
   1. путем декомпозиции одной из функций контекстной диаграммы;
   2. путем декомпозиции основной функции контекстной диаграммы;
   3. путем декомпозиции диаграмм нижнего уровня;
   4. путем декомпозиции одной из функций на диаграмме верхнего уровня;
   5. правильного ответа нет.
8. **Какой инструмент необходимо выбрать, чтобы с его помощью произвести декомпозицию блока?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **2.** | **3.** | **4.** | **5.** |

1. **Каким инструментом производится процесс связывания мигрирующих дуг?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **2.** | **3.** | **4.** | **5.** |

1. **Мигрирующая дуга – это дуга, которая…**
2. создает обратную связь по управлению;
3. переходит из родительской диаграммы и не касается блока;
4. обязательно тоннелируется;
5. создает обратную связь по входу;
6. правильного ответа нет.

Практическое занятие№12-13 Моделирование IDF3

Практическое занятие№14 Диаграммы потока данных DFD

«Диаграммы потоков данных DFD**»**

С помощью этой работы Вы сможете:

* освоить назначение и принципы построения DFD - диаграммы потоков данных;
* научиться устанавливать внутренние ссылки;
* научиться производить детализацию диаграммы DFD.

**Теоретические сведения**

**Диаграммы DFD**

🕮 **Диаграммы потоков данных** (Data flow diagramming, **DFD**) можно использовать как дополнение к модели **IDEF0** для более *наглядного отображения* текущих операций документооборота в системах обработки информации.

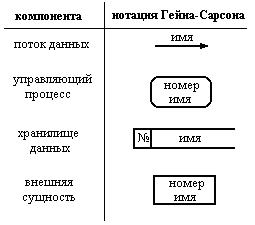
***Диаграммы потоков данных*** используются для описания документооборота и обработки информации и представляют модельную систему как ***сеть связанных между собой работ***.

Диаграммы потоков данных **(DFD)** показывают *внешние источники и приемники данных, потоки данных и хранилища (накопители) данных*, к которым осуществляется доступ.

DFD описывает:

1. функции обработки информации (работы);
2. документы (стрелки*, arrows*), объекты, сотрудников или отделы, которые участвуют в обработке информации;
3. внешние ссылки (*external references*), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
4. таблицы для хранения документов (хранилища данных*, data store*).

Для изображения диаграмм потоков данных в **BPwin** используется нотация **Гейна-Сарсона** (рис.1.).



*Рис. 1. Нотация Гейна-Сарсона*

**Этапы построения модели**

1. Выделение внешних объектов, с которыми система должна быть связана.
2. Формирование DFD диаграммы первого уровня.
3. Функциональная декомпозиция каждого подпроцесса с помощью диаграмм нижнего уровня.
4. Составление словаря данных.
5. Построение спецификаций процесса, если его нельзя выразить комбинацией подпроцессов.

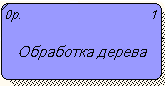
При дополнении модели IDEF0 диаграммой DFD, в палитре инструментов на новой диаграмме **DFD** появляются новые кнопки:

 - добавить в диаграмму внешнюю ссылку (*External Reference*). Внешняя ссылка является источником или приемником данных извне модели.

 - добавить в диаграмму хранилище данных (*Data store*). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах.

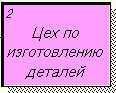
В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов.

**Работы** в DFD представляют собой функции системы, преобразующие *входы в выходы*. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ в IDEF0, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0 (рис. 2.).



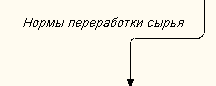
*Рис. 2. Изображение работы*

**Внешние сущности** изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы (рис. 3.). Одна внешняя сущность может быть использована многократно на одной или не­скольких диаграммах. Обычно такой прием применяют, чтобы не рисовать слишком длинных и запутанных стрелок.



*Рис. 3. Внешняя сущность*

**Стрелки (потоки данных)** описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника работы. В DFD также приме­няются двунаправленные стрелки для описания диалогов типа команды-ответа между работами, между работой и внешней сущностью и между внешними сущностями (рис. 4.).



*Рис. 4. Потоки данных (стрелки)*

**Хранилище данных** изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов (рис. 5.).



*Рис. 5. Хранилище данных*

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более похожими на физические характеристики системы - движение объектов (*data flow*), хранение объектов (*data stores*), поставка и распространение объектов (*external entities*) (рис. 6.).

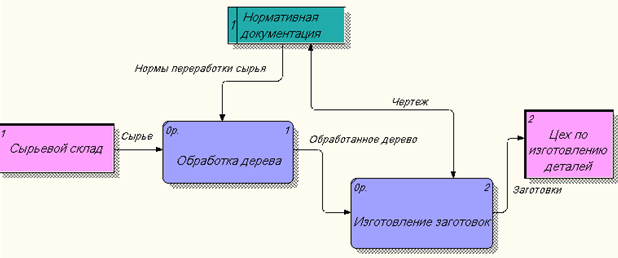


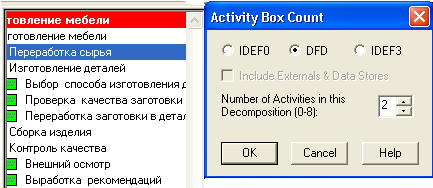
Рис. 6. Пример диаграммы DFD

**Практическое задание**

Построение модели рассмотрим на примере бизнес-процесса "Переработка сырья".

 **Задание 26. Создание диаграммы DFD.**

1. Откройте файл **Lab4.bp1**, сохраненный на предыдущем уроке.
2. Выберите инструмент  и в процессе декомпозиции работы «Переработка сырья» в диалоге **Activity Box Count** «кликните» по радио-кнопке **DFD.**
3. В диалоге **Activity Box Count** выберите количество работ – 2 (рис. 7.).



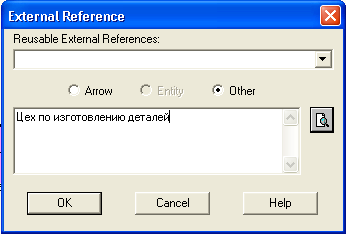
*Рис. 7. Диалог Activity Box Count*

1. Щелкните по кнопке ОК и внесите в новую диаграмму имена работ: «Обработка дерева», «Изготовление заготовок».

 **Задание 27. Изображение внешних сущностей.**

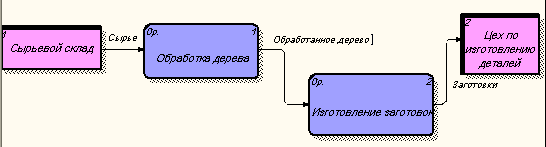
Выполните следующие действия:

1. Используя кнопку , внесите внешние сущности (внешние ссылки).
2. В появившемся диалоговом окне введите название сущностей: «Сырьевой склад», «Цех по изготовлению деталей» (рис. 8.) и нажмите ОК.



*Рис. 8. Диалоговое окно внешней сущности*

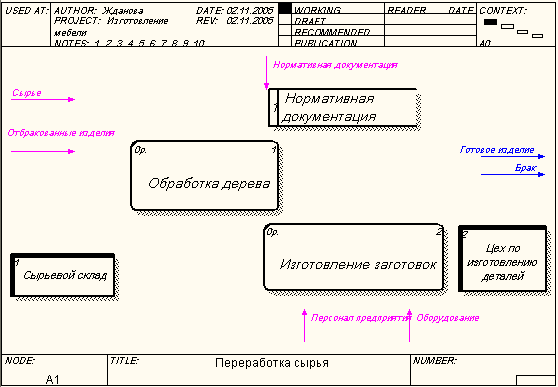
После выполнения задания у вас должна получиться следующая диаграмма (рис. 9.):

**

*Рис. 9. Изображение внешних сущностей*

 **Задание 28. Построение хранилищ.**

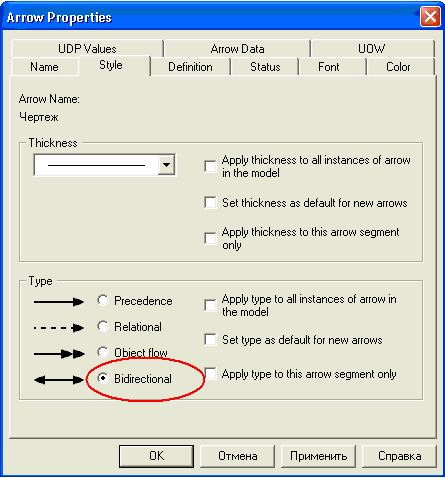
1. Используя кнопку  на палитре инструментов, внесите хранилище данных: «Нормативная документация» (рис. 10.).
2. Удалите граничные стрелки с диаграммы;



*Рис. 10. Диаграмма DFD*

 **Задание 29. Создание внутренних ссылок.**

1. Используя инструмент , создайте внутренние ссылки: «Сырье», «Обработанное дерево»; «Заготовки», «Нормы переработки сырья», «Чертеж».
2. Стрелку «Чертеж» необходимо сделать двунаправленной, для этого щелкните правой кнопкой по стрелке, выберите в контекстном меню пункт **Style**.
3. Выберите в диалоговом окне **Arrow Properties** опцию **Bidirectional** (рис 11.).

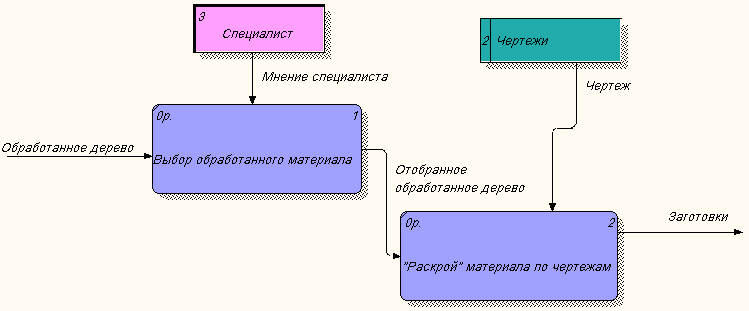


*Рис. 11. Диалоговое окно выбора стиля стрелки*

1. У вас должна получиться диаграмма (рис. 6.).

 **Задание 30. Детализация процесса «Изготовление заготовок».**

Самостоятельно проведите детализацию процесса "Изготовление заготовок" в соответствии с рисунком (рис. 12.).



*Рис. 12. Детализация процесса «Изготовление заготовок»*

 **Задание 31. Сохранение полученной диаграммы.**

Сохраните полученную диаграмму.

1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab3.bp1**.
3. Нажмите ОК.

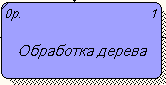
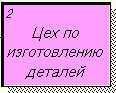
**Контрольные вопросы**

1. Для чего служит DFD - диаграмма?
2. В чем отличие DFD - диаграммы от IDFE0?
3. Какой инструмент используется для построения внешних сущностей?
4. Каким инструментом можно построить Хранилище данных?
5. Какие графические элементы используются для обозначения на диаграмме Работы, Потоков данных, Хранилищ данных?

****Тестовое задание**

1. **DFD - это:**
2. диаграмма бизнес - процесса;
3. диаграмма потока данных;
4. контекстная диаграмма;
5. функциональная диаграмма;
6. диаграмма сущность - связь.
7. **Какие объекты описываются на диаграмме потоков данных:**
8. функции (работы);
9. хранилища данных;
10. внешние объекты;
11. внешние ссылки;
12. все ответы правильные.
13. **Что показывают диаграммы потоков данных?**
14. внешние источники и приемники данных;
15. потоки данных;
16. хранилища (накопители) данных;
17. все ответы правильные;
18. правильного ответа нет.
19. **Какой инструмент нужно выбрать для создания диаграммы DFD?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1.** | **2.** | **3.** | **4.** | **5.** |

1. **Укажите номер рисунка, обозначающий на диаграмме Хранилище данных.**
2.  b)  c) 
3. **Каким инструментом можно обозначить в диаграмме DFD Внешнюю сущность?**

*a) ; b) ; c) ; d)* ; e) *.*

**Тема 6. Структурное тестирование программного обеспечения**

Тема 6.1. Основные понятия и принципы тестирования ПО Особенности тестирования «белого ящика».

Тема 6.2. Способ тестирования базового пути. Способы тестирования условий

Тема 6.3. Способ тестирования потоков данных. Тестирование циклов

**Вопросы для устного опроса:**

1. Определите понятие тестирования.
2. Что такое тест? Поясните содержание процесса тестирования.
3. Что такое исчерпывающее тестирование?
4. Какие задачи решает тестирование?
5. Каких задач не решает тестирование?
6. В чем состоит суть тестирования «черного ящика»?
7. В чем состоит суть тестирования «белого ящика»?
8. Каковы особенности тестирования «белого ящика»?
9. Какие недостатки имеет тестирование «белого ящика»?
10. Какие достоинства имеет тестирование «белого ящика»?
11. Дайте характеристику способа тестирования базового пути.
12. Поясните шаги способа тестирования базового пути.
13. Поясните достоинства, недостатки и область применения способа тестирования базового пути.
14. Дайте общую характеристику способов тестирования условий.
15. Какие методики тестирования условий вы знаете?
16. Поясните суть способа тестирования ветвей и операторов отношений. Какие он имеет ограничения?
17. Поясните шаги способа тестирования ветвей и операторов отношений.
18. Поясните достоинства, недостатки и область применения способа тестирования ветвей и операторов отношений.
19. Поясните суть способа тестирования потоков данных.
20. Поясните шаги способа тестирования потоков данных.
21. Поясните достоинства, недостатки и область применения способа тестирования потоков данных.
22. Поясните особенности тестирования циклов.
23. Какие методики тестирования простых циклов вы знаете?
24. Каковы шаги тестирования вложенных циклов?

Практическое занятие№15 Тестирование «Белого ящика»

**Тема 7. Функциональное тестирование программного обеспечения**

Тема 7.1. Особенности тестирования «черного ящика». Способ разбиения по эквивалентности.

Тема 7.2. Способ анализа граничных значений

Тема 7.3. Способ диаграмм причин-следствий

**Вопросы для устного опроса:**

1. Каковы особенности тестирования методом «черного ящика»?
2. Какие категории ошибок выявляет тестирование методом «черного ящика»?
3. Какие достоинства имеет тестирование методом «черного ящика»?
4. Поясните суть способа разбиения по эквивалентности.
5. Что такое класс эквивалентности?
6. Какие правила формирования классов эквивалентности вы знаете?
7. Как выбирается тестовый вариант при тестировании по способу разбиения по эквивалентности?
8. Поясните суть способа анализа граничных значений.
9. Чем способ анализа граничных значений отличается от разбиения по эквивалентности?
10. В чем суть способа диаграмм причин-следствий?
11. Дайте общую характеристику графа причинно-следственных связей.
12. Поясните шаги способа диаграмм причин-следствий.
13. Какую структуру имеет таблица решений в способе диаграмм причин-следствий?
14. Как таблица решений преобразуется в тестовые варианты?

Практическое занятие№16 Тестирование «Черного ящика»

**Тема 8. Организация процесса тестирования программного обеспечения**

Тема 8.1. Методика тестирования программных систем. Тестирование элементов.

Тема 8.2.Тестирование интеграции. Тестирование правильности. Системное тестирование. Искусство отладки

**Вопросы для устного опроса:**

1. Поясните суть методики тестирования программной системы.
2. Когда и зачем выполняется тестирование элементов? Какой этап конструирования оно проверяет?
3. Когда и зачем выполняется тестирование интеграции? Какой этап конструирования оно проверяет?
4. Когда и зачем выполняется тестирование правильности? Какой этап конструирования оно проверяет?
5. Когда и зачем выполняется системное тестирование? Какой этап конструирования оно проверяет?
6. Поясните суть тестирования элементов.
7. Что такое драйвер тестирования?
8. Что такое заглушка?
9. В чем цель тестирования интеграции?
10. В чем суть нисходящего тестирования интеграции?
11. В чем суть восходящего тестирования интеграции?
12. Что такое отладка?
13. Какие способы проявления ошибок вы знаете?
14. Какие симптомы ошибки вы знаете?
15. В чем суть аналитических методов отладки?
16. Поясните достоинства и недостатки аналитических методов отладки.
17. В чем суть экспериментальных методов отладки?
18. Поясните достоинства и недостатки экспериментальных методов отладки.

Практическое занятие№17 NUnit тестирование

**Тема 9. Основы объектно- ориентированного представления программных систем**

Тема 9.1. Принципы объектно- ориентированного представления программных систем.

Тема 9.2. Объекты.

Тема 9.3. Классы

**Вопросы для устного опроса:**

1. В чем отличие алгоритмической декомпозиции от объектно-ориентированной декомпозиции сложной системы?
2. В чем особенность объектно-ориентированного абстрагирования?
3. В чем особенность объектно-ориентированной инкапсуляции?
4. Каковы средства обеспечения объектно-ориентированной модульности?
5. Дайте общую характеристику объектов.
6. Чем отличается объект от класса?
7. Какие формы видимости между объектами вы знаете?
8. Охарактеризуйте отношение агрегации между объектами. Какие разновидности агрегации вы знаете?
9. Дайте общую характеристику класса.
10. Поясните внутреннее и внешнее представление класса.
11. Какие вы знаете секции в интерфейсной части класса?
12. Какие виды отношений между классами вы знаете?
13. Поясните ассоциации между классами.
14. Поясните наследование классов.
15. Поясните понятие полиморфизма.
16. Поясните отношения агрегации между классами.

**Тема 10. Базис языка визуального моделирования**

Тема 10.1. Унифицированный язык моделирования

Тема 10.2. Предметы в UML. Отношения в UML

Тема 10.3. Диаграммы в UML. Механизмы расширения в UML

**Вопросы для устного опроса:**

1. Сколько поколений языков визуального моделирования вы знаете?
2. Назовите численность языков визуального моделирования 2-го поколения.
3. Какая необходимость привела к созданию языка визуального моделирования третьего поколения?
4. Поясните назначение UML.
5. Какие строительные блоки образуют словарь UML? Охарактеризуйте их.
6. Какие разновидности предметов UML вы знаете? Их назначение?
7. Перечислите известные вам разновидности структурных предметов UML.
8. Какие разновидности отношений предусмотрены в UML? Охарактеризуйте каждое отношение.
9. Дайте характеристику диаграммы классов.
10. Дайте характеристику диаграммы объектов.
11. Охарактеризуйте диаграмму Use Case.
12. Охарактеризуйте диаграммы взаимодействия.
13. Дайте характеристику диаграммы последовательности.
14. Дайте характеристику диаграммы сотрудничества.
15. Охарактеризуйте диаграмму схем состояний.
16. Охарактеризуйте диаграмму деятельности.
17. Дайте характеристику компонентной диаграммы.
18. Охарактеризуйте диаграмму размещения.
19. Для чего служат механизмы расширения в UML?
20. Поясните механизм ограничений в UML.
21. Объясните механизм теговых величин в UML.
22. В чем суть механизма стереотипов UML?

Практическое занятие№18 Методология объектно-ориентированного программирования

Практическое занятие№19 Методология объектно-ориентированного программирования

Практическое занятие№20 Методология объектно-ориентированного программирования

**Тема 11.Статические модели объектно-ориентированных программных систем**

Тема 11.1. Вершины в диаграммах классов Отношения в диаграммах классов

Тема 11.2. Деревья наследования Примеры диаграмм классов

**Вопросы для устного опроса:**

1.Поясните назначение статических моделей объектно-ориентированных программных систем.

2. Что является основным средством для представления статических моделей?

3. Как используются статические модели?

4. Какие секции входят в графическое обозначение класса?

5. Какие секции класса можно не показывать?

6. Какие имеются разновидности области действия свойства (операции)?

7. Поясните общий синтаксис представления свойства.

8. Какие уровни видимости вы знаете? Их смысл?

9. Какие характеристики свойств вам известны?

10. Поясните общий синтаксис представления операции.

11. Какой вид имеет форма представления параметра операции?

12. Какие характеристики операций вам известны?

13. Что означают три точки в списке свойств (операций)?

14. Как организуется группировка свойств (операций)?

15. Как ограничить количество экземпляров класса?

16. Перечислите известные вам «украшения» отношения ассоциации.

17. Может ли статическая модель программной системы не иметь отношений ассоциации?

18. Какой смысл имеет квалификатор? К чему он относится?

19. Какие отношения могут иметь пометки видимости и что эти пометки обозначают?

20. Какой смысл имеет класс-ассоциация?

21. Чем отличается агрегация от композиции? Разновидностями какого отношения (в UML) они являются?

22. Что обозначает в UML простая зависимость?

23. Какой смысл имеет отношение обобщения?

Практическое занятие№21 Диаграмма классов

**Тема 12. Динамические модели объектно- ориентированных программных систем**

Тема 12.1. Моделирование поведения программной системы Диаграммы схем состояний Диаграммы деятельности

Тема 12.2. Диаграммы взаимодействия Диаграммы сотрудничества Диаграммы последовательности

Тема 12.3. Диаграммы UseCase Спецификация элементов UseCase Пример диаграммы UseCase Построение модели требований Кооперации и паттерны Бизнес-модели

**Вопросы для устного опроса:**

1. Поясните два подхода к моделированию поведения системы. Объясните достоинства и недостатки каждого из этих подходов.
2. Охарактеризуйте вершины и дуги диаграммы схем состояний. В чем состоит назначение этой диаграммы?
3. Как отображаются действия в состояниях диаграммы схем состояний?
4. Как показываются условные переходы между состояниями?
5. Как задаются вложенные состояния в диаграммах схем состояний?
6. Охарактеризуйте средства и возможности диаграммы деятельности.
7. Какие средства диаграммы деятельности позволяют отобразить параллельные действия?
8. Как представляется имя объекта в диаграмме сотрудничества?
9. Поясните синтаксис представления свойства в диаграмме сотрудничества.
10. Какие стереотипы видимости используются в диаграмме сотрудничества? Поясните их смысл.
11. В какой форме записываются сообщения в языке UML? Поясните смысл сообщения.
12. В каком отношении находятся сообщения и действия? Перечислите разновидности действий.
13. Как указывается повторение сообщений?
14. Как показать ветвление сообщений?
15. Что общего в диаграмме последовательности и диаграмме сотрудничества? Чем они отличаются друг от друга?
16. Как отображается порядок передачи сообщений в диаграмме последовательности?
17. Когда удобнее применять диаграммы последовательности?
18. Из каких элементов состоит диаграмма Use Case?
19. Какие отношения разрешены между элементами диаграммы Use Case?
20. Для чего применяют диаграммы Use Case?
21. Чем отличаются друг от друга отношения включения и расширения с точки зрения управления?
22. Каково назначение спецификации элемента Use Case и как она оформляется?
23. Что такое сценарий элемента Use Case?

Практическое занятие№22. Временная диаграмма

Практическое занятие№23. Концептуальные диаграммы

**Тема 13. Модели реализации объектно- ориентированных программных систем**

Тема 13.1. Компонентные диаграммы Использование компонентных диаграмм. Основы компонентной объектной модели.

**Вопросы для устного опроса:**

1. В чем основное назначение моделей реализации?
2. Какие вершины и дуги образуют компонентную диаграмму?
3. Что такое компонент? Чем он отличается от класса?
4. Что такое интерфейс?
5. Какие формы представления интерфейса вы знаете?
6. Чем полезен интерфейс?
7. Какие разновидности компонентов вы знаете?
8. Для чего используют компонентные диаграммы?
9. Каково назначение СОМ? Какие преимущества дает использование СОМ?
10. Чем СОМ-объект отличается от обычного объекта?
11. Что должен иметь клиент для использования операции СОМ-объекта?
12. Как идентифицируется СОМ-интерфейс?
13. Как описывается СОМ-интерфейс?
14. Как реализуется СОМ-интерфейс?
15. Чего нельзя делать с СОМ-интерфейсом? Обоснуйте ответ.
16. Что такое сервер СОМ-объекта и какие типы серверов вы знаете?
17. В чем назначение библиотеки СОМ?
18. Как создается одиночный СОМ-объект?
19. В чем состоят особенности повторного использования СОМ-объектов?
20. Какие требования предъявляет агрегация к внутреннему СОМ-объекту?
21. Как применяют диаграммы размещения?

Практическое занятие№24. Диаграммы размещения

Практическое занятие№25. Диаграммы размещения

**Тема 14. Метрики объектно-ориентированных программных систем**

Тема 14.1. Метрические особенности объектно-ориентированных программных систем

Тема 14.2. Эволюция мер связи для объектно-ориентированных программных систем

Тема 14.3. Набор метрик Чидамбера и Кемерера. Использование метрик Чидамбера-Кемерера. Метрики Лоренца и Кидда.

Тема 14.4. Набор метрик Фернандо Абреу. Метрики для объектно-ориентированного тестирования

**Вопросы для устного опроса:**

1. Какие факторы объектно-ориентированных систем влияют на метрики для их оценки и как проявляется это влияние?
2. Какое влияние оказывает наследование на связность классов?
3. Охарактеризуйте метрики связности классов по данным.
4. Охарактеризуйте метрики связности классов по методам.
5. Какие характеристики объектно-ориентированных систем ухудшают сцепление классов?
6. Объясните, как определить сцепление классов с помощью метрики «зависимость изменения между классами».
7. Поясните смысл метрики локальности данных.
8. Какие метрики входят в набор Чидамбера и Кемерера? Какие задачи они решают?
9. Как можно подсчитывать количество методов в классе?
10. Какие метрики Чидамбера и Кемерера оценивают сцепление классов? Поясните их смысл.
11. Какая метрика Чидамбера и Кемерера оценивает связность класса? Поясните ее смысл.
12. Как добиться независимости метрики WMC от реализации?
13. Как можно оценить информационную закрытость класса?
14. Сравните наборы Чидамбера-Кемерера и Лоренца-Кидда. Чем они похожи? В чем различие?
15. На какие цели ориентирован набор метрик Фернандо Абреу?
16. Охарактеризуйте состав набора метрик Фернандо Абреу.
17. Сравните наборы Чидамбера-Кемерера и Фернандо Абреу. Чем они похожи? В чем различие?
18. Сравните наборы Лоренца-Кидда и Фернандо Абреу. Чем они похожи? В чем различие?
19. Дайте характеристику метрик для объектно-ориентированного тестирования.

**Тема 15. Унифицированный процесс разработки объектно-ориентированных ПС**

Тема 15.1. Эволюционно-инкрементная организация жизненного цикла разработки.

Тема 15.2. Управление риском Этапы унифицированного процесса разработки.

Тема 15.3. Пример объектно-ориентированной разработки. Разработка в стиле экстремального программирования

**Вопросы для устного опроса:**

1. Что является критерием управления унифицированным процессом разработки? Как он применяется?
2. Какую структуру имеет унифицированный процесс разработки?
3. Какие этапы входят в унифицированный процесс разработки? Поясните назначение этих этапов.
4. Какие рабочие потоки имеются в унифицированном процессе разработки? Поясните назначение этих потоков.
5. Какие модели предусмотрены в унифицированном процессе разработки? Поясните назначение этих моделей.
6. В чем суть управления риском?
7. Какие действия определяют управление риском?
8. Какие источники проектного риска вы знаете?
9. В чем суть анализа риска?
10. В чем состоит ранжирование риска?
11. Какие метрики используют для оценки качества унифицированного процесса разработки?
12. Охарактеризуйте содержание ХР-реализации.
13. В чем разница между пользовательскими историями и обычными требованиями к системе?
14. Поясните содержание ХР-итерации.
15. В чем заключается планирование ХР-итерации?
16. Что такое скорость проекта?
17. Поясните структуру элемента ХР-разработки.
18. В чем заключается коллективное владение кодом? Охарактеризуйте содержание такого владения.
19. Как организуется взаимодействие с ХР-заказчиком?
20. Прокомментируйте стоимость ХР-изменения.
21. Поясните особенности ХР-проектирования.

Практическое занятие№26. Разработка в стиле экстремального программирования

**Тема 16. Объектно-ориентированное тестирование**

Тема 16.1. Расширение области применения объектно-ориентированного тестирования Изменение методики при объектно-ориентированном тестировании

Тема 16.2. Проектирование объектно-ориентированных тестовых вариантов. Способы тестирования содержания класса

Тема 16.3. Способы тестирования взаимодействия классов Предваряющее тестирование при экстремальной разработке

**Вопросы для устного опроса:**

1. Поясните особенности тестирования объектно-ориентированных модулей.
2. В чем состоит суть методики тестирования интеграции объектно-ориентированных систем, основанной на потоках?
3. Поясните содержание методики тестирования интеграции объектно-ориентированных систем, основанной на использовании.
4. В чем заключаются особенности объектно-ориентированного тестирования правильности?
5. К чему приводит учет инкапсуляции, полиморфизма и наследования при проектировании тестовых вариантов?
6. Поясните содержание тестирования, основанного на ошибках.
7. Поясните содержание тестирования, основанного на сценариях.
8. Чем отличается тестирование поверхностной структуры от тестирования глубинной структуры системы?
9. В чем состоит стохастическое тестирование класса?
10. Охарактеризуйте тестирование разбиений на уровне классов. Как в этом случае получить категории разбиения?
11. Поясните на примере разбиение на категории по состояниям.
12. Приведите пример разбиения на категории по свойствам.
13. Перечислите известные вам методы тестирования взаимодействия классов. Поясните их содержание.
14. Приведите пример стохастического тестирования взаимодействия классов.
15. Приведите пример тестирования взаимодействия классов путем разбиений.
16. Приведите пример тестирования взаимодействия классов на основе состояний. В чем заключается особенность методики «преимущественно в ширину»?
17. Поясните суть предваряющего тестирования.

Практическое занятие№27. Проектирование объектно-ориентированных тестовых вариантов.

Практическое занятие№28 Тестирования содержания класса

**Тема 17. Автоматизация конструирования визуальной модели программной системы**

Тема 17.1. Общая характеристика CASE- системы RationalRose

Тема 17.2. Создание диаграммы UseCase. Создание диаграммы последовательности. Создание диаграммы классов

Тема 17.3. Создание компонентной диаграммы. Генерация программного кода

**Вопросы для устного опроса:**

1. Чем отличается тестирование поверхностной структуры от тестирования глубинной структуры системы?
2. В чем состоит стохастическое тестирование класса?
3. Охарактеризуйте тестирование разбиений на уровне классов. Как в этом случае получить категории разбиения?
4. Поясните на примере разбиение на категории по состояниям.
5. Приведите пример разбиения на категории по свойствам.
6. Перечислите известные вам методы тестирования взаимодействия классов. Поясните их содержание.
7. Приведите пример стохастического тестирования взаимодействия классов.
8. Приведите пример тестирования взаимодействия классов путем разбиений.
9. Приведите пример тестирования взаимодействия классов на основе состояний. В чем заключается особенность методики «преимущественно в ширину»?
10. Поясните суть предваряющего тестирования.

Практическое занятие№29. Создание диаграммы USE CASE

Практическое занятие№30. Создание диаграммы USE CASE

Практическое занятие№31. Создание диаграммы последовательности

Практическое занятие№32. Создание диаграммы классов

Практическое занятие№33. Создание диаграммы классов

Практическое занятие№34. Создание компонентной диаграммы

Практическое занятие№35. Создание компонентной диаграммы

Практическое занятие№36. Генерация программного кода

Практическое занятие№37.Генерация программного кода