

Аккредитованное образовательное частное учреждение высшего образования
«Московский финансово-юридический университет МФЮА»
Информация о владельце:
ФИО: Забелин Алексей Григорьевич
Должность: Ректор
Дата подписания: 25.08.2022 21:20:40
Уникальный программный ключ:
672b4d4e1ca30b0f66ad5b6309d064a94afcd9bc652d927620ac07f8fdabb79
Рассмотрено и одобрено на заседании
учебно-методического совета

УТВЕРЖДАЮ

Проректор



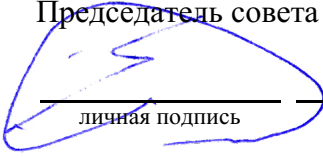
О.А. Минаева

личная подпись инициалы, фамилия

« 14 » марта 2022 г.

Протокол № 7 от 14.03.2022

Председатель совета



Г.С. Горшков

личная подпись

инициалы, фамилия

Калашникова Ольга Алексеевна

(уч. звание, степень, ФИО авторов программы)

Рабочая программа дисциплины (модуля)

Разработка и стандартизация программных средств и информационных технологий

(наименование дисциплины (модуля))

Направление подготовки (специальность): 09.03.03 Прикладная информатика

(код, наименование без кавычек)

ОПОП: Прикладная информатика в экономике

(наименование)

Форма освоения ОПОП: очная, очно-заочная, заочная

(очная, очно-заочная, заочная)

Общая трудоемкость: 4 (з.е.)

Всего учебных часов: 144 (ак. час.)

Формы промежуточной аттестации	СЕМЕСТР		
	очная	очно-заочная	заочная
Экзамен	6	7	7

Москва 2022 г.

Год начала подготовки студентов - 2022

1. Цель и задачи освоения дисциплины

Цель освоения дисциплины	формирование теоретических знаний и практических навыков по проведению разработки и стандартизации программных средств и информационных технологий
Задачи дисциплины	изучение методики применения стандартов разработки программного обеспечения; получение навыков практического применения теоретических подходов к проведению разработки и стандартизации программных средств и информационных технологий; овладение техническими навыками, связанными с использованием современных средств разработки и реализации информационных технологий.

2. Место дисциплины в структуре ОПОП

Блок 1 «Дисциплины (модули)»	
Дисциплины и практики, знания и умения по которым необходимы как "входные" при изучении данной дисциплины	Алгоритмизация и программирование Базы данных Информационные системы и технологии Операционные системы
Дисциплины, практики, ГИА, для которых изучение данной дисциплины необходимо как предшествующее	Проектный практикум

3. Требования к результатам освоения дисциплины

**Компетенции обучающегося, формируемые в результате освоения дисциплины.
Степень сформированности компетенций**

Индикатор	Название	Планируемые результаты обучения	ФОС
ПК2 Способность разрабатывать и адаптировать прикладное программное обеспечение			
ПК-2.1	Знает современные технологии разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки.	Студент должен знать: -современные технологии разработки и адаптации прикладного программного обеспечения, их достоинства и недостатки; -методы научных исследований по теории, технологии и разработки прикладного программного обеспечения	Тест

ПК-2.2	Умеет использовать среду программирования для разработки и адаптации ПО	Студент должен уметь: -формализовать прикладную задачу, выбирать для неё подходящие структуры данных и алгоритмы обработки; -программировать алгоритмы, используя средства языков программирования; -осуществлять разработку кода программного модуля на современных языках программирования; -создавать программу по разработанному алгоритму как отдельный модуль	Практическое задание
ПК-2.3	Владеет навыками проектирования программного обеспечения и разработки прикладных программ	Студент должен владеть навыками: -разработки алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования; -разработки кода программного продукта на основе готовой спецификации на уровне модуля	Практическое задание
ПК3 Способность проектировать ИС по видам обеспечения			
ПК-3.1	Знает технологии проектирования ИС; методы и средства сбора и обработки проектных исследований	Студент должен знать: -понятийный аппарат ИС; -принципы информационного обеспечения; -технологии проектирования ИС; -технологии построения и адаптации ИС; -методы и средства сбора и обработки проектных исследований	Тест
ПК-3.2	Умеет применять типовые решения, классы объектов, библиотеки программных модулей при проектировании ИС	Студент должен уметь: -применять типовые решения, классы объектов, библиотеки программных модулей при проектировании ИС; -проводить выбор ИС при построении сложных профессионально-ориентированных ИС	Практическое задание

ПК-3.3	Владеет навыками осуществления технического контроля состояния и работоспособности программного обеспечения	Студент должен владеть навыками: -использования инструментальных средств программирования информационной системы; -инсталляции, настройки и сопровождения одной из информационных систем; -осуществления технического контроля состояния и работоспособности программного обеспечения	Практическое задание
ПК8 Способность проводить тестирование компонентов программного обеспечения ИС			
ПК-8.1	Знает современные технологии и методы организации тестирования компонентов программного обеспечения ИС	Студент должен знать: -современные технологии и методы организации тестирования компонентов программного обеспечения ИС	Тест
ПК-8.2	Умеет разрабатывать программу и методику тестирования, проводить тестирование компонентов программного обеспечения ИС в соответствии с ними	Студент должен уметь: -использовать методы для получения кода с заданной функциональностью и степенью качества; -разрабатывать программу и методику тестирования; -проводить тестирование компонентов программного обеспечения ИС в соответствии с ними	Практическое задание
ПК-8.3	Владеет основными инструментальными средствами тестирования компонентов программного обеспечения ИС	Студент должен владеть: -навыками разработки тестовых наборов и тестовых сценариев; -основными инструментальными средствами тестирования компонентов программного обеспечения ИС	Практическое задание

4. Структура и содержание дисциплины

Тематический план дисциплины

№	Название темы	Содержание	Литература	Индикаторы
1.	Особенности разработки сложных программных средств	Особенности программных средств (ПС) и задач их разработки. Классификация ПС. Жизненный цикл (ЖЦ) ПС.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-3.1

2.	Создание сложных программных средств	Особенности современной технологии программной инженерии. Модели ЖЦ ПС. Этапы жизненного цикла программного средства. Спецификация. Проектирование, кодирование и тестирование, эксплуатация и сопровождение программы. Автоматизированные средства разработки. Разработка объектно-ориентированного программного обеспечения.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-3.1
3.	Технологический процесс разработки программ и программной документации	Стадии разработки программ и программной документации. Сопровождающая документация. Основные требования к содержанию документации. Техническое задание и требования к его содержанию. Эскизный и технический проекты. Рабочий проект. Внедрение. Единая система программной документации (ЕСПД). Международные стандарты ИСО по разработке и использованию программного обеспечения, документации, интернет - приложений.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-2.1
4.	Проектирование и разработка программных средств	Методы проектирования программных продуктов и признаки их классификации. Структура программного продукта. Стил программирования. Модульное программирование как метод разработки программ. Теория и методы структурного программирования. Объектно-ориентированное программирование.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-2.1 ПК-3.1
5.	Технологические средства разработки программного обеспечения	CASE-системы, системы ускоренной разработки приложений. Системный анализ. Принципы объектно-ориентированного анализа и их обсуждение. Рабочие продукты, методологии и средства анализа и проектирования. Средства визуального программирования.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-2.1 ПК-3.1
6.	Объектное моделирование. Стандарт UML	Язык объектного моделирования UML. Структура системы, декомпозиция, иерархия элементов. Процессы в системе и потоки информации. Исследование действий. Построение моделей доменов и подсистем, связей и взаимодействия подсистем, взаимодействия объектов, событий, процессов, потоков данных, действий. Описание классов и их взаимосвязей. Динамика поведения объектов, диаграммы перехода состояний. Диаграммы объектов. Видимость и синхронизация объектов, временные диаграммы. Диаграмма процессов. Обработка исключительных ситуаций.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-2.1 ПК-8.1

7.	Разработка программного обеспечения	<p>Инструментальные средства операционной системы. Библиотеки операционной системы и их использование в разработке программ. Визуальное проектирование программ. Визуальные среды. Инструменты Visual Studio.</p> <p>Подбор и редактирование компонент, разработка компонент.</p> <p>Построение интерфейса программы. Методы построения интерфейса программного обеспечения.</p>	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-2.2 ПК-2.3 ПК-3.2 ПК-2.1 ПК-3.1
8.	Отладка, тестирование и сопровождение программ	<p>Ошибки программного обеспечения. Источники ошибок программного обеспечения.</p> <p>Классификация ошибок программного обеспечения. Основные пути и методы борьбы с ошибками программного обеспечения.</p> <p>Обнаружение и локализация ошибок ввода и обработки данных.</p> <p>Составляющие процесса отладки. Принципы и виды отладок. Автономная и комплексная отладки программ. Методы отладки. Средства отладки. Рекомендации по организации отладки.</p> <p>Автономная отладка модуля. Использование средств отладки.</p> <p>Сущность и необходимость тестирования программного обеспечения. Различие между тестированием и отладкой программного обеспечения.</p> <p>Основные принципы организации тестирования. Стадии тестирования. Виды тестовых проверок. Объекты тестирования и категории тестов. Виды тестирования.</p> <p>Средства тестирования. Ручное и автоматизированное тестирование. Применение методов и инструментальных средств тестирования.</p> <p>Сопровождение программных продуктов, внесение изменений, обеспечение надежности при эксплуатации. Необходимая документация и предпродажная подготовка программных средств.</p>	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-8.1 ПК-8.2 ПК-8.3
9.	Обеспечение качества программного средства	<p>Эффективность и оптимизация программ.</p> <p>Основные критерии эффективности программного средства.</p> <p>Оптимизация программ на этапе отладки.</p> <p>Принципы обеспечения показателей качества программного продукта.</p> <p>Стандарты, регламентирующие качество программных средств.</p>	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-3.3 ПК-3.2 ПК-8.1

10.	Защита программ	Методы защиты программного средства. Защита программного средства от копирования. Системы разграничения доступа. Криптографические методы защиты программного средства. Аппаратные средства защиты программного средства. Патентная защита. Лицензионные соглашения.	9.1.1, 9.2.1, 9.1.2, 9.1.3, 9.2.2	ПК-8.1
-----	-----------------	--	---	--------

Распределение бюджета времени по видам занятий с учетом формы обучения

Форма обучения: очная, 6 семестр

№	Контактная работа	Аудиторные учебные занятия			Самостоятельная работа
		занятия лекционного типа	лабораторные работы	практические занятия	
1.	3	1	0	2	4
2.	3	1	0	2	6
3.	3	1	0	2	6
4.	3	1	0	2	6
5.	5	1	0	4	6
6.	5	1	0	4	6
7.	6	2	0	4	8
8.	6	2	0	4	8
9.	6	2	0	4	8
10.	4	2	0	2	6
	Промежуточная аттестация				
	4	0	0	0	32
	Консультации				
	0	0	0	0	0
Итого	48	14	0	30	96

Форма обучения: очно-заочная, 7 семестр

№	Контактная работа	Аудиторные учебные занятия			Самостоятельная работа
		занятия лекционного типа	лабораторные работы	практические занятия	
1.	2	1	0	1	6
2.	2	1	0	1	8
3.	2	1	0	1	8
4.	2	1	0	1	8
5.	2	1	0	1	8
6.	2	1	0	1	10
7.	3	1	0	2	10
8.	3	1	0	2	8
9.	4	2	0	2	8
10.	4	2	0	2	8
	Промежуточная аттестация				
	4	0	0	0	32
	Консультации				

	0	0	0	0	0
Итого	30	12	0	14	114

Форма обучения: заочная, 7 семестр

№	Контактная работа	Аудиторные учебные занятия			Самостоятельная работа
		занятия лекционного типа	лабораторные работы	практические занятия	
1.	0.5	0.5	0	0	8
2.	1.5	0.5	0	1	8
3.	1.5	0.5	0	1	10
4.	1.5	0.5	0	1	10
5.	1.5	0.5	0	1	10
6.	1.5	0.5	0	1	10
7.	1.5	0.5	0	1	10
8.	1.5	0.5	0	1	10
9.	1	0	0	1	10
10.	0	0	0	0	10
	Промежуточная аттестация				
	4	0	0	0	32
	Консультации				
	0	0	0	0	0
Итого	16	4	0	8	128

5. Методические указания для обучающихся по освоению дисциплины

В процессе освоения дисциплины студенту необходимо посетить все виды занятий, предусмотренные рабочей программой дисциплины и выполнить контрольные задания, предлагаемые преподавателем для успешного освоения дисциплины. Также следует изучить рабочую программу дисциплины, в которой определены цели и задачи дисциплины, компетенции обучающегося, формируемые в результате освоения дисциплины и планируемые результаты обучения. Рассмотреть содержание тем дисциплины; взаимосвязь тем лекций и практических занятий; бюджет времени по видам занятий; оценочные средства для текущей и промежуточной аттестации; критерии итоговой оценки результатов освоения дисциплины. Ознакомиться с методическими материалами, программно-информационным и материально техническим обеспечением дисциплины.

Работа на лекции

Лекционные занятия включают изложение, обсуждение и разъяснение основных направлений и вопросов изучаемой дисциплины, знание которых необходимо в ходе реализации всех остальных видов занятий и в самостоятельной работе студентов. На лекциях студенты получают самые необходимые знания по изучаемой проблеме. Непременным условием для глубокого и прочного усвоения учебного материала является умение студентов сосредоточенно слушать лекции, активно, творчески воспринимать излагаемые сведения. Внимательное слушание лекций предполагает интенсивную умственную деятельность студента. Краткие записи лекций, конспектирование их помогает усвоить материал. Конспект является полезным тогда, когда записано самое существенное, основное. Запись лекций рекомендуется вести по возможности собственными формулировками. Желательно запись осуществлять на одной странице, а следующую оставлять для проработки учебного материала самостоятельно в домашних условиях. Конспект лучше подразделять на пункты, параграфы, соблюдая красную строку. Принципиальные места, определения, формулы следует сопровождать замечаниями. Работая над конспектом лекций, всегда следует использовать не только основную литературу, но и ту литературу, которую дополнительно рекомендовал лектор.

Практические занятия

Подготовку к практическому занятию следует начинать с ознакомления с лекционным материалом, с изучения плана практических занятий. Определившись с проблемой, следует обратиться к рекомендуемой литературе. Владение понятийным аппаратом изучаемого курса является необходимым, поэтому готовясь к практическим занятиям, студенту следует активно пользоваться справочной литературой: энциклопедиями, словарями и др. В ходе проведения практических занятий, материал, излагаемый на лекциях, закрепляется, расширяется и дополняется при подготовке сообщений, рефератов, выполнении тестовых работ. Степень освоения каждой темы определяется преподавателем в ходе обсуждения ответов студентов.

Самостоятельная работа

Студент в процессе обучения должен не только освоить учебную программу, но и приобрести навыки самостоятельной работы. Самостоятельная работа студентов играет важную роль в воспитании сознательного отношения самих студентов к овладению теоретическими и практическими знаниями, привитии им привычки к направленному интеллектуальному труду. Самостоятельная работа проводится с целью углубления знаний по дисциплине. Материал, законспектированный на лекциях, необходимо регулярно дополнять сведениями из литературных источников, представленных в рабочей программе. Изучение литературы следует начинать с освоения соответствующих разделов дисциплины в учебниках, затем ознакомиться с монографиями или статьями по той тематике, которую изучает студент, и после этого – с брошюрами и статьями, содержащими материал, дающий углубленное представление о тех или иных аспектах рассматриваемой проблемы. Для расширения знаний по дисциплине студенту необходимо использовать Интернет-ресурсы и специализированные базы данных: проводить поиск в различных системах и использовать материалы сайтов, рекомендованных преподавателем на лекционных занятиях.

Подготовка к сессии

Основными ориентирами при подготовке к промежуточной аттестации по дисциплине являются конспект лекций и перечень рекомендуемой литературы. При подготовке к сессии студенту следует так организовать учебную работу, чтобы перед первым днем начала сессии были сданы и защищены все практические работы. Основное в подготовке к сессии – это повторение всего материала курса, по которому необходимо пройти аттестацию. При подготовке к сессии следует весь объем работы распределять равномерно по дням, отведенным для подготовки, контролировать каждый день выполнения работы.

6. Фонды оценочных средств для текущего контроля успеваемости, промежуточной аттестации и самоконтроля по итогам освоения дисциплины

Технология оценивания компетенций фондами оценочных средств:

- формирование критериев оценивания компетенций;
- ознакомление обучающихся в ЭИОС с критериями оценивания конкретных типов оценочных средств;
- оценивание компетенций студентов с помощью оценочных средств;
- публикация результатов освоения ОПОП в личном кабинете в ЭИОС обучающегося;

Тест для формирования «ПК-2.1»

Вопрос №1 .

С точки зрения пользователя программного обеспечения качество последнего заключается в

Тип ответа: Многие из многих

Варианты ответов:

1. легкости эксплуатации
2. модификации
3. Безотказности
4. Производительности
5. воспроизводимости

Вопрос №2 .

Среди уровней абстракции стадий проектирования различают

Тип ответа: Многие из многих

Варианты ответов:

1. способы проектирования
2. специфика дизайна системы
3. детальное кодирование
4. атрибуты и требования приложений
5. стандарты разработки

Вопрос №3 .

UML – это

Варианты ответов:

1. оболочка высокоуровневого языка программирования
2. группа разработчиков программного обеспечения
3. язык моделирования программных систем
4. формат общения «разработчик» — «заказчик»

Вопрос №4 .

Недостаток использования оценки работы по размеру кода связан с

Тип ответа: Многие из многих

Варианты ответов:

1. квалификацией разработчиков
2. сложностью подсчета
3. сложностью реализации
4. его субъективностью
5. Относительностью

Вопрос №5 .

При тестировании методом черного ящика используются следующие критерии

Тип ответа: Многие из многих

Варианты ответов:

1. покрытия операторов
2. синтаксического управляющего тестирования
3. управления логическими спецификациями
4. графа причин и следствий

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	от 0% до 30% правильных ответов из общего числа тестовых заданий
Удовлетворительно	от 31% до 50% правильных ответов из общего числа тестовых заданий
Хорошо	от 51% до 80% правильных ответов из общего числа тестовых заданий
Отлично	от 81% до 100% правильных ответов из общего числа тестовых заданий

Практическое задание для формирования «ПК-2.2»

Задание

Составить программу, работающую в диалоговом режиме (диалог организовать с помощью оператора switch.case, например выбор пунктов меню: "Произвести расчет", "Выход").

Программа должна выполнять следующие действия:

- вывод меню;
- вывод сообщения о назначении программы;

– ввод данных;

– вычисления и вывод результатов.

Результаты работы выводятся в отформатированном виде.

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя
Отлично	Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Практическое задание для формирования «ПК-2.3»

Задание

1. Разработать базовую диаграмму вариантов использования простого механического будильника.
2. Разработать базовую диаграмму вариантов использования банкомата. При разработке диаграммы необходимо учесть, что наличные деньги в банкомате со временем заканчиваются и их требуется пополнить.
3. Разработать базовую диаграмму вариантов использования маршрутного такси.
4. Разрабатывается система для работы с Email. Система должна уметь отправлять и принимать сообщения. При приеме сообщений, если их в системе оказывается более 1000, то предлагается их заархивировать. Администратор системы должен иметь возможность настроить почтовые сервера для отправки и приема сообщений. Администратор системы должен каждую неделю сохранять резервную копию файла почтовых сообщений. Построить расширенную диаграмму вариантов использования.

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя
Отлично	Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Тест для формирования «ПК-3.1»

Вопрос №1 .

Тестирование выполнения программы без знания того, как она спроектирована и запрограммирована называют тестированием методом

Варианты ответов:

1. белого ящика
2. черного ящика
3. темной комнаты
4. методом «орел-решка»

Вопрос №2 .

Процесс обнаружения и исправления ошибок называют

Варианты ответов:

1. верификацией
2. Отладкой
3. тестированием
4. интерпретацией

Вопрос №3 .

Первичной целью любого инженерного продукта является его

Варианты ответов:

1. консолидированность
2. безопасность
3. надежность ПО
4. соответствие требованиям заказчика

Вопрос №4 .

Если планируется использовать абстрактные объекты в распределенном приложении, существует два способа повышения эффективности доступа к ним

Тип ответа: Многие из многих

Варианты ответов:

1. распределение частей абстрактного объекта на нескольких машинах
2. тиражирование распределенного объекта на нескольких компьютерах
3. использование нескольких компьютеров как один
4. создание виртуальных пользователей

Вопрос №5 .

Назначение методологии инженерии программного обеспечения состоит в том, чтобы

Тип ответа: Многие из многих

Варианты ответов:

1. обеспечении применения эффективных методов и приемов проектирования
2. обеспечивать своевременное завершение проекта
3. направлять действия пользователя программного обеспечения
4. выдвигать определенный подход к решению проблемы путем отбора используемых методов и приемов проектирования

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	от 0% до 30% правильных ответов из общего числа тестовых заданий
Удовлетворительно	от 31% до 50% правильных ответов из общего числа тестовых заданий
Хорошо	от 51% до 80% правильных ответов из общего числа тестовых заданий
Отлично	от 81% до 100% правильных ответов из общего числа тестовых заданий

Практическое задание для формирования «ПК-3.2»

Класс, Абстрагирование, Инкапсуляция, Модульность, Иерархия, Наследование

Цели: разработка архитектуры системы и интерфейса пользователя информационной системы.

Средства выполнения задания: MS Visio, Java, C++, Visual Basic, средства пакета MS Office и другие редакторы.

Изучить теоретический материал, дать письменные ответы на контрольные вопросы и выполнить практическое задание.

Практическое задание

Выполнить задания по двум направлениям и составить отчет: по обследованию существующих разработок и проектированию пользовательского интерфейса оригинального приложения.

Рекомендуемая структура отчета:

1. Словесное описание интерфейса пользователя некоторого приложения:

- назначение приложения, сведения о разработчике приложения;
- схема реализованного меню приложения и использованные технологии;
- реализованные разработчиком модели прецедентов (опираясь на опыт работы в приложении или на основе справочной системы).

2. Проектирование интерфейса для создаваемого приложения:

- цели и задачи, область и предмет исследования, используемые технологии, глоссарий предметной области;
- проектирование архитектуры приложения;
- проектирование схемы меню;
- диаграммы состояний интерфейса или графы диалога;
- описание моделей прецедентов.

Справочная система. Дистрибутив

Цели: изучение технологии разработки архитектуры системы, проектирование интерфейса пользователя автоматизированной системы, знакомство с технологией создания дистрибутива.

Теоретический материал

Справочная система

Каждая серьезная программная разработка должна иметь контекстно-чувствительную справочную службу – Help-службу.

Для реализации процесса понадобятся как минимум две утилиты:

- текстовый RTF-редактор и
- компилятор Help-файлов.

В качестве текстового редактора обычно используется Microsoft Word (далее просто Word), поддерживающий расширенный текстовый формат RTF (стандартный для Windows 32 текстовый редактор Word-Pad также поддерживает формат RTF, но в нем нет удобных средств вставки специальных управляющих символов).

Разработка Help-службы требует решения следующих основных задач.

- Планирование системы справок. На этом этапе составляется перечень разделов справочной службы и необходимых перекрестных ссылок.
- Создание текстовых файлов, содержащих описания справочных разделов. Текстовые файлы готовятся с помощью любого текстового редактора, поддерживающего расширенный текстовый формат RTF. В них включаются специальные управляющие символы для создания перекрестных ссылок и подключения растровых изображений.
- Разработка проектного файла, содержащего специальные команды для Help-компилятора. Проектный

файл описывает структуру справочной службы в целом, в нем каждому разделу присваивается уникальный целочисленный идентификатор.

– Разработка файла содержания. Содержание активизируется при запуске HLP-файла, а также после щелчка по кнопке содержание в окне справочной службы.

– Компиляция Hlp-файлов.

– Тестирование и отладка Hlp-службы.

– Связь программы с разделами Hlp-службы.

На этапе планирования необходимо составить перечень разделов справочной службы (СС) и нужных перекрестных ссылок. Структура разделов и количество перекрестных ссылок зависит от сложности программы, для которой создается Hlp-служба, и опыта (и вкуса) разработчика.

Вся справочная информация помещается в один или несколько текстовых файлов в формате RTF. Для их создания может использоваться любой текстовый редактор, поддерживающий этот формат.

Графические врезки

Если создается RTF-файл с помощью редактора Word, можно вставлять в текст СС графические врезки средствами редактора. Единственное ограничение: файлы с графикой должны соответствовать формату bmp. Однако Hlp-компилятор имеет и собственные средства вставки графических врезок с помощью внешних bmp-файлов.

Графический образ может служить ссылкой на пояснение или перекрестной ссылкой. Чтобы придать ему такие свойства, необходимо набрать текст директивы вместе с обрамляющими фигурными скобками перечеркнутым или дважды подчеркнутым шрифтом, если графическая врезка используется в качестве перекрестной ссылки, или подчеркнутым одной линией, если она служит ссылкой на пояснение.

Макрокоманды

При открытии того или иного раздела можно выполнить одну или несколько макрокоманд. С помощью макрокоманд можно гибко воздействовать на состояние окна справочной службы: его положение, размеры, цвет, содержимое меню и инструментальных кнопок, отображать другие Hlp-файлы, выполнять внешние программы и т. д.

Помимо основного окна в справочной службе можно определить множество дополнительных окон, каждое из которых в общем случае будет иметь свой цвет, заголовок, размеры и т. д.

Дистрибутив

Ранее для инсталляции программы достаточно было переписать на жесткий диск ее единственный EXE-файл. Теперь все стало гораздо сложнее. В состав современных прикладных пакетов входят специальные программы инсталляции, которые самостоятельно создают каталоги, переписывают нужные файлы, изменяют содержимое служебных

файлов и всю новую информацию сохраняют на жестком диске, как правило, в нескольких местах. Если необходимо удалить такую программу из системы, то встает вопрос – какие именно файлы удалять, и выяснить это очень трудно.

Программы для создания инсталляторов работают по принципу, схожему с визуальным программированием. Данный метод позволяет просто давать указания: куда и какие файлы копировать, какой экран отображать, выводить ли файл Readme, будет ли фоновый рисунок. При этом достаточно отметить нужные объекты в списке возможностей определенной программы для создания Setup, а дальше пакет все делает автоматически. И через некоторое время будет готов исполняемый файл установки программы, который можно записать на дискету или на компакт-диск.

Вопросы для самоконтроля

1. Какие основные задачи требуется решить при разработке СС?
2. Какие существуют форматы файлов справочной системы?

Варианты представления моделей и средства объектно-ориентированного проектирования

программных средств

Цели: обзор средств проектировщика ПО, получение и закрепление навыков в использовании ООП в проектировании ПО.

Средства выполнения задания: MS Visio, Model Maker, средства пакета MS Office.

Изучить теоретический материал, дать письменные ответы на контрольные вопросы и выполнить практическое задание.

Практическое задание

Построить для задания предыдущей работы и описать с предоставлением отчета по выполнению работы в распечатанном виде:

- диаграмму вариантов использования;
- карту диалогов для двух вариантов использования;
- диаграмму взаимодействия (диаграммы последовательности или кооперативные диаграммы);
- диаграмму классов и пакетов – концептуальную модель;
- диаграммы состояния или активности (диаграммы перехода состояний), диаграммы деятельности;
- построить дерево решений.

Нормативные документы и ЖЦ ПО. Стандарты ЕСПД

Цели: рассмотрение основных понятий, связанных с жизненным циклом программного обеспечения; изучение нормативных документов, регламентирующих состав процессов ЖЦ ПО.

Теоретический материал

После системного анализа предметной области и построения концептуальной модели, содержащей описание основных компонентов и функций, выполняется проектирование структуры приложения. Результатом на этапе проектирования структуры приложения становится детализированная схема программы, на которой указываются все классы и взаимосвязи между ними. Оформление результатов системного анализа предметной области и проектирования структуры приложения должно выполняться единообразно. Существует не одна группа документов рекомендательного характера, позволяющих придерживаться общих принципов.

Специалистам в области информационных технологий практически невозможно обойтись без знаний соответствующих стандартов и спецификаций. Необходимость следования некоторым стандартам закреплена законодательно.

Стандарт – (от англ. standard – норма, образец), в широком смысле слова образец, эталон, модель, принимаемые за исходные для сопоставления с ними др. подобных объектов. Стандарт как нормативно-технический документ устанавливает комплекс норм, правил, требований к объекту стандартизации. Стандарт может быть разработан как на материальные предметы (продукцию, эталоны, образцы веществ), так и на нормы, правила, требования в различных областях. Стандарт также может содержать требования к терминологии, символике, упаковке, маркировке или этикеткам и правилам их нанесения.

Основным нормативным документом, регламентирующим состав процессов ЖЦ ПО, является международный стандарт по ISO/IEC 12207:1995.

Под моделью ЖЦ ПО понимается структура, определяющая последовательность выполнения и взаимосвязь процессов, действий и задач на протяжении всего ЖЦ. Существует несколько моделей ЖЦ ПО: каскадная модель, модель с промежуточным контролем, эволюционная модель, модель формальной разработки систем, модель разработки систем на основе ранее созданных компонентов, инкрементальная модель и спиральная модель.

Состав процессов жизненного цикла регламентируется международным стандартом ISO/IEC 12207:1995 «Information Technology – Software Life Cycle Processes» («Информационные технологии – Процессы жизненного цикла программного обеспечения»). ISO – International Organization for Standardization – Международная организация по стандартизации. IEC (International Electrotechnical Commission) – Международная комиссия по электротехнике.

Стандартизация – деятельность по установлению правил и характеристик в целях их добровольного многократного использования, направленная на достижение в сферах производства и обращения продукции и повышения конкурентоспособности продукции, работ или услуг.

Несмотря на длительное существование компьютерной отрасли, многие компании-разработчики ПО прикладывают значительные усилия для сбора и документирования требований к ПО, для управления ими. Масса проблем с ПО возникает из-за несовершенства способов, которые применяются для сбора, документирования согласования и модификации требований к ПО. Проблемы могут возникать из-за неформального сбора информации, предполагаемой функциональности, ошибочных или несогласованных предположений, недостаточных определенных требований и бессистемного изменения процесса.

Недостаточный объем информации, поступающей от пользователей, требований, сформулированные не полностью, их изменения на этапе разработки – основные причины, из-за которых не удается командам вовремя и в рамках бюджета предоставить клиентам всю запланированную функциональность (см. рис. 5.1).

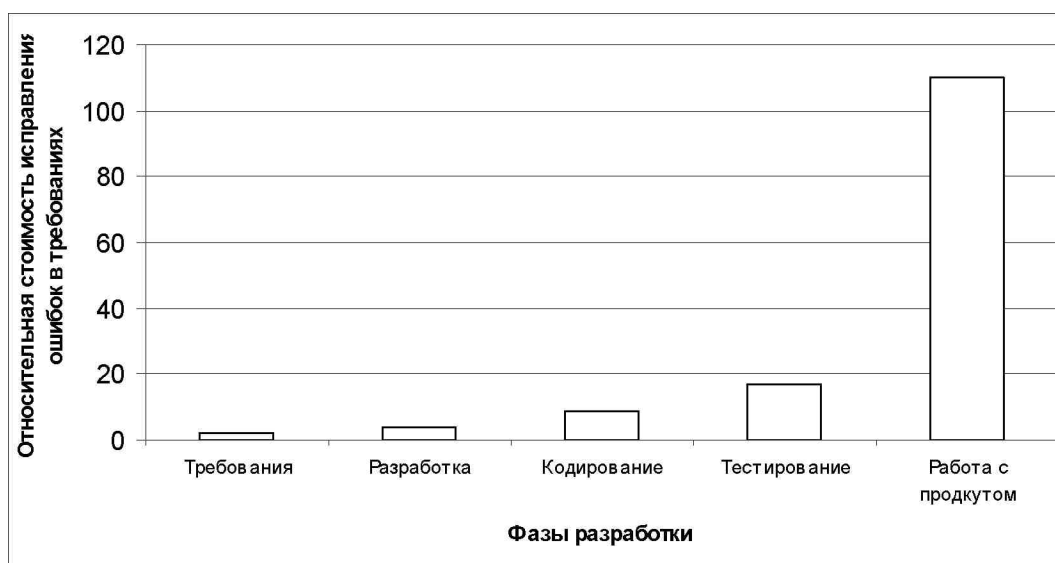


Рис. 5.1. Относительная стоимость исправлений ошибок в требованиях в зависимости от того, когда они обнаружены [24]

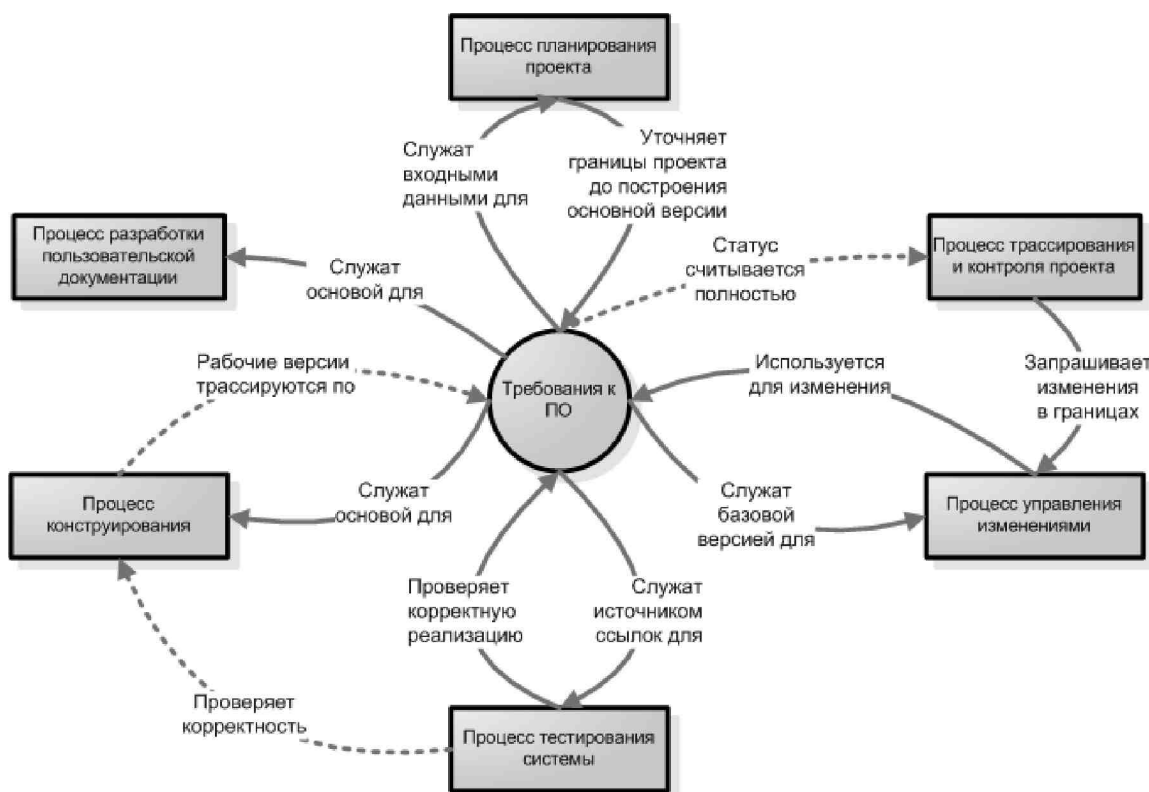


Рис. 5.2. Взаимоотношения требований и других процессов проекта

Существуют особенности в разработке ПО. Разработка ПО включает столько же общения, сколько и обычная работа с компьютером. Иногда участники проекта не могут прийти к единому мнению, что такое «требование». Но при этом требования составляют основу любого хорошо организованного проекта по разработке ПО, поддерживая технические и организационные задачи (рис. 5.2).

Что называть «требованием к ПО»? Одна из проблем индустрии программного обеспечения — отсутствие общепринятых определений терминов, используемых для описания работы.

IEEE Standard Glossary of Software Engineering Terminology (1990) содержит следующее определение термина «требования»:

- условия или возможности, необходимые пользователю для решения проблем или достижения целей;
- условия или возможности, которыми должна обладать системы или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
- документированное представление условий или возможностей для пунктов 1 и 2.

Это определение охватывает требования как пользователей (внешнее поведение системы), так и разработчиков (некоторые скрытые параметры).

Требования к ПО состоят из трех уровней: бизнес-требования, требования пользователей и функциональные требования. Каждая система требований имеет свои нефункциональные требования. Эти типы требований имеют различный способ представления: документы, диаграммы, базы данных.

Процесс создания требований состоит из выявления, анализа, спецификации и проверки требований (рис. 5.3), и он не выполняется последовательно и за один проход.

Многие разработчики ПО вынуждены на практике осваивать теорию сбора требований, осваивают обязанности аналитика требований: работают с клиентом, собирая, анализируя и документируя требования.

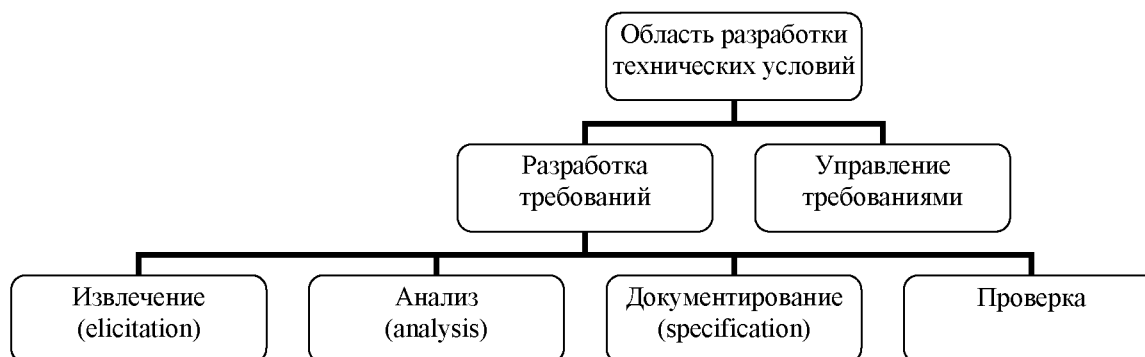


Рис. 5.3. Компоненты области разработки технических условий

На практике эти действия выполняются попеременно, поэтапно и повторяются (см. рис. 5.4). Процесс формулирования требований очень важен, и все участники проекта должны понимать концепцию и приемы формулирования требований. Пример разработки требований к системе представлен в Приложении А.

Во время работы с проектом разработчик ПО сталкивается с тремя уровнями требований: бизнес-уровень, пользовательский уровень и функциональный.

Итог разработки требований – документированное соглашение между клиентами и разработчиками о создаваемом продукте. Существует несколько способов представления требований: документация, графические модели и формальные спецификации

Анализ требований подразумевает их детализацию, гарантирующую, что требования понимают все заинтересованные лица, детальное исследование требований на предмет ошибок, пробелов и других недостатков.

Составлению технического задания предшествует составление «Спецификаций требований».

Спецификация – достаточно точное и полное описание задачи, доступное для понимания всех

заинтересованных лиц. Спецификация требований (specification, requirements) – процесс документирования системы в структурированном, доступном всем и управляемом формате.

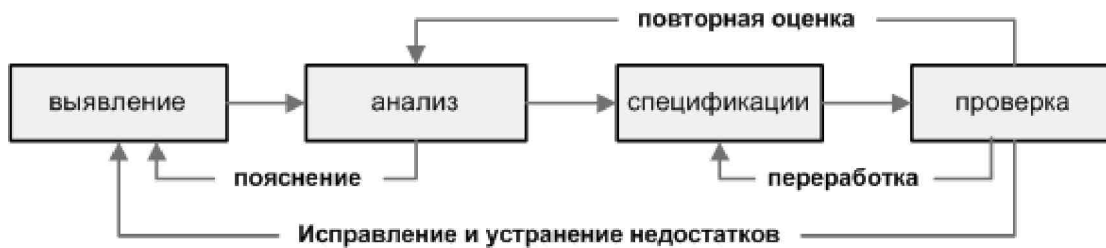


Рис. 5.4. Итеративный процесс формирования требований

Спецификация требований к продукту (software requirements specification) – набор функциональных и нефункциональных требований к продукту ПО.

Анализ включает создание прототипов, анализ осуществимости и согласованности приоритетов.

Прототип (prototype) – частичная, предварительная или возможная реализация программы.

Применяется для исследования и утверждения требований, для разработки приемов. Виды прототипов: эволюционные, одноразовые, бумажные, горизонтальные и вертикальные.

Модели прецедентов описывает способ взаимодействия пользователя с системой.

Прецедент – это набор сценариев использования, в котором каждый экземпляр сценария представляет собой последовательность действий, выполняемых системой для достижения ощутимого для конкретного исполнителя результата.

Любое техническое задание должно содержать разделы, отражающие сведения:

- что надо сделать;
- для чего, с какой целью надо создать продукт;
- где, в какой области применения, на каком объекте продукт должен решать задачи и выполнять свои функции;
- какие требования будут предъявлены к продукту;
- какие работы потребуется выполнить, чтобы создать продукт;
- каков порядок приемки-сдачи работ Заказчику;
- как должно быть задокументировано проведение работ;
- на основании каких нормативно-технических документов должны проводиться работы.

При разработке технического задания на автоматизированную систему используется ГОСТ 34.602-89, при разработке ТЗ на программу - ГОСТ 19.201-78.

В основе методов управления проектами лежат методики сетевого планирования:

- заблаговременное планирование работ над проектом;
- планирование завершения работ в нужные сроки;
- координирование и контролирование выполнения работ для соблюдения календарного графика и завершения работ в срок.

Планирование проекта включает определение ресурсов (человеческих, вычислительных и организационных) и составление «карты» задач и времени их выполнения. Пример определения таких ресурсов, требований к ним представлен в Приложении В.

Существует несколько методик оценок времени и затрат.

Вопросы для самоконтроля

1. Дать определения:
 - программа;

- программный продукт;
- программное средство;
- ПО;

– жизненный цикл ПО; – качество ПО.

1. Дать определение терминов: «требования», «спецификация».
2. Что подразумевается под «успех проекта»?
3. Характеристики превосходных требований.
4. Какой стандарт ЕСПД определяет содержание Технического задания? Назначение документа и его обязательные разделы.
5. Характеристика основных уровней стандартизации.
6. Стандарты документирования ПО. Перечислите основные виды нормативных документов.
1. Какие проблемы сопровождают внутрифирменные стандарты?
2. Схема классификации стандартов в области ИТ.
3. Эволюция стандартов ПО.
4. ЖЦ ПО. Эволюция ЖЦ ПО (по ISO/IEC 12207:1995). Процессы ЖЦ, регламентируемые стандартом ISO/IEC 12207.
5. Содержание государственного стандарта «Единая система программной документации».
6. Критерии качества ПО, факторы влияющие на качество ПО.
7. Уровни требований к ПО.

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя
Отлично	Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Практическое задание для формирования «ПК-3.3»

Модель профиля стандартов жизненного цикла сложных программных средств

Цели: изучение международных стандартов в области ПО, рассмотрение технологии функционального и системного моделирования, построение структурных моделей бизнеса.

Средства выполнения задания: средства пакета MS Office, ВРwin 4.1 и выше, Visible Analyst.

Изучить теоретический материал, дать письменные ответы на контрольные вопросы и выполнить практическое задание.

Практическое задание

1. На основе результатов предыдущих лабораторных работ описать логику взаимодействия информационных потоков в нотации IDEF3.
2. Дать предложения по автоматизации работ модели «КАК-ЕСТЬ» в виде модели «КАК-ДОЛЖНО-БЫТЬ» с использованием CASE-средства ВРwin. Полученные диаграммы будут использоваться при выполнении последующих лабораторных работ.

Объектно-ориентированная методология анализа и построения спецификаций

Цели: обзор средств проектировщика ПО, получение и закрепление навыков в использовании ООП в

проектировании ПО.

Теоретический материал

На данный момент не является тайной, что для успешной конкуренции стало недостаточно просто укладываться в бюджет и сроки. Опыт софтверных компаний как зарубежных, так и отечественных, позволил сформулировать так называемое Правило «80/20»:

Правило «80/20»: успешность проекта на 80% определяется регламентом работ и только на 20% – применяемым инструментарием.

Проектирование архитектуры для структурной методологии включает следующие основные методы: метод нисходящего проектирования, метод восходящего проектирования, метод расширения ядра.

В основе объектно-ориентированного подхода лежит объектная декомпозиция: структура системы описывается в терминах объектов и связей между ними, поведение системы описывается в терминах обмена сообщениями между объектами. Рекомендуется перед выполнением практических заданий ознакомиться с источниками.

Объектно-ориентированный подход позволяет сократить время на разработку систем при условии повторного использования заготовок (шаблоны, паттерны).

Объектно-ориентированный анализ (ООА) – процесс применения объектно-ориентированного подхода для понимания задачи. После завершения анализа требования к информационной системе должны стать понятными членам группы, и должен быть составлен словарь предметной области.

Спецификация разрабатываемого ПО с использованием UML (Unified Modeling Language) объединяет несколько моделей: модель использования, логическую модель, модель реализации, модель процессов и модель развертывания (см. рис. 4.1). Предлагается создавать дополняющие друг друга диаграммы: 1) диаграммы вариантов использования; 2) диаграммы классов; 3) диаграммы пакетов; 4) диаграммы последовательностей действий; 5) диаграммы коопераций; 6) диаграммы деятельностей; 7) диаграммы состояний объектов, 8) диаграммы компонентов; 9) диаграммы размещения.

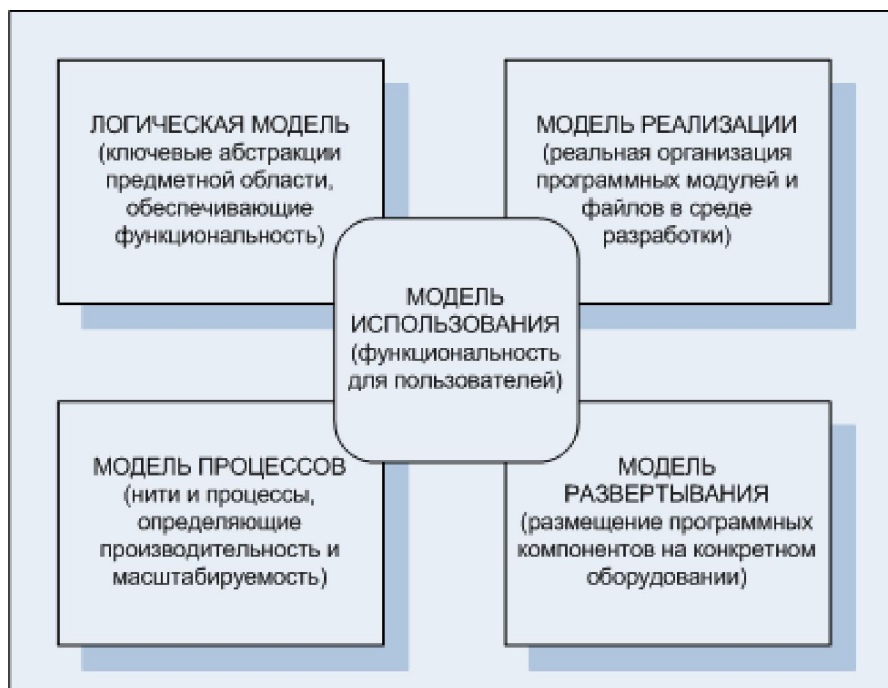


Рис. 4.1. Спецификация ПО при использовании UML

Немаловажную роль при проведении ООА играют модели прецедентов. Модели прецедентов описывает способ взаимодействия пользователя с системой.

Прецедент — это набор сценариев использования, в котором каждый экземпляр сценария представляет собой последовательность действий, выполняемых системой для достижения ощутимого для конкретного исполнителя результата.

Создание прецедентов выполняется при последовательности действий:

– определение действующих субъектов;

- создание предварительного списка прецедентов;
- уточнение и обозначение прецедентов;
- определение последовательности событий для каждого прецедента;
- моделирование прецедентов.

При выполнении анализа прецедентов устанавливается картина взаимодействий в системе. После построения описания прецедентов и создания диаграмм взаимодействия, приступают к построению концептуальной модели – основы системы. Пример описания прецедента представлен в Приложении Б. Концептуальная модель не зависит от конкретных способов реализации или использования системы.

Диаграммы вариантов использования

Вариантом использования называется характерная процедура применения разрабатываемой системы конкретным лицом (люди, другие системы, устройства).

Варианты использования в зависимости от целей выполнения конкретной процедуры: основные, вспомогательные и дополнительные.

Пример описания варианта использования системы для формирования выписки из учетной карточки студента:

Название варианта	Формирование выписки из учетной карточки студента
Цель	Получение выписки изученных дисциплин с оценками
Действующие лица	Документовед
Краткое описание	Формирование выписки предполагает выбор студента, проверка полученных данных, внесение тем курсовых работ и печать результата
Тип варианта	Основной

Основные варианты использования должны быть описаны подробно. Подробная форма включает описание типичного хода событий, альтернатив и исключений (Приложение Б).

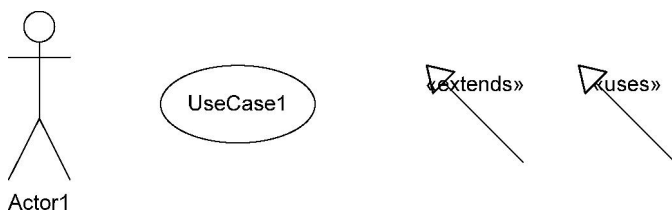


Рис. 4.2. Условные обозначения диаграмм вариантов использования

Диаграммы вариантов использования используют объекты (рис. 4.2): фактор (действующее лицо), вариант и связь.

Варианты использования могут быть связаны между собой: связи использования (uses) и связи расширения (extends).

Состояние объекта (диаграмма состояний) – есть ситуация в жизненном цикле объекта, во время которого он:

1. удовлетворяет некоторому условию,
2. осуществляет определенную деятельность или
3. ожидает некоторое событие.

Диаграмма состояний показывает состояния объекта, возможные переходы, события или сообщения, вызывающие каждый переход. Пример диаграммы состояний представлен на рис. 4.3.

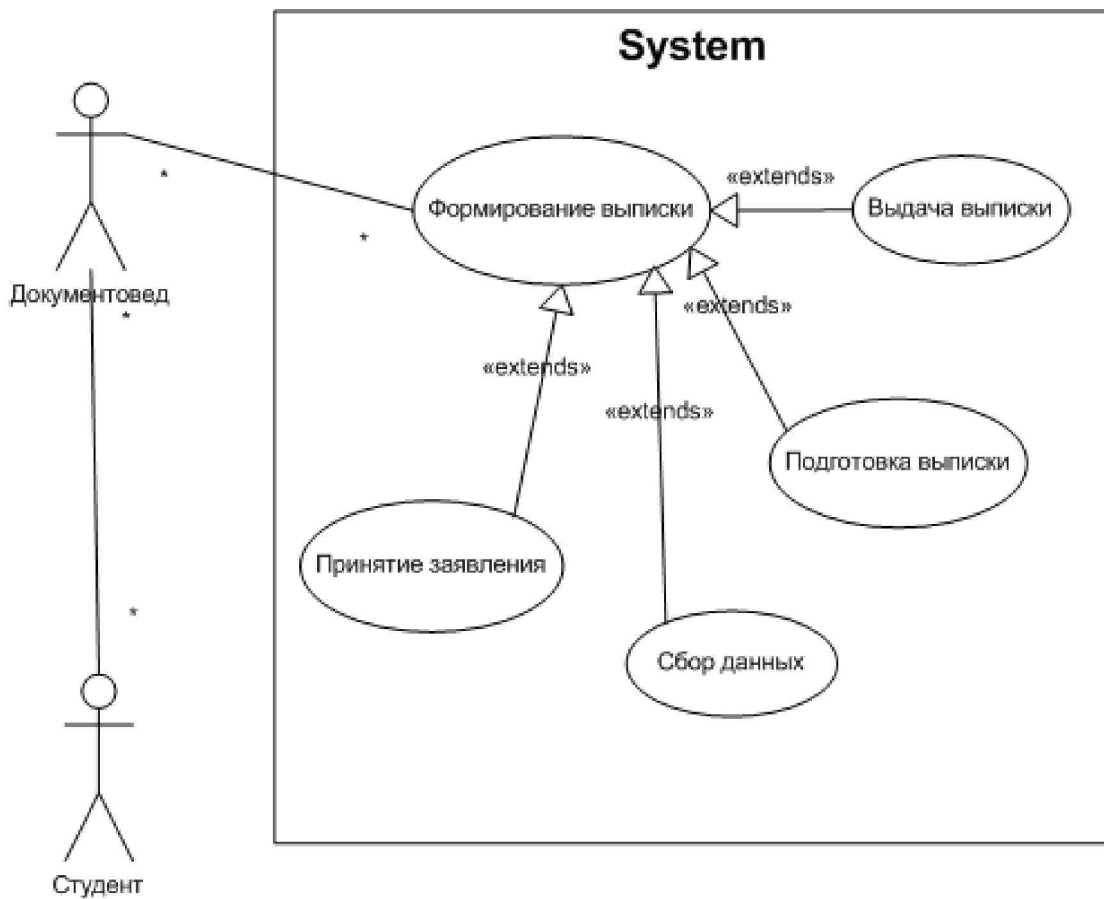


Рис. 4.3. Диаграмма варианта использования системы

Обзор CASE-средств

Rational Rose использует синтез-методологию объектно-ориентированного анализа и проектирования, основанную на подходах трех ведущих специалистов в данной области: Буча, Рамбо и Джекобсона. Разработанная ими универсальная нотация для моделирования объектов (UML) претендует на роль стандарта в области объектно-ориентированного анализа и проектирования.

В основе работы Rational Rose лежит построение различного рода диаграмм и спецификаций, определяющих логическую и физическую структуры модели, ее статические и динамические аспекты. В их число входят диаграммы классов, состояний, сценариев, модулей, процессов.

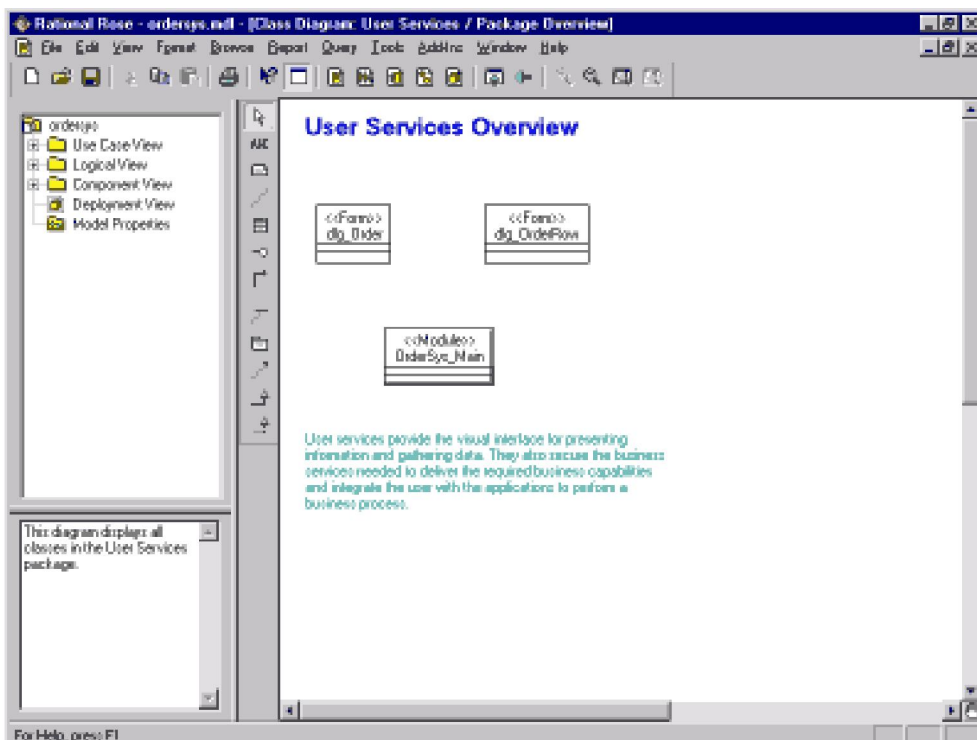


Рис. 4.4. CASE-средство Rational Rose

В результате разработки проекта с помощью CASE-средства Rational Rose (рис. 4.4) формируются следующие документы:

- диаграммы классов;
- диаграммы состояний;
- диаграммы сценариев;
- диаграммы модулей;
- диаграммы процессов;
- спецификации классов, объектов, атрибутов и операций;
- заготовки текстов программ;
- модель разрабатываемой программной системы.

Последний из перечисленных документов является текстовым файлом, содержащим всю необходимую информацию о проекте (в том числе необходимую для получения всех диаграмм и спецификаций).

Для организации групповой работы в Rational Rose возможно разбиение модели на управляемые подмодели. Каждая из них независимо сохраняется на диске или загружается в модель. В качестве подмодели может выступать категория классов или подсистема.

Пакеты MS Visio и ModelMaker также использует методологию ООА. Принцип создания моделей отличается незначительно (рис. 4.5, 4.6).

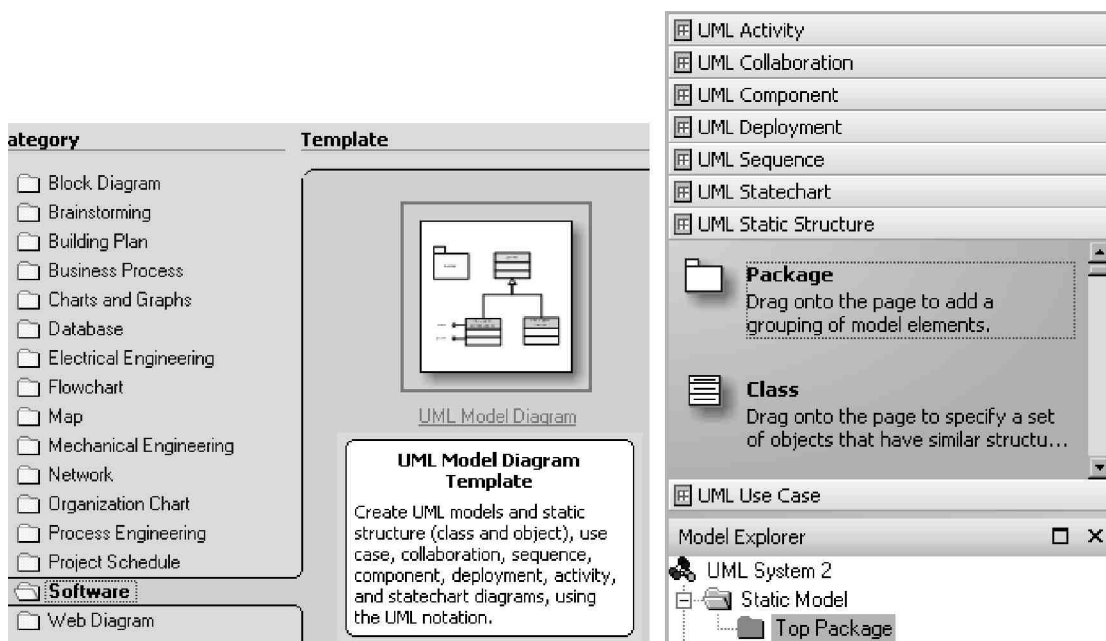


Рис. 4.5. Создание ОО-модели средствами MS Visio

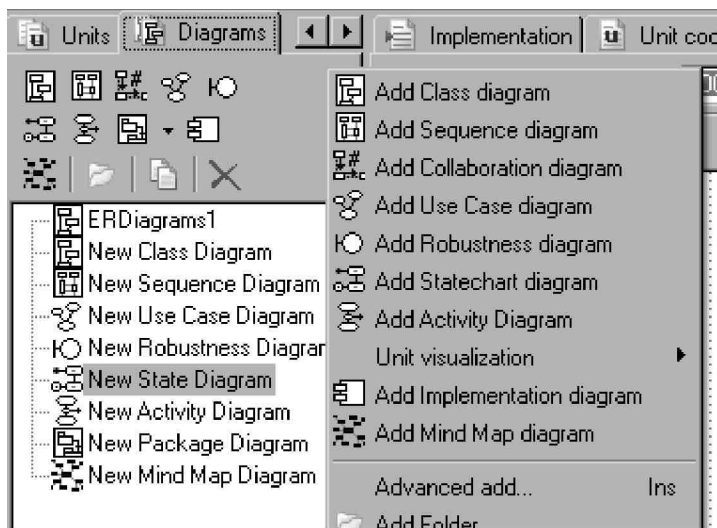


Рис. 4.6. Выбор диаграмм в ModelMaker

Вопросы для самоконтроля

1. В чем заключаются основные принципы ООА?
2. Различия и общее в структурном и объектно-ориентированном подходах.

Сертификация программного обеспечения как средство обеспечения адекватности, надежности и безопасности использования ИС

Цели: закрепление основных понятий, связанных с жизненным циклом программного обеспечения.

Средства выполнения задания: средства пакета MS Office.

Изучить теоретический материал, дать письменные ответы на вопросы для самоконтроля и выполнить практическое задание.

Практическое задание

1. Выполнить описание программы статистической обработки числовых данных, хранящихся в тестовом файле, согласно ГОСТ 19.402-78 ЕСПД.
2. Найти в WWW примеры спецификаций требований к рекламируемому программному обеспечению (3-4 примера).
3. Создать шаблон спецификаций требований к ИС. На базе шаблона создать документ Спецификации требований на создание «АРМ Секретарь руководителя предприятия».
4. Описать требования в контексте модели прецедентов (см. Приложение Б).
5. Создать одностраничный проект на разработку автоматизированной системы (см. Приложение В).
6. Создать шаблон технического задания на разработку ИС. Используя различные методологии, на базе шаблона создать документы Техническое задание на создание: локальной ИС; распределенной ИС; Web-сайта.
7. Составить график работ, список участников группы и произвести расчет стоимости работ коллектива для автоматизации процессов согласно предложениям по предыдущим лабораторным работам.

Цели: изучение технологии разработки интерфейса пользователя автоматизированной системы.

Теоретический материал

После оформления всех необходимых документов, содержащих сведения для реализации программистом проекта, наступает этап написания программного кода – реализации проектных решений. Немалую роль сыграло появление средств быстрой разработки – Rapid Application Development (RAD), позволивших значительно сократить время и средства на реализацию. RAD-системы позволяют спроектировать архитектуру и интерфейс пользователя будущей системы, внести безболезненно необходимые изменения, и ввести необходимый программный код. Использование RAD-систем привело к тому, что традиционный подход к оценке сложности кода и оценке труда программиста потерял смысл.

Пользовательский интерфейс обеспечивает взаимодействие между пользователем и системой. Почти каждая современная система имеет ту или иную форму пользовательского интерфейса: графическую, командной строки, телефонную или речевую.

Пользовательский интерфейс – важная составляющая любой системы. Для некоторых пользователей интерфейс – единственная часть системы, с которой они взаимодействуют, для них интерфейс и есть сама система.

К анализу, проектированию и реализации пользовательского интерфейса следует подходить так же, как и к анализу проектирования и реализации любой части программы. Требования к пользовательскому интерфейсу могут постоянно изменяться. Улучшения системы – результат добавления новых возможностей и устранение выявленных недостатков пользователями.

Процесс разработки и программирования пользовательского интерфейса по существу не отличается от процесса разработки и программирования других частей системы. Создавая пользовательский интерфейс, иногда приходится изучать некоторые функции программного интерфейса приложения (API-функции).

Различают четыре типа пользовательских интерфейса: примитивные, меню, со свободной навигацией, прямого манипулирования (рис. 6.1). Выбор интерфейса включает выбор технологии работы с документами: однодокументная (SDI) и многодокументная (MDI) технологии.

При создании системы необходимо предусмотреть несколько различных, зачастую несвязанных между собой, пользовательских интерфейсов.

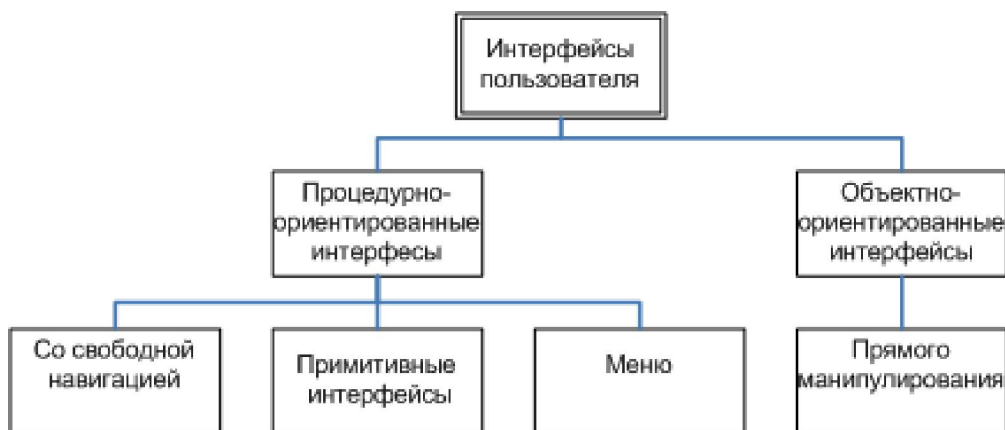


Рис. 6.1. Типы интерфейсов

Требования к пользовательскому интерфейсу могут постоянно изменяться. Улучшение системы являются результатом добавления новых возможностей и устранения обнаруженных пользователем недостатков. Для предоставления новых возможностей и устранения обнаруженных дефектов, пользовательский интерфейс приходится обновлять постоянно. Поэтому пользовательский интерфейс должен быть спроектирован так, чтобы он был достаточно гибким и легко адаптирован к переменам.

Разработчики не должны навязывать пользователю свою точку зрения на проектируемый интерфейс. Пользователи должны принимать активное участие в процессе проектирования интерфейса. Поход, ориентированный на пользователя, называется user-centered design. Он основан на прототипировании интерфейса и участии пользователя в процессе его проектирования.

Из-за динамической природы пользовательских интерфейсов текстовых описаний и диаграмм недостаточно для формирования требований к интерфейсу. Поэтому эволюционное прототипирование с участием конечного пользователя – единственный способ разработки графического интерфейса системы.

Генераторы интерфейсов – графические системы проектирования экранных форм, где интерфейсы компонуются из элементов типа меню, полей, пиктограмм и кнопок, которые легко выбрать из хранилища и поместить на экранную форму. Системы такого типа – необходимая часть систем программирования баз данных.

Процесс проектирования и разработки интерфейса пользователя

Для достижения гибкости разрабатываемая система должна быть полностью независимой от

пользовательского интерфейса. Это можно объяснить тем, что если система независима от пользовательского интерфейса, то легко можно изменить вид существующего интерфейса или добавить новый. Причем изменения в самой системе не требуются.

При этом систему можно тестировать еще до того, как завершена разработка интерфейса пользователя. Объектно-ориентированное программирование идеально подходит для этого: удастся разумно распределить выполняемые функции между частями программы, можно в любое время добавить к программе любой интерфейс без изменений в основной программе. Код интерфейса и код системы должны быть независимыми. При реализации пользовательского интерфейса используются метафоры и анимация.

Процесс проектирования и разработки пользовательского интерфейса пользователя состоит из четырех основных этапов:

Первый. Сбор и анализ информации от пользователя.

- определение профиля пользователей;
- анализ стоящих перед ними задач;
- сбор требований, предъявляемых клиентами;
- анализ рабочей среды пользователей;
- соответствие требований пользователей стоящим перед ними задачам.

Второй. Разработка пользовательского интерфейса.

- определение цели с точки зрения удобства применения продукта;
- разработка задач и сценария действия пользователя;
- определение целей и операций пользователя;
- определение иконок объектов и визуального представления;
- разработка меню объектов и окон;
- оптимизация визуальной разработки.

Третий. Построение пользовательского интерфейса.

- прототипирование на ранних стадиях с учетом итерационного принципа;
- создание альтернативных вариантов;
- готовность отказа от кода прототипа.

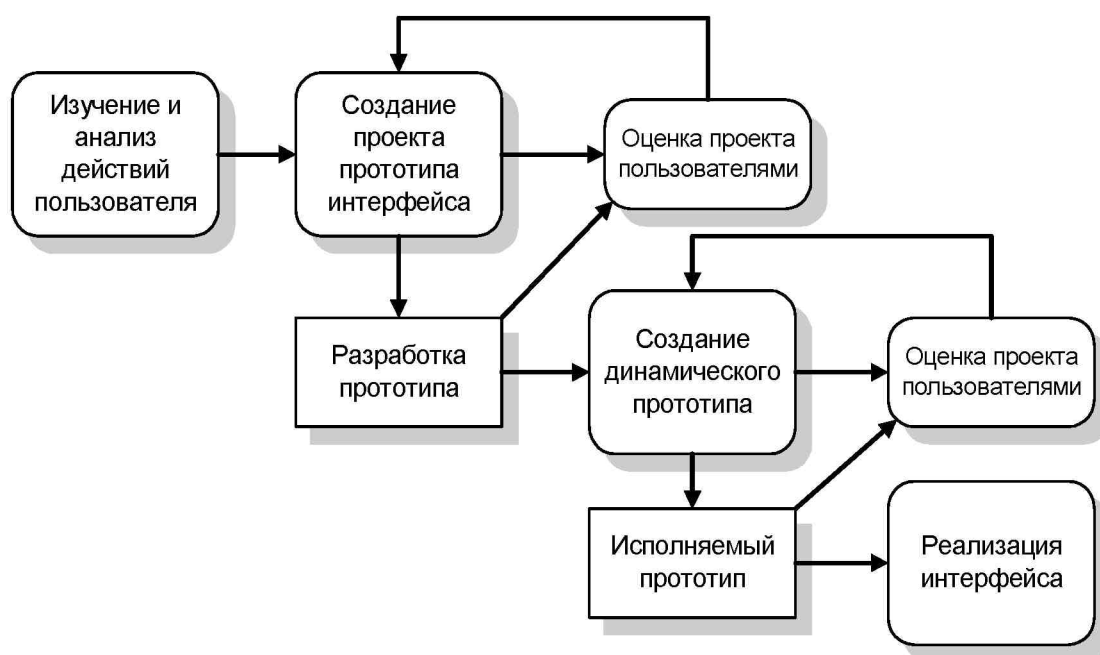


Рис. 6.2. Процесс проектирования интерфейса пользователя

Четвертый. Подтверждение качества созданного пользовательского интерфейса.

– проведение тестирования на удобство применения на каждой стадии разработки продукта.

На рис. 6.2 представлен итерационный процесс проектирования пользовательского интерфейса.

Основой принципов проектирования интерфейсов пользователя являются человеческие возможности. Представлены основные принципы, применимые при проектировании любых интерфейсов пользователя:

- учет знаний пользователя (интерфейс должен быть удобен при реализации, должны использоваться понятные пользователю термины, объекты, которыми управляет система, должны быть напрямую связаны с рабочей средой пользователя),
- согласованность (команды и меню системы должны быть одного формата, параметры должны передаваться во все команды одинаково и пунктуация должна быть одинаковой);
- минимум неожиданностей;
- способность к восстановлению (т.е. наличие возможности отмены действий и требование подтверждения деструктивных действий);
- поддержка пользователя (средства поддержки должны быть встроены в интерфейс, должно быть несколько уровней справочной информации – от основ до полного описания возможностей системы);
- учет разнородности пользователей (нерегулярно использующие систему, опытные).

Критерии оценки интерфейса пользователя:

- простота освоения и запоминания операций системы;
- скорость достижения результатов при использовании системы;
- субъективная удовлетворенность при эксплуатации.

Вопросы для самоконтроля

1. Раскрыть принципы RAD-технологий.
2. Дать определения для:
 - Интерфейс
 - Глоссарий проекта
 - Интерфейс управления –
 - Информационный интерфейс
 - Интерфейс ввода-вывода
 - Внешний интерфейс
1. Принципы и правила проектирования интерфейса пользователя.
2. Что значит «user-centered design» интерфейс?
3. Система стандартов «Common user access» фирмы IBM.

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя

Отлично

Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Тест для формирования «ПК-8.1»

Вопрос №1 .

Совокупность организационных структур, методик, технологий и ресурсов, необходимых для осуществления общего руководства качеством – это

Варианты ответов:

1. система качества
2. стандартизация
3. сертификация
4. метрология

Вопрос №2 .

Совокупность нескольких базовых стандартов и/или других нормативных документов с четко определенными и гармонизированными подмножествами обязательных и дополнительных возможностей, предназначенная для реализации заданной функции или группы функций – это

Варианты ответов:

1. профиль стандартов
2. группа стандартов
3. классификация стандартов
4. множества стандартов

Вопрос №3 .

Снижение трудоемкости, длительности проектов ПС, повышение качества разрабатываемых ПС, разработке, эксплуатации и сопровождении, обеспечение возможности расширять программное средство по набору прикладных функций и масштабировать в зависимости от размерности решаемых задач и другое являются:

Варианты ответов:

1. целями применения стандартов
2. методами применения стандартов
3. поводами применения стандартов
4. заменой применения стандартов

Вопрос №4 .

Какими бывают первичные ошибки

Тип ответа: Многие из многих

Варианты ответов:

1. технологические ошибки
2. программные ошибки
3. алгоритмические ошибки
4. системные ошибки

Вопрос №5 .

Результатом системного проектирования являются

Тип ответа: Многие из многих

Варианты ответов:

1. системный проект
2. техническое задание
3. договор на продолжение проектирования
4. выявление системных ошибок

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	от 0% до 30% правильных ответов из общего числа тестовых заданий
Удовлетворительно	от 31% до 50% правильных ответов из общего числа тестовых заданий
Хорошо	от 51% до 80% правильных ответов из общего числа тестовых заданий
Отлично	от 81% до 100% правильных ответов из общего числа тестовых заданий

Практическое задание для формирования «ПК-8.2»

Состав требований к разрабатываемой системе

Задача: Разработать требования к системе «Справочник по компьютерной технике» и построить для нее модель прецедентов.

Описание предметной области

В настоящее время в городе работает около 2-х десятков наиболее известных фирм по продаже компьютеров и комплектующих к ним. Большую часть ассортимента составляют одинаковые товары, отличающиеся только ценой. Как правило, конкретная фирма имеет преимущества в цене перед другими по какому-то определенному виду комплектующих (например, относительно дешевые жесткие диски, а цены на другие комплектующие не ниже средних по городу).

При выборе компьютера часто лимитирующим фактором для покупателя является цена. Любой покупатель желает приобрести компьютер с наивысшими техническими показателями по минимальной цене. Может оказаться, что выгоднее покупать компьютер «по частям» в разных фирмах. Для этого пользователю необходимо оперативно иметь информацию о ценах на комплектующие в нескольких фирмах города.

Потенциальный покупатель перед покупкой анализирует прайс-листы отдельных фирм. При этом им выбираются при помощи эксперта комплектующие с интересующими характеристиками, и сравниваются цены на них в разных фирмах. Для покупателя рассчитывается цена собранного компьютера и рассматриваются альтернативные варианты. На основании этого анализа принимается решение о том, какие комплектующие и в каких фирмах приобретать.

Также покупателя или какого-то специалиста могут очень заинтересовать технические характеристики определенной модели, отзывы о ней специалистов и другая подобная информация.

Функциональные требования к системе

1. В системе должна храниться информация о ценах на комплектующие компьютеров в нескольких фирмах по городу.
 2. В системе должна быть предусмотрена возможность обновления каталога товаров и цен на основе общей формы прайс-листов.
 3. Пользователь системы должен иметь возможность подбирать комплектующие по интересующим значениям параметров и пределам цены в разных фирмах.
 4. При подборе полной конфигурации компьютера система должна «запоминать» выбранные пользователем комплектующие и вносить их в комплектацию компьютера с учетом совместимости.
 5. Система должна обеспечивать хранение нескольких вариантов конфигурации компьютера, выбранных пользователем, для последующего анализа.
1. Система должна содержать справочную информацию о комплектующих компьютера, отзывы экспертов, мнения специалистов и т.п.
 2. Система должна обеспечивать пересылку файла с выбранными пользователем вариантами конфигураций компьютера.

Количественные требования к системе

Проектируемая система должна хранить информацию не менее, чем о 20 фирмах, в каждой из которых имеется приблизительно от 100 до 1500 наименований товаров и соответственно цены на них.

Требования по совместимости

Система должна корректно работать со стеком протоколов TCP/IP. Система должна иметь интерфейс для обновления каталога товаров через стандартные формы прайс-листов.

Построение модели прецедентов

Требования, предъявляемые к функционированию проектируемой системы, удобно выразить с помощью языка прецедентов. Прецедент – это набор сценариев использования, в котором каждый экземпляр сценария представляет собой последовательность действий, выполняемых системой для достижения ощутимого для конкретного исполнителя результата. Таким образом, с помощью прецедентов на понятном и доступном языке можно описать основные процессы, происходящие в системе и значения этих процессов для актера (пользователя системы).

Описание прецедента «Выдать варианты конфигурации»

Прецедент «Выбрать узел комплектации компьютера»

Основной исполнитель: Пользователь
Заинтересованные лица и их требования

Пользователь. Желает быстро получить приемлемые варианты узла комплектации компьютера с заданными значениями параметров и заданной ценой.

Компьютерные фирмы. Заинтересованы в корректной работе системы, т.к. пользователь является их потенциальным клиентом. Владелец сайта. Желает удовлетворить интересы пользователя, заинтересован в привлечении новых клиентов и рекламодателей. Предусловия

В системе имеется каталог комплектующих с указанием значений основных параметров, цены и отзывов. Результаты (постусловия)

Узел компьютера и фирма-продавец выбраны пользователем и запомнены системой.

Основной (успешный) сценарий

1. Пользователь заходит на сайт, где размещен справочник по компьютерной технике.
 1. Система предлагает выбрать основные узлы компьютера (материнская плата, процессор, оперативная память и т.п.).
 2. Пользователь выбирает конкретный узел комплектации компьютера.
 3. Система предлагает ввести значения интересующих параметров и границы цены для сужения круга поиска.
 4. Пользователь указывает значения интересующих параметров.
 5. Система выводит список вариантов узла с указанием всех параметров и предлагает пользователю выбрать фирму для вывода цены.
 6. Пользователь выбирает из списка фирму.
 7. Система указывает цену на узел в данной фирме и дату обновления цен.
 8. Пользователь выбирает конкретный вариант узла в конкретной фирме.
 9. Пользователь нажимает кнопку «Запомнить вариант».
 10. Система заносит выбранный вариант узла комплектации с выбранной фирмой в память.

Альтернативные сценарии ба. В каталоге нет узла, удовлетворяющего запросу пользователя

1. Система сообщает пользователю о параметре(-ах), значение которого(-ых) не позволяет выбрать товар из каталога.
2. Пользователь задает новые значения параметров.
- 7а. Пользователь желает прочесть отзывы о данном варианте узла.
 1. Пользователь нажимает кнопку «Отзывы».
 2. Система выводит список отзывов на данный узел.
- 8а. В выбранной пользователем фирме нет данного узла.
 1. Система сообщает, что данный узел отсутствует в выбранной фирме и выводит дату обновления информации о комплектующих в данной фирме.

Задачи процесса сопровождения

Цели: применение системы стандартов в области ПО, методов функционального и системного моделирование, построение структурных моделей бизнеса.

Средства выполнения задания: MS Visio, средства пакета MS Office и Design/IDEF 3.5, Visible Analyst, VPwin 2-4.1.

Изучить теоретический материал, дать письменные ответы на контрольные вопросы и выполнить практическое задание.

Практическое задание

1. Выполнить структурный анализ предметной области и проектирование системы одним из структурно-функциональных подходов для задач по вариантам;
2. Описать структуру внешних и внутренних данных в IDEF1X;
3. Оформить отчет по результатам.

Цель выполнения практического задания: Понять, какие функции должны быть включены в процесс «Процесс», как эти функции взаимосвязаны между собой.

В результате структурного анализа должны быть построены диаграммы потоков данных и IDEF0-модель. Разрешается изменить указанные цели и задачи при условии обязательного согласования изменений с преподавателем. Декомпозиция процессов (функций, работ) выполняется студентом до третьего уровня.

Вариант 1. Компания предоставляет своим клиентам услуги согласно заключаемым договорам. Вид услуги регламентирован законодательством и уставом компании. В составлении договора участвует юрист компании.

Цель: определить действия, необходимые для привлечения клиента и заключения с ним договора на оказание услуг. Точка зрения: Отдел рекламы.

Вариант 2. Отдел регистратуры поликлиники в конце каждого рабочего дня предоставляет бухгалтерии сводный отчет по приему пациентов врачами. Запись на прием пациентов осуществляется согласно расписанию каждого врача.

Цель: определить действия, необходимые для ведения записи и учета принятых пациентов в поликлинике. Точка зрения: Регистратура.

Вариант 3. Отдел продаж крупной компьютерной компании осуществляет продажу компакт-дисков с учебными курсами как по безналу, так и при оплате наличными клиентам. Существует система скидок по типу клиента. Существует отчетная документация. Клиент обращается в отдел как лично, так и через Internet. Цель: определить действия, необходимые для составления отчета по выполненным продажам за отчетный период. Точка зрения: Отдел продаж.

Вариант 4. Авиационный завод выпускает летательные аппараты различной конфигурации по государственным заказам и для частных лиц и компаний. Завод осуществляет гарантийное техническое обслуживание и сопровождение.

Цель: определить действия отдела связи с общественностью, необходимые для привлечения клиентов. Точка зрения: Отдел связи с общественностью.

Вариант 5. Научно-исследовательский отдел пищекомбината отчитывается по расходным материалам. По разработанным рецептурам технолог отчитывается каждую неделю с демонстрацией продукции перед дирекцией.

Цель: определить действия, необходимые для составления отчета по расходу материалов при разработке новой рецептуры. Точка зрения: Научно-исследовательский отдел.

Вариант 6. Книжный магазин принимает заказы от населения на издания с условием частичной предоплаты (от 10% до 50%). На формирование объема заказа в издательстве учитывается спрос издания среди населения по категории.

Цель: определить действия, необходимые для составления отчета по оформлению заказов за отчетный

период. Точка зрения: Отдел заказов.

Вариант 7. Библиотека вуза включает издания различных лет. Каждые два года выполняется ревизия и списание пособий с учетом потребности в учебном процессе. Для преподавателей кафедр публикуются сводки и отчеты.

Цель: определить действия, необходимые для составления сводок и отчетов по наличию пособий и их востребованности среди студентов. Точка зрения: Отдел комплектования.

Вариант 8. Компьютерная компания предоставляет населению помимо Интернет–услуг сопровождение ПО и обслуживание ВТ. Для привлечения клиентов используется система скидок. За отчетный период клиентам выставляется счет с описанием оказанных услуг. Цель: определить действия, необходимые для выставления счета клиенту. Точка зрения: Бухгалтерия.

Вариант 9. Художественная галерея периодически проводит выставки персоналий и распродажи художественных произведений. Информация как публикуется, так и рассылается постоянным клиентам галереи.

Цель: определить действия, необходимые для оповещения потенциальных посетителей выставки. Точка зрения: Организатор выставки.

Вариант 10. Редколлегия газеты для оплаты соборов использует систему учета публикаций и рейтинга статей среди читателей и критиков. Раз в месяц подводятся итоги и выплачиваются премии. Цель: определить действия, необходимые для ведения рейтинга и назначения премий. Точка зрения: Редколлегия.

Структура отчета.

1. Постановка задачи.
2. Основные элементы модели процесса.

– название проекта;

– цель проекта;

– точка зрения;

– инструментарий;

– список данных;

1. Дерево функций.
2. Словарь.
3. Диаграммы процессов.

2. Диаграмма потоков данных

Цели: рассмотрение систем международных стандартов в области ПО и их применения при построении диаграмм описания документооборота и обработки информации.

Теоретический материал

Для дополнения к диаграммам IDEF0 для описания документооборота и обработки информации используются диаграммы потоков данных.

Диаграмма потоков данных (Data Flow Diagram – DFD) – информационная модель, основными компонентами которой являются потоки данных, переносящие информацию от одного модуля к другому.

DFD – основной инструмент структурного анализа. DFD позволяет определять трансформационные процессы системы, совокупность данных (хранение) или материалов, которыми система управляет, потоки данных или материалов между процессами, хранилищами и внешним миром.

Логическая DFD показывает внешние по отношению к системе источник и потоки данных, идентифицирует логические функции (процессы) и группы элементов данных, определяет хранилища (накопители) данных, то есть диаграммы потоков данных ориентированы на задачи обработки данных.

Главная цель построения диаграмм потоков данных – раскрыть, как каждый процесс преобразует свои

входные данные в выходные, указать отношения между этими процессами. В основе DFD– технологии лежат три группы средств моделирования: функциональные диаграммы (диаграммы потоков данных – DFD), диаграммы «сущность-связь» (ER–диаграммы) и диаграммы моделирования поведения систем (STD). Диаграммы потоков данных просты, наглядны и понятны (см. рис. 2.1).

В качестве примера представлен верхний уровень функциональной модели взаимодействия студента и документоведа деканата при составлении выписки изученных дисциплин из учетной карточки студента.

Для построения диаграмм потоков данных используются нотации Гейна-Сарсона (Gane-Sarson) или Йордана (Yourdon).

Основные символы данных нотаций представлены в табл. 1 согласно принятым обозначениям, используемым в пакете MS Visio.

Моделью системы будет совокупность диаграмм потоков данных, построенным с различными уровнями абстрагирования (см. рис. 1.2), описывающие асинхронный процесс преобразования информации от её ввода в систему до выдачи пользователю.

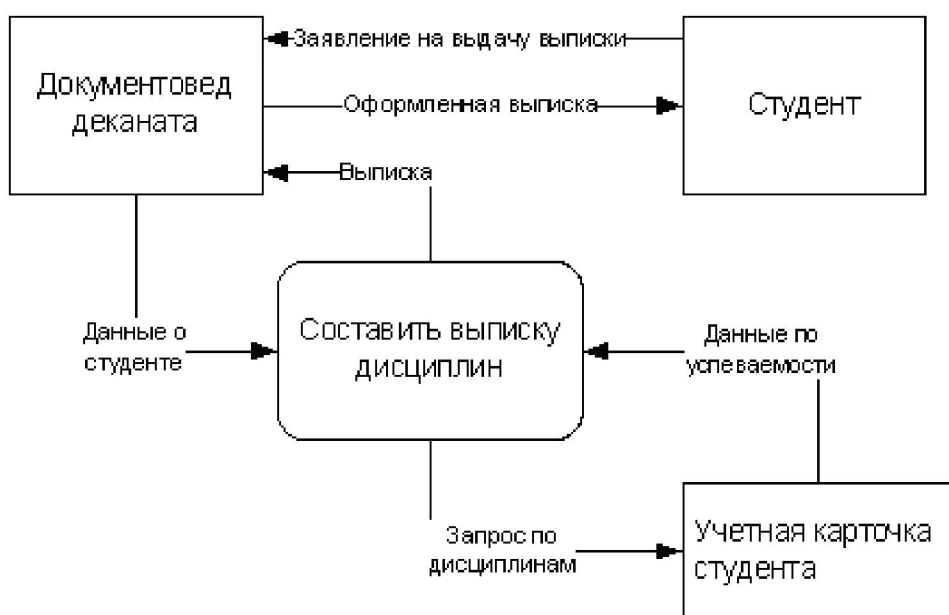






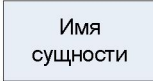



Рис. 2.1. Пример диаграммы потоков данных

Потоки данных – механизмы, используемые для моделирования передачи информации (материальных потоков) из одной в другую часть системы. Хранилище позволяет определять данные, хранимые в памяти между процессами. Содержащиеся в нем данные могут использоваться в любой момент и выбираться в любом порядке. Внешняя сущность представляет сущность вне системы, являющуюся источником или приемником данных, не обрабатывающих их. Внешняя сущность на диаграмме может использоваться несколько раз.

Таблица 1. Основные символы DFD

Название	Нотация Йордана	Нотация Гейна-Сарсона
Поток данных		
Процесс		
Хранилище (Журналы, БД)		
Внешняя сущность		
MS Visio	Flowchart/Data Flow Diagram	Software/Data Flow Model Diagram

Декомпозиция DFD осуществляется на основе процессов: каждый процесс может раскрываться с помощью DFD нижнего уровня. При детализации должны выполняться правила:

Правило балансировки – в качестве внешних источников и приемников данных могут быть только те компоненты, с которыми детализируемая система (процесс) имеет информационную связь;

Правило нумерации – должна поддерживаться иерархическая нумерация процессов.

Процесс детализации продолжается до тех пор, пока процессы могут быть эффективно описаны с помощью коротких спецификаций процессов.

При анализе составления модели «AS-IS» (КАК-ЕСТЬ) процесса составления выписки из учетной карточки студента было принято решение описать с помощью IDEF3 листовую функцию Выполнить сбор данных по успеваемости для изучения порядка обращения к системам хранения. Для этого были определены составляющие процессы, используемые хранилища и типы связей.

В качестве хранилищ данных выступают учетная карточка студента, учебный план потока и зачетно-экзаменационные ведомости. Внешними сущностями для системы формирования выписки будут студент, источник заявления на выписку, и документовед, выполняющий основные процессы.

Результат построения диаграммы представлен на рис. 2.2.

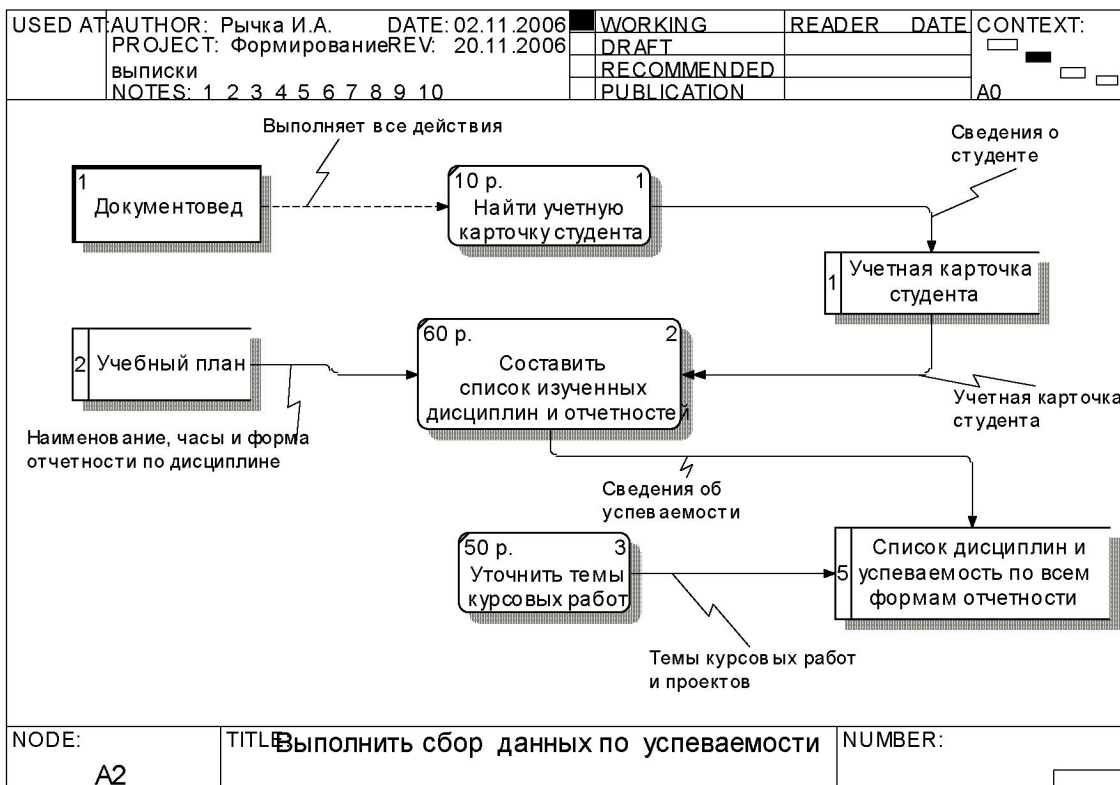


Рис. 2.2. Схема бизнес-процесса «Выполнить сбор данных по успеваемости» в стандарте DFD

Нормализованная диаграмма «сущность-связь» в нотации IDEF1X может быть построена с помощью одного из пакетов IDEF3.5 (Meta Software Corporation) или ALLFusion ERwin (Computer Associates International, Inc.).

Исходя из существующей системы хранения данных об успеваемости, аналитиком предложена даталогическая модель данных, представленная на рис. 2.2.

Вопросы для самоконтроля

1. Каково назначение диаграмм потоков данных?
2. Какая система обозначений используется в пакете BPwin для построения диаграмм потоков данных?
3. Какие символы DFD используются?

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя
Отлично	Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Практическое задание для формирования «ПК-8.3»

Задание

1. Следуя указаниям, создайте программу, которая запрашивает у нового сотрудника имя, фамилию и

дату рождения. Хранить эту информацию требуется в свойствах нового класса с именем Person, и создайте метод класса, который будет вычислять текущий возраст нового сотрудника.

2. Изменить проект MyPersonClass, добавив в модуль класса Person второй класс, определенный пользователем. Этот новый класс с именем Teacher будет наследовать от класса Person свойство FirstName, свойство LastName и метод Age, и будет добавлять свойство с именем Grade, в которое будет записываться уровень, на котором обучает новый учитель.

3. Подготовить тест-кейсы для проверки работоспособности проекта

Критерии оценки выполнения задания

Оценка	Критерии оценивания
Неудовлетворительно	Работа выполнена не полностью и объем выполненной части работы не позволяет сделать правильных выводов
Удовлетворительно	Работа выполнена не полностью, но не менее 50% объема, что позволяет получить правильные результаты и выводы; в ходе проведения работы были допущены ошибки
Хорошо	Работа выполнена в полном объеме с соблюдением необходимой последовательности действий, но допущена одна ошибка или не более двух недочетов и обучающийся может их исправить самостоятельно или с небольшой помощью преподавателя
Отлично	Работа выполнена в полном объеме без ошибок с соблюдением необходимой последовательности действий

Вопросы для проведения промежуточной аттестации по итогам освоения дисциплины

Тема 1. Особенности разработки сложных программных средств

1. Особенности программных средств и задач их разработки.
2. Классификация программных средств.
3. Жизненный цикл программных средств.

Тема 2. Создание сложных программных средств

4. Модели ЖЦ ПС.
5. Итерационные модели.
6. Спецификация.
7. Проектирование и реализация.
8. Эксплуатация и сопровождение программы.
9. Автоматизированные средства разработки.
10. Разработка объектно-ориентированного программного обеспечения.

Тема 3. Технологический процесс разработки программ и программной документации

11. Стадии разработки программ и программной документации.
12. Сопровождающая документация. Основные требования к содержанию документации.
13. Единая система программной документации (ЕСПД).
14. Международные стандарты ИСО по разработке и использованию программного обеспечения, документации, интернет - приложений.

Тема 4. Проектирование и разработка программных средств

15. Неавтоматизированное и автоматизированное проектирование программ.
16. Принцип системного проектирования. Нисходящее проектирование. Модульное проектирование. Структурное программирование. Функционально-ориентированные методы и методы структурирования данных.
17. Информационное моделирование предметной области и его составляющие. Технологии информационного моделирования.
18. Сущность объектно-ориентированного подхода к проектированию программных продуктов.

19. Объектно-ориентированный анализ предметной области и объектно-ориентированное проектирование.
20. Цели структуризации программных продуктов. Типовая структура программного продукта.
21. Головной, управляющий модуль, рабочие и сервисные модули.
22. Комбинирование языков программирования в рамках одной задачи.
23. Типовая структура программного модуля. Порядок разработки программного модуля.
24. Методы восходящей и нисходящей разработки структуры программы. Конструктивный и архитектурный подходы к разработке программы.

Тема 5. Технологические средства разработки программного обеспечения

25. CASE-системы, системы ускоренной разработки приложений.
26. Системный анализ.
27. Принципы объектно-ориентированного анализа и их обсуждение.
28. Рабочие продукты, методологии и средства анализа и проектирования.
29. Среды программирования.

Тема 6. Объектное моделирование. Стандарт UML

30. Язык объектного моделирования UML.
31. Структура системы, декомпозиция, иерархия элементов.
32. Процессы в системе и потоки информации. Исследование действий. Построение моделей доменов и подсистем, связей и взаимодействия подсистем, взаимодействия объектов, событий, процессов, потоков данных, действий. Описание классов и их взаимосвязей.
33. Динамика поведения объектов, диаграммы перехода состояний.
34. Диаграммы объектов. Видимость и синхронизация объектов, временные диаграммы.
35. Диаграмма процессов. Обработка исключительных ситуаций.

Тема 7. Разработка программного обеспечения

36. Инструментальные средства операционной системы.
37. Библиотеки операционной системы и их использование в разработке программ.
38. Визуальное проектирование программ.
39. Визуальные среды. Инструменты Visual Studio.
40. Подбор и редактирование компонент, разработка компонент.
41. Построение интерфейса программы. Методы построения интерфейса программного обеспечения.

Тема 8. Отладка, тестирование и сопровождение программ

42. Классификация ошибок программного обеспечения. Основные пути и методы борьбы с ошибками программного обеспечения.
43. Обнаружение и локализация ошибок ввода и обработки данных.
44. Составляющие процесса отладки. Принципы и виды отладок. Автономная и комплексная отладки программ.
45. Методы отладки.
46. Средства отладки.
47. Сущность и необходимость тестирования программного обеспечения.
48. Различие между тестированием и отладкой программного обеспечения.
49. Стадии тестирования. Виды тестовых проверок.
50. Объекты тестирования и категории тестов.
51. Виды тестирования.
52. Средства тестирования. Ручное и автоматизированное тестирование.
53. Методы и инструментальные средства тестирования.
54. Сопровождение программных продуктов, внесение изменений, обеспечение надежности при эксплуатации.
55. Необходимая документация и предпродажная подготовка программных средств.

Тема 9. Обеспечение качества программного средства

56. Эффективность и оптимизация программ. Основные критерии эффективности программного средства.

- 57. Оптимизация программ на этапе отладки.
- 58. Принципы обеспечения показателей качества программного средства.
- 59. Стандарты, регламентирующие качество программных средств.

Тема 10. Защита программ

- 60. Методы защиты программного средства.
- 61. Защита программного средства от копирования.
- 62. Системы разграничения доступа.
- 63. Криптографические методы защиты программного средства.
- 64. Аппаратные средства защиты программного средства.
- 65. Патентная защита. Лицензионные соглашения.

Уровни и критерии итоговой оценки результатов освоения дисциплины

	Критерии оценивания	Итоговая оценка
Уровень 1. Недостаточный	Незнание значительной части программного материала, неумение даже с помощью преподавателя сформулировать правильные ответы на задаваемые вопросы, невыполнение практических заданий	Неудовлетворительно/Незачтено
Уровень 2. Базовый	Знание только основного материала, допустимы неточности в ответе на вопросы, нарушение логической последовательности в изложении программного материала, затруднения при решении практических задач	Удовлетворительно/зачтено
Уровень 3. Повышенный	Твердые знания программного материала, допустимые несущественные неточности при ответе на вопросы, нарушение логической последовательности в изложении программного материала, затруднения при решении практических задач	Хорошо/зачтено
Уровень 4. Продвинутый	Глубокое освоение программного материала, логически стройное его изложение, умение связать теорию с возможностью ее применения на практике, свободное решение задач и обоснование принятого решения	Отлично/зачтено

7. Ресурсное обеспечение дисциплины

Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства	<ol style="list-style-type: none"> 1. Microsoft Windows (лицензионное программное обеспечение) 2. Microsoft Office (лицензионное программное обеспечение) 3. Google Chrome (свободно распространяемое программное обеспечение) 4. Kaspersky Endpoint Security (лицензионное программное обеспечение) 5. AnyLogic (свободно распространяемое программное обеспечение) 6. ArgoUML (свободно распространяемое программное обеспечение) 7. ARIS EXPRESS (свободно распространяемое программное обеспечение) 8. Erwin (свободно распространяемое программное обеспечение) 9. Inkscape (свободно распространяемое программное обеспечение) 10. iTALC (свободно распространяемое программное обеспечение) 11. Maxima (свободно распространяемое программное обеспечение) 12. Microsoft SQL Server Management Studio (лицензионное программное обеспечение) 13. Microsoft Visio (лицензионное программное обеспечение) 14. Microsoft Visual Studio (лицензионное программное обеспечение) 15. MPLAB (свободно распространяемое программное обеспечение) 16. Notepad++ (свободно распространяемое программное обеспечение) 17. Oracle VM VirtualBox (свободно распространяемое программное обеспечение) 18. Paint .NET (свободно распространяемое программное обеспечение) 19. SciLab (свободно распространяемое программное обеспечение) 20. WinAsm (свободно распространяемое программное обеспечение) 21. Консультант+ (лицензионное программное обеспечение отечественного производства) 22. GNS 3 (свободно распространяемое программное обеспечение) 23. Спутник (свободно распространяемое программное обеспечение отечественного производства) 24. Microsoft Project (лицензионное программное обеспечение) 25. «Антиплагиат.ВУЗ» (лицензионное программное обеспечение)
Современные профессиональные базы данных	<ol style="list-style-type: none"> 1. Консультант+ (лицензионное программное обеспечение отечественного производства) 2. http://www.garant.ru (ресурсы открытого доступа)
Информационные справочные системы	<ol style="list-style-type: none"> 1. https://elibrary.ru - Научная электронная библиотека eLIBRARY.RU (ресурсы открытого доступа) 2. https://www.rsl.ru - Российская Государственная Библиотека (ресурсы открытого доступа) 3. https://link.springer.com - Международная реферативная база данных научных изданий Springerlink (ресурсы открытого доступа) 4. https://zbmath.org - Международная реферативная база данных научных изданий zbMATH (ресурсы открытого доступа)
Интернет-ресурсы	<ol style="list-style-type: none"> 1. http://window.edu.ru - Информационная система "Единое окно доступа к образовательным ресурсам" 2. https://openedu.ru - «Национальная платформа открытого образования» (ресурсы открытого доступа)

Материально-техническое обеспечение	<p>Учебные аудитории для проведения:</p> <p>занятий лекционного типа, обеспеченные наборами демонстрационного оборудования и учебно-наглядных пособий, обеспечивающих тематические иллюстрации, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду организации, помещения для хранения и профилактического обслуживания учебного оборудования.</p> <p>Лаборатории и кабинеты:</p> <p>1. Учебная аудитория Лаборатория информатики Компьютерный класс , включая оборудование: Комплекты учебной мебели, демонстрационное оборудование – проектор и компьютер, учебно-наглядные пособия, обеспечивающие тематические иллюстрации, доска, персональные компьютеры.</p>
-------------------------------------	---

8. Учебно-методические материалы

№	Автор	Название	Издательство	Год издания	Вид издания	Кол-во в библиотеке	Адрес электронного ресурса	Вид доступа
1	2	3	4	5	6	7	8	9
9.1 Основная литература								
9.1.1	Липаев В.В.	Сертификация программных средств	СИНТЕГ	2010	учебник	-	http://www.iprbookshop.ru/27299.html	по логину и паролю
9.1.2	Синицын С.В. Хлытчиев О.И.	Основы разработки программного обеспечения на примере языка С	Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа	2021	учебник	-	http://www.iprbookshop.ru/102039.html	по логину и паролю
9.1.3	Зубкова Т.М.	Технология разработки программного обеспечения	Оренбургский государственный университет, ЭБС АСВ	2017	учебное пособие	-	http://www.iprbookshop.ru/78846.html	по логину и паролю
9.2 Дополнительная литература								
9.2.1	Липаев В.В.	Сопровождение и управление конфигурацией сложных программных средств	СИНТЕГ	2006	монография	-	http://www.iprbookshop.ru/27300.html	по логину и паролю
9.2.2	сост. Добролюбов В.В. Андрюков А.А. Максименко В.Н.	Методические указания и задание на контрольную работу по дисциплине Технологии разработки программных комплексов и CASE-средства	Московский технический университет связи и информатики	2016	учебно-методическое пособие	-	http://www.iprbookshop.ru/63365.html	по логину и паролю

9. Особенности организации образовательной деятельности для лиц с ограниченными возможностями здоровья

В МФЮА созданы специальные условия для получения высшего образования по образовательным программам обучающимися с ограниченными возможностями здоровья (ОВЗ).

Для перемещения инвалидов и лиц с ограниченными возможностями здоровья в МФЮА созданы специальные условия для беспрепятственного доступа в учебные помещения и другие помещения, а

также их пребывания в указанных помещениях с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

При получении образования обучающимся с ограниченными возможностями здоровья при необходимости предоставляются бесплатно специальные учебники и учебные пособия, иная учебная литература. Также имеется возможность предоставления услуг ассистента, оказывающего обучающимся с ограниченными возможностями здоровья необходимую техническую помощь, в том числе услуг сурдопереводчиков и тифлосурдопереводчиков.

Получение доступного и качественного высшего образования лицами с ограниченными возможностями здоровья обеспечено путем создания в университете комплекса необходимых условий обучения для данной категории обучающихся. Информация о специальных условиях, созданных для обучающихся с ограниченными возможностями здоровья, размещена на сайте университета (<http://www.mfua.ru/sveden/objects/#objects>).

Для обучения инвалидов и лиц с ОВЗ, имеющих нарушения опорно-двигательного аппарата обеспечиваются и совершенствуются материально-технические условия беспрепятственного доступа в учебные помещения, столовую, туалетные, другие помещения, условия их пребывания в указанных помещениях (наличие пандусов, поручней, расширенных дверных проемов и др.).

Для адаптации к восприятию обучающимися инвалидами и лицами с ОВЗ с нарушенным слухом справочного, учебного материала, предусмотренного образовательной программой по выбранным направлениям подготовки, обеспечиваются следующие условия:

- для лучшей ориентации в аудитории, применяются сигналы, оповещающие о начале и конце занятия (слово «звонок» пишется на доске);
- внимание слабослышащего обучающегося привлекается педагогом жестом (на плечо кладется рука, осуществляется нерезкое похлопывание);
- разговаривая с обучающимся, педагог смотрит на него, говорит ясно, короткими предложениями, обеспечивая возможность чтения по губам.

Компенсация затруднений речевого и интеллектуального развития слабослышащих инвалидов и лиц с ОВЗ проводится за счет:

- использования схем, диаграмм, рисунков, компьютерных презентаций с гиперссылками, комментирующими отдельные компоненты изображения;
- регулярного применения упражнений на графическое выделение существенных признаков предметов и явлений;
- обеспечения возможности для обучающегося получить адресную консультацию по электронной почте по мере необходимости.

Для адаптации к восприятию инвалидами и лицами с ОВЗ с нарушениями зрения справочного, учебного, просветительского материала, предусмотренного образовательной программой МФЮА по выбранной специальности, обеспечиваются следующие условия:

- ведется адаптация официального сайта в сети Интернет с учетом особых потребностей инвалидов по зрению, обеспечивается наличие крупношрифтовой справочной информации о расписании учебных занятий;
- в начале учебного года обучающиеся несколько раз проводятся по зданию МФЮА для запоминания месторасположения кабинетов, помещений, которыми они будут пользоваться;
- педагог, его собеседники, присутствующие представляются обучающимся, каждый раз называется тот, к кому педагог обращается;
- действия, жесты, перемещения педагога коротко и ясно комментируются;
- печатная информация предоставляется крупным шрифтом (от 18 пунктов), тотально озвучивается;
- обеспечивается необходимый уровень освещенности помещений;
- предоставляется возможность использовать компьютеры во время занятий и право записи объяснения на диктофон (по желанию обучающегося).

Форма проведения текущей и промежуточной аттестации для обучающихся с ОВЗ определяется преподавателем в соответствии с учебным планом. При необходимости обучающемуся с ОВЗ с учетом его индивидуальных психофизических особенностей дается возможность пройти промежуточную аттестацию устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п., либо предоставляется дополнительное время для подготовки ответа.

Год начала подготовки студентов - 2022