

Федеральное агентство по образованию
ГОУ ВПО «Кемеровский государственный университет»
Кафедра ЮНЕСКО по Новым информационным технологиям

«УТВЕРЖДАЮ»
Декан математического факультета

_____ Н.Н. Данилов
« » _____ 20 г.

РАБОЧАЯ ПРОГРАММА

по курсу *«Технологии разработки программного обеспечения»*

Специальность: **010503.65 «Математическое обеспечение и администрирование информационных систем»**, ДС.Ф.04

Факультет: *Математический*

Курс:	III		
Семестр:	5		
Лекции:	36	Экзамен:	5 семестр
практические занятия:			
лабораторные занятия:	36		
коллоквиум:	3		
самостоятельные занятия:	57		
Всего часов:	132		
Составитель: доцент Гудов А.М.			

Кемерово, 2008

Рабочая программа дисциплины федерального компонента цикла ДС.Ф.04 «Технологии разработки программного обеспечения» составлена на основании Государственного образовательного стандарта высшего профессионального образования второго поколения по специальности 010503 «Математическое обеспечение и администрирование информационных систем», утвержденном 10 марта 2000 г., № 72 МЖД/ СП.

Рабочая программа обсуждена на заседании кафедры

Протокол № _____ от « _____ » _____ 200__ г.

Зав.кафедрой _____ К.Е. Афанасьев

Одобрено методической комиссией

Протокол № _____ от « _____ » _____ 200__ г.

Председатель _____

1. Пояснительная записка

Рабочая программа полностью соответствует Государственному образовательному стандарту высшего профессионального образования второго поколения по специальности 010503 «Математическое обеспечение и администрирование информационных систем».

Актуальность и значимость дисциплины. Дисциплина является одной из основных в цикле общих дисциплин специальности, в рамках которой изучаются основные понятия, модели и технологии создания программных систем, организация и реализация проектов по производству программного обеспечения.

Цель и задачи изучения дисциплины. Главная цель преподавания курса - освоение базовых знаний по вопросам проектирования и разработки информационных систем.

Объектами изучения в данной дисциплине являются: технологии проектирования, модели и методы поддержки жизненного цикла программного обеспечения; средства и методы создания и реализации проектов по созданию программных систем.

Основными задачами изучения дисциплины являются:

- знакомство с основными этапами жизненного цикла программного обеспечения;
- знакомство с технологиями функционального и объектно-ориентированного проектирования;
- приобретение навыков работы со средствами автоматизации разработки ПО;
- приобретение навыков по созданию программного средства с использованием базы данных;
- подготовка студентов к изучению других дисциплин по информационным технологиям.

Место дисциплины в профессиональной подготовке специалистов. Курс занимает особое место в учебном плане среди дисциплин специальности факультета по его значению. Вместе с курсами по программированию, курс «Технологии разработки программного обеспечения» составляет основу образования студента в части информационных технологий. Курс рассчитан на студентов-математиков, имеющих подготовку по математике и информатике в объеме программы средней школы. В течение преподавания курса предполагается, что студенты знакомы с основными понятиями процедурного и объектно-ориентированного программирования, логики, информатики, которые читаются на факультете перед изучением данной дисциплины.

Требования ГОС ВПО к обязательному минимуму содержания дисциплины.

Индекс	Наименование дисциплин и их основные разделы	Всего часов
ДС	ДИСЦИПЛИНЫ СПЕЦИАЛЬНОСТИ	
<i>ДС.Ф.00</i>	<i>Федеральный компонент</i>	
ДС.Ф.04	ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ Программные продукты; процесс производства: методы, технология и инструментальные средства; тестирование и отладка; документирование; проектирование программного обеспечения; абстрактные структуры данных; способы эффективного хранения и обработки; технологический цикл разработки программных систем; коллективная работа по созданию программ; организация труда в коллективе и инструментальные средства поддержки; автоматизация проектирования программных продуктов; принципы построения, структура и технология использования САПР ПО.	

Структура учебной дисциплины. Дисциплина включает следующие разделы: технологии, модели и процессы создания ПО, основы создания ПО, разработка требований к ПО, управление проектами по созданию и внедрению ПО, управление персоналом при реализации проектов, оценка стоимости программного продукта, управление качеством созданных программных систем, создание спецификации программной системы с использованием элементов объектного проектирования.

Особенности изучения дисциплины. Особенностью курса является его «разделение» на два больших блока: изучение теоретического материала на лекционных занятиях и получения навыков проектирования и разработки приложения с использованием CASE-систем на лабораторных занятиях и в рамках самостоятельной работы студента.

Форма организации занятий по дисциплине. По дисциплине читаются лекции в течение одного семестра по два часа в неделю. В течение одного семестра лабораторные занятия в объеме двух часов в неделю в компьютерном классе. Кроме того, студенту предлагается выполнить семестровое задание в рамках самостоятельной работы. Один раз в семестре проводится коллоквиум по дисциплине.

Требования к уровню усвоения содержания материала. В результате изучения курса студенты должны усвоить основные теоретические и практические вопросы, определенные содержанием дисциплины, научиться пользоваться полученными знаниями в смежных предметах, научиться применять технологии, методики, учебную и методическую литературу для решения поставленных задач по созданию программных средств.

Виды контроля знаний и их отчетности. По разделам основной части курса предусмотрены самостоятельные задания, семестровые задания, для выполнения которых требуются элементы исследовательской работы, коллоквиум, тест. По итогам изучения курса предусмотрен экзамен.

Критерии оценки знаний студентов. Предусмотрена рейтинговая система оценки всех видов деятельности. Максимальное число баллов – 100. Каждый вид деятельности оценивается следующим образом:

- посещение лекций – 1 балл каждое занятие (максимально 18 баллов);
- выполнение самостоятельных заданий – 2 балла каждое (максимально 50 баллов);
- семестровые задания – 2 балла (сдано в срок) + 10-ти бальная оценка за выполнение работы;
- коллоквиум – максимально 20 баллов (написаны 3 теоретических вопроса и решено 2 задачи).

Студент, набравший 50 баллов и более, выполнивший семестровое задание получает допуск к экзамену.

Экзамен проходит в 2 этапа: тестирование и опрос по дополнительным вопросам (если есть пропуски лекций и лабораторных занятий).

При выставлении оценки экзамена учитываются следующие параметры:

1. Результат теста (100 баллов максимум, ниже 50 баллов – студент получает оценку «неудовлетворительно»);
2. Работа студента в семестре (максимально 100 баллов, ниже 50 баллов студент не получает допуск к экзамену).

Итоговая оценка экзамена выставляется на основании 2 параметров указанных выше. Максимальное число баллов 200. Оценка за экзамен: «отлично» – от 170 до 200 баллов; «хорошо» - от 120 до 169 баллов; «удовлетворительно» - от 80 – 119 баллов.

2. Тематический план

№	Название и содержание разделов, тем, модулей	Объем часов				Формы контроля
		Общий	Аудиторная работа		Самостоятельная работа (в часах)	
			лекции	практические		
1	2	3	4	5	6	7
Очная форма обучения						
1	Технологии, модели и процессы создания ПО	12	6	4	2	Отчет по лабораторной работе
2.	Основы создания ПО	8	6		2	Отчет по лабораторной работе

3.	Разработка требований к ПО	10	4	4	2	Отчет по лабораторной работе
4.	Управление проектами по созданию и внедрению ПО	10	4	2	4	Отчет по лабораторной работе
5.	Управление персоналом при реализации проектов	12	4	2	6	Отчет по лабораторной работе
6.	коллоквиум	10	2	2	6	Задания для коллоквиума
7.	Оценка стоимости программного продукта	13	4	6	3	Отчет по лабораторной работе
8.	Управление качеством созданных программных систем	8	4	2	2	Отчет по лабораторной работе
9.	Создание проекта программной системы с использованием элементов объектного проектирования	46	2	14	30	Отчет о семестровой работе
	Итого:	132	36	36	57	Экзамен

3. Содержание дисциплины

3.1 Содержание теоретического курса

1. Технологии, модели и процессы создания ПО.

Терминология, Процессы создания ПО. Методы создания ПО. Структуры затрат на создание ПО. Основные вопросы, встающие перед специалистами по созданию ПО.

2. Основы создания ПО.

Базовые процессы создания ПО. Модели создания ПО. Спецификация ПО. Реализация ПО. Аттестация ПО. Эволюция ПО. CASE-средства автоматизации процессов создания ПО.

3. Разработка требований к ПО.

Анализ осуществимости. Формирование и анализ требований к ПО. Аттестация требований. Управление требованиями. Управление изменением требований. Модели окружения. Поведенческие модели. Модели потоков данных. Модели конечных автоматов. Модели данных. CASE-средства проектирования.

4. Реализация ПО.

Архитектурное проектирование. Структурирование системы. Модели управления. Модульная декомпозиция. Проблемно-зависимые архитектуры. Проектирование с повторным использованием кода. Проектирование интерфейса пользователя.

5. Управление проектами по созданию и внедрению ПО.

Процессы управления. Планирование проекта. Контрольные отметки. График работ. Временные и сетевые диаграммы. Управление рисками.

6. Управление персоналом при реализации проектов.

«Людской» капитал. Организация человеческой памяти. Решение задач. Мотивация. Групповая работа. Создание команды разработчиков. Сплоченность команды. Организация и общение в группе. Подбор и сохранение персонала.

7. Оценка стоимости программного продукта.

Понятие стоимости продукта. Параметры, используемые при оценке проекта. Факторы, влияющие на стоимость программного продукта. Методы оценивания. Моделирование стоимости.

8. Управление качеством созданных программных систем.

Понятие качество программного продукта. Процесс управления качеством. Стандарты для управления качеством. Советы «менеджеру по качеству». Стандарты на техническую документацию. План обеспечения качества. Контроль качества. Проверки качества. Измерение показателей. Показатели качества программного продукта.

9. Создание проекта программной системы с использованием элементов объектного проектирования

Выбор программного средства (ПС) для реализации. Создание спецификации (технического задания) ПС. Создание основной проектной документации на ПС.

3.2 Содержание лабораторных занятий

1. Разработка описания и анализ информационной системы

Лабораторная работа направлена на ознакомление с процессом описания информационной системы и получение навыков по использованию основных методов анализа ИС.

Требования к результатам выполнения лабораторной работы:

- наличие описания информационной системы;
- наличие заключения о возможности реализации проекта, содержащего рекомендации относительно разработки системы, базовые предложе-

2. Разработка требований к информационной системе

Лабораторная работа направлена на ознакомление с процессом разработки требований к информационной системе и составления технического задания на разработку программного обеспечения, получение навыков по использованию основных методов формирования и анализа требований.

Требования к результатам выполнения лабораторной работы:

- наличие диаграммы идентификации точек зрения и диаграммы иерархии точек зрения;
- наличие сценариев событий (последовательности действий);
- наличие пользовательских требований, четко описывающих будущий функционал системы;
- наличие системных требований, включающих требования к структуре, программному интерфейсу, технологиям разработки, общие требования к системе (наджность, масштабируемость, распределённость, модульность, безопасность, открытость, удобство пользования и т.д.);
- наличие составленного технического задания.

3. Методология функционального моделирования

Лабораторная работа направлена на ознакомление с методологиями функционального моделирования IDEF0 и IDEF3, получение навыков по применению данных методологий для построения функциональных моделей на основании требований к информационной системе.

Требования к результатам выполнения лабораторного практикума:

- модель должна отражать весь указанный в описании функционал, а также чётко отражать существующие потоки данных и описывать правила их движения;
- наличие в модели не менее трёх уровней;
- не менее двух уровней декомпозиции в стандарте IDEF0 (контекстная диаграмма + диаграммы A0);
- на диаграмме 1-го уровня (A0) не менее 4-х функциональных блоков;
- на диаграмме 2-го и далее уровнях должна быть декомпозиция в стандарте IDEF3, на каждой диаграмме не менее 2-х функциональных блоков.

4. Методология объектно-ориентированного моделирования

Лабораторная работа направлена на ознакомление с основными элементами определения, представления, проектирования и моделирования программных систем с помощью языка UML, получение навыков по применению данных элементов для построения объектно-ориентированных моделей ИС на основании требований.

Требования к результатам выполнения лабораторной работы:

- модель системы должна содержать: диаграмму вариантов использования; диаграммы взаимодействия для каждого варианта использования; диаграмму классов, позволяющая реализовать весь описанный функционал ИС; объединенную диаграмму компонентов и размещения
- для классов указать стереотипы;
- в зависимости от варианта задания диаграмма размещения должна показывать расположение компонентов в распределенном приложении или связи между встроенным процессором и устройствами.

5. Методология управление проектами

Лабораторная работа направлена на ознакомление с основными понятиями методологии управления проектами, получение навыков по применению данных понятий при построении плана проекта, построения графика работ, распределения исполнителей, управления рисками.

Требования к результатам выполнения лабораторной работы:

- Построить модель управления проектом. Модель включает:
 - определение всех этапов проекта, зависимых этапов, определение длительности этапов;
 - построение на основе полученных данных сетевой и временной диаграмм;
 - построение диаграммы распределения работников по этапам;
- при определении этапа указывается его название – отражающее суть этапа (например, определение пользовательских требований, проектирование интерфейса и т.д.);
- этапов должно быть не менее 7, срок реализации проекта – пол года с 1.06.2008 по 31.12.2008;
- в проекте задействовано 6 человек персонала (фамилии необходимо придумать), некоторые из них участвуют на нескольких этапах проекта.

При составлении и оформлении отчета следует придерживаться рекомендаций, представленных на странице <http://unesco.kemsu.ru/student/rule/rule.html>.

№ п/п	Тема занятия	Задания	
		В аудитории	Дома
1	Разработка описания и анализ информационной системы	1. Изучить предлагаемый теоретический материал. 2. Составить подробное описание информационной системы.	3. На основании описания системы провести анализ осуществимости. В ходе анализа ответить на вопросы <ul style="list-style-type: none"> • Что произойдет с организацией, если система не будет введена в эксплуатацию?

			<ul style="list-style-type: none"> • <i>Какие текущие проблемы существуют в организации и как новая система поможет их решить?</i> • <i>Каким образом система будет способствовать целям бизнеса?</i> • <i>Требуется ли разработка системы технологии, которая до этого не использовалась в организации?</i> <p>Результатом анализа должно явиться заключение о возможности реализации проекта.</p> <p>4. Распределить роли в группе (руководитель проекта-разработчик, системный аналитик-разработчик, тестер-разработчик).</p> <p>5. Заполнить разделы плана:</p> <ul style="list-style-type: none"> • <i>Введение</i> • <i>Организация выполнения проекта</i> • <i>Анализ рисков</i> <p>Разделы должны содержать рекомендации относительно разработки системы, базовые предложения по объёму требуемого бюджета, числу разработчиков, времени и требуемому программному обеспечению.</p> <p>6. Составить отчет о проделанной работе.</p>
2	Разработка требований к информационной системе	1. Изучить предлагаемый теоретический материал.	4. Составить информационную модель будущей системы, включающую в себя описание основ-

		<p>2. Построить опорные точки зрения на основании метода VORD для формирования и анализа требований. Результатом должны явиться две диаграммы: диаграмма идентификации точек зрения и диаграмма иерархии точек зрения.</p> <p>3. Составить сценарии основных событий.</p>	<p>ных объектов системы и взаимодействия между ними. На основании полученной информационной модели и диаграмм идентификации точек зрения, диаграмма иерархии точек зрения, а также сценария событий сформировать требования пользователя и системные требования.</p> <p>5. На основании описания системы (Лабораторная работа №1), информационной модели, пользовательских и системных требований составить техническое задание на создание программного обеспечения (см. Приложение А). ТЗ должно содержать основные разделы, описанные в ГОСТ 34.602-89.</p> <p>6. Построить отчет, включающий все полученные уровни модели, описание функциональных блоков, потоков данных, хранилищ и внешних объектов.</p>
3	<p>Методология объектно-ориентированного моделирования</p>	<p>1. Изучить предлагаемый теоретический материал.</p> <p>2. Постройте диаграмму вариантов использования для выбранной информационной системы.</p>	<p>3. Выполните реализацию вариантов использования в терминах взаимодействующих объектов и представляющую собой набор диаграмм:</p> <ul style="list-style-type: none"> • диаграмм классов, реализующих вариант использования; • диаграмм взаимодействия (диаграмм последовательности и кооперативных диаграмм), отражающих взаимодействие объектов в процессе реализации варианта использования. <p>4. Разделить классы по пакетам используя один из механизмов разбиения.</p> <p>5. Постройте диаграмму состояний</p>

			для конкретных объектов информационной системы.
4	Методология управление проектами	<ol style="list-style-type: none"> 1. Изучить предлагаемый теоретический материал. 2. Построить временную и сетевую диаграммы для выбранного проекта. 3. Построить диаграмму распределения работников по этапам. 	<ol style="list-style-type: none"> 4. Построить список возможных рисков с указанием названия риска, его описание и типа. 5. Провести анализ рисков. 6. Описать стратегию планирования рисков. 7. Построить отчет, включающий все полученные диаграммы и описание стратегии планирования рисков.

3.3 Семестровая работа, ее характеристика

Семестровая работа направлена на самостоятельное создание проекта по реализации программного продукта на предложенную тему. Семестровые задания даются на группу студентов (при использовании проектного метода) или индивидуально.

Проект должен включать:

- анализ осуществимости внедрения системы;
- Техническое задание (можно оформить в виде приложения к проекту);
- описание информационных объектов;
- функциональную модель системы;
- объектные модели, описывающие состояние системы, варианты использования, объекты, архитектуру системы;
- этапы реализации проекта;
- планирование рисков;
- календарное планирование с учетом распределения работ по членам коллектива;
- планирование контрольных мероприятий;
- оценку стоимости проекта.

Результаты семестровой работы должны быть представлены в виде Отчета по семестровой работе, который должен содержать обязательные элементы и разделы, отражающие предъявленные выше требования.

При составлении и оформлении отчета следует придерживаться рекомендаций, представленных на странице <http://unesco.kemsu.ru/student/rule/rule.html>.

4. Учебно-методические материалы по дисциплине

Презентации в формате MS PowerPoint. Тексты лабораторных заданий в формате MS Word.

Основная литература

1. Соммервиль Иан. Инженерия программного обеспечения, 6-е издание. : Пер. с англ. – М.: Издательский дом “Вильямс”, 2002. – 624 с.

Дополнительная литература

2. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. – С-П.: Издательство «Питер», 2003. – 432 с.
3. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы
4. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению
5. Материалы сайта <http://www.uml.org>
6. Материалы сайта <http://www.omg.org/technology/documents/formal/uml.htm>
7. Материалы сайта <http://www.uml.ru>
8. Материалы сайта <http://www.citforum.ru>

Список основной учебной литературы

<i>Сведения об учебниках</i>			<i>Количество экземпляров в библиотеке на момент утверждения программы</i>
<i>Наименование, гриф</i>	<i>Автор</i>	<i>Год издания</i>	
Инженерия программного обеспечения	Соммервиль Иан	2003	2
Язык UML. Руководство пользователя	Буч Г., Рамбо Дж., Джекобсон А.	2003	5

5. Формы текущего, промежуточного и рубежного контроля

Примерные варианты семестровых заданий

Для полученной информационной системы построить:

1. Сформировать пользовательские и системные требования:

- На основании описания системы провести анализ осуществимости, результатом которого должно явиться заключение о возможности реализации проекта.
 - Построить опорные точки зрения на основании метода VOID для формирования и анализа требований. Результатом должны явиться две диаграммы: диаграмма идентификации точек зрения и диаграмма иерархии точек зрения.
 - Сформировать пользовательские и системные требования.
2. Функциональную модель системы:
- Модель должна отражать весь указанный в описании функционал, а также чётко отражать существующие потоки данных и описывать правила их движения.
 - В модели должно быть не менее трёх уровней:
 - Не менее двух уровней декомпозиции в стандарте IDEF0 (контекстная диаграмма + диаграммы A0).
 - На 3-м и далее уровнях должна быть декомпозиция в стандарте IDEF3.
 - На диаграмме 2-го уровня не менее 4-х функциональных блоков.
 - На диаграммах 3-го уровня и далее не менее 2-х функциональных блоков.
3. Комплекс UML диаграмм:
- Диаграммы вариантов использования.
 - Диаграммы классов, реализующие варианты использования.
 - Диаграммы последовательности, отражающих взаимодействие объектов в процессе реализации варианта использования.
 - Диаграммы состояний для основных объектов информационной системы.
4. Модель управления проектом. Построение модели включает:
- Определение всех этапов проекта.
 - Определение зависимых этапов.
 - Определение длительности этапов.
 - Построение на основе полученных данных сетевой диаграммы
 - Построение на основе полученных данных временной диаграммы.
 - Распределение персонала по этапам.
 - Построение диаграммы распределения работников по этапам.
- Требования:
- При определении этапа указывается его название – отражающее суть этапа (например, определение пользовательских требований, проектирование интерфейса и т.д.).
 - Этапов должно быть не менее 7.
 - Срок реализации проекта – пол года с 1.01.2007 по 1.07.2007.
 - В проекте задействовано 6 человек персонала, некоторые из них уча-

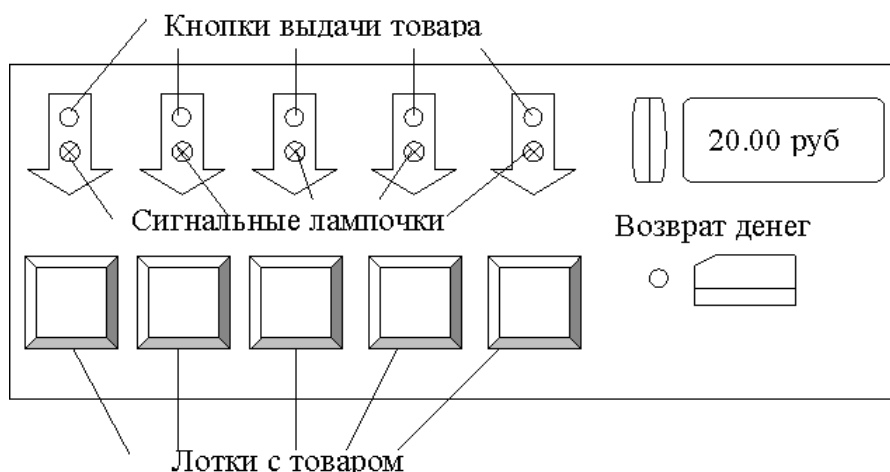
ствуют на нескольких этапах проекта. Персонал:

- Иванов В.П.
- Петров С.Н.
- Сидоров Е.Л.
- Тимофеева О.В.
- Лазарева С.Ю.
- Фралов Е.С.

Варианты

№1 Торговый автомат

Программное обеспечение встроенного процессора универсального торгового автомата.



Лицевая панель торгового автомата

Внешний вид автомата изображен на рисунке. В автомате имеется пять лотков для хранения и выдачи товаров. Загрузка товаров на лотки осуществляется обслуживающим персоналом. Автомат следит за наличием товара. Если какой-либо товар распродан, автомат отправляет сообщение об этом на станцию обслуживания и информирует покупателей (зажигается красная лампочка рядом с лотком данного товара).

Автомат принимает к оплате бумажные купюры и монеты. Специальный индикатор высвечивает текущую сумму денег, принятых автоматом к оплате. После ввода денег клиент нажимает на кнопку выдачи товара. Выдача товара производится только в том случае, если введенная сумма денег соответствует цене товара. Товар выдается поштучно. При нажатии на кнопку «Возврат» клиенту возвращаются все принятые от него к оплате деньги. Возврат денег не производится после выдачи товара. Автомат должен корректно работать при одновременном нажатии на кнопки выдачи товара и возврата денег.

В специальном отделении автомата, закрываемом замком, есть «секретная кнопка», которая используется обслуживающим персоналом для выемки выручки. При нажатии на эту кнопку открывается доступ к ящику с деньгами.

Автомат получает со станции обслуживания данные о товарах и хранит их в своей памяти. Данные включают в себя цену, наименование товара, номер лотка, на котором находится товар и количество товара на лотке.

№ 2 Табло на станции метро

Программное обеспечение табло для информационной службы метрополитена.

Табло расположены на каждой станции метро. Они работают под управлением единого пункта управления (ПУ) информационной службы метро. Табло отображает текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, находящейся на ПУ.

На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется с ПУ.

В «бегущей строке» табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает с ПУ.

Дополнительная функция табло – по запросу с ПУ оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях).

№ 3 Система автоматизации для пункта проката видеокассет

Программное обеспечение системы автоматизации работы пункта проката видеокассет (далее в тексте – системы).

Пункт проката содержит каталог кассет, имеющихся в наличии в данный момент времени. Система поддерживает работу каталога, позволяя служащим проката добавлять новые наименования кассет, удалять старые и редактировать данные о кассетах.

Клиент, обратившийся в пункт, выбирает кассету по каталогу, вносит залог и забирает ее на определенный срок. Срок проката, измеряемый в сутках, оговаривается при выдаче кассеты. Стоимость проката вычисляется системой исходя из тарифа за сутки и срока проката. Клиент возвращает кассету и оплачивает прокат. Если кассета не повреждена, клиенту возвращается залог. Служащий пункта проката регистрирует сдачу кассеты клиенту и ее возврат в системе. Если клиент повредил кассету, то кассета удаляется из каталога, а залог остается в кассе проката.

При необходимости служащий может запросить у системы следующие данные:

- имеется ли в наличии кассета с данным названием;
- когда будет возвращена какая-либо кассета из тех, что сданы в прокат;
- является ли данный клиент постоянным клиентом пункта проката (пользовался ли прокатом 5 или более раз).

Постоянным клиентам предоставляются скидки, а также от них принимаются заявки на пополнение ассортимента кассет. Заявки регистрируются в системе. По ним готовится итоговый отчет, руководствуясь которым, служащие пункта проката обновляют ассортимент кассет.

№ 4 Мини-АТС

Программное обеспечение встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции)

Мини-АТС осуществляет связь между служащими учреждения. Каждый абонент подключен к ней линией связи. Мини-АТС соединяет линии абонентов (осуществляет коммутацию линий). Абоненты имеют номера, состоящие из трех цифр. Специальный номер «9» зарезервирован для внешней связи.

Телефонное соединение абонентов производится следующим образом. Абонент поднимает трубку телефона, и мини-АТС получает сигнал «**Трубка**». В ответ мини-АТС посылает сигнал «**Тон**». Приняв этот сигнал, абонент набирает телефонный номер (посылает три сигнала «**Цифра**»). Мини-АТС проверяет готовность вызываемого абонента. Если абонент не готов (его линия занята), мини-АТС посылает вызывающему абоненту сигнал «**Занято**». Если абонент готов, мини-АТС посылает обоим абонентам сигнал «**Вызов**». При этом телефон вызываемого абонента начинает звонить, а вызывающий абонент слышит в трубке длинные гудки. Вызываемый абонент снимает трубку, и мини-АТС получает от него сигнал «**Трубка**», после чего осуществляет коммутацию линии. Абоненты обмениваются сигналами «**Данные**», которые мини-АТС должна передавать от одного абонента к другому. Когда один из абонентов опускает трубку, мини-АТС получает сигнал «**Конец**» и посылает другому абоненту сигнал «**Тон**».

В любой момент абонент может положить трубку, при этом мини-АТС получает сигнал «**Конец**». После получения этого сигнала сеанс обслуживания абонента завершается.

Если абонент желает соединиться с абонентом за пределами учреждения, то он набирает номер «9». Мини-АТС посылает по линии, соединяющей с внешней (городской) АТС, сигнал «**Трубка**» и в дальнейшем служит посредником между телефоном абонента и внешней АТС. Она принимает и передает сигналы и данные между ними, не внося никаких изменений. Единственное исключение касается завершения сеанса. Получив от городской АТС сигнал «**Ко-**

нец», мини-АТС посылает абоненту сигнал **«Тон»** и ждет сигнала **«Конец»** для завершения обслуживания абонента. Если вызывавший абонент первым вешает трубку, то мини-АТС получает сигнал **«Конец»** и передает его городской АТС и завершает сеанс.

Мини-АТС может получить сигнал **«Вызов»** от городской АТС. Это происходит, когда нет соединений с внешними абонентами. Сигнал **«Вызов»** от городской АТС передается абоненту с кодом **«000»**. Только этот абонент может отвечать на внешние звонки.

№ 5 Телефон

Программное обеспечение встроенного микропроцессора для аппарата учрежденческой телефонной сети.

Аппарат подключен к линии связи, ведущей к мини-АТС. В его задачу входит прием и передача сигналов (в том числе и голосовых данных) мини-АТС. Аппарат имеет кнопочную панель управления, экран для отображения набираемых номеров, звонок и трубку, в которую встроены микрофон и громкоговоритель.

В начальном состоянии трубка телефона повешена, телефон не реагирует на нажатия кнопок. Телефон реагирует только на сигнал **«Вызов»** от мини-АТС, при этом включается звонок.

При снятии трубки на АТС подается сигнал **«Трубка»**. При получении ответного сигнала **«Тон»** от АТС телефон воспроизводит звуковой тон **«Готов»** (длинный непрекращающийся гудок) в трубку. При получении сигнала **«Занято»**, в трубке воспроизводится тон **«Занято»** (частые короткие гудки).

Пользователь, слыша в трубке тон **«Готов»**, набирает трехзначный номер. Номер может быть набран при помощи кнопок с цифрами или нажатием на специальную кнопку **« # »**. При нажатии на кнопку с цифрой соответствующий ей сигнал **«Цифра»** передается АТС. Нажатия на кнопки с цифрами после третьего игнорируются. Во время набора номера введенные цифры отображаются на экране. Последний полностью набранный номер запоминается в памяти аппарата для того, чтобы можно было его воспроизвести при нажатии на кнопку **« # »**. При нажатии на эту кнопку номер из памяти аппарата высвечивается на экране, и АТС передается последовательность из трех сигналов **«Цифра»**. В ответ на набранный номер от АТС приходит либо сигнал **«Занято»**, либо сигнал **«Вызов»**. При получении сигнала **«Вызов»** телефон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал **«Данные»**.

Телефон воспроизводит данные, передаваемые с сигналом, в трубку. Ответ пользователя воспринимается микрофоном трубки, преобразуется в сигнал **«Данные»** и передается АТС. Обмен данными прерывается, если повешена трубка одного из телефонов, участвующих в обмене. О том, что трубку повесил вызываемый абонент, сообщает сигнал **«Занято»**, посылаемый АТС. После того, как трубка аппарата была повешена, телефон посылает АТС сигнал **«Конец»**, и телефон переходит в начальное состояние.

№ 6 Стиральная машина

Программное обеспечение встроенного микропроцессора стиральной машины.

Машина предназначена для автоматической стирки белья. Машина включает в себя следующие устройства: бак для белья, клапаны для забора и слива воды, мотор, устройство подогрева воды, термометр, таймер, дверца для доступа в бак, несколько емкостей для различных моющих средств, панель управления с кнопками и индикатором. В памяти машины хранятся 5 программ стирки, заданные изготовителем. Пользователи не могут вносить в них изменения. Каждая программа определяет температуру воды, длительность стирки, используемые моющие средства (номер емкости и время подачи), скорость вращения бака во время стирки и отжима. Вариант задания предусматривает разработку схемы базы данных для хранения программ стирки в памяти машины.

Для использования машины необходимо открыть дверцу, поместить белье в бак, поместить моющие средства в емкости, закрыть дверцу, выбрать программу стирки и нажать на кнопку «Пуск». Перед тем как приступить к стирке машина открывает клапан для забора воды, набирает необходимое количество воды, после чего закрывает клапан. Далее, машина действует по выбранной пользователем программе:

- Подогревает, если необходимо воду до нужной температуры.
- Включает таймер и запускает вращение бака для стирки.
- По таймеру подает в бак моющие средства, предусмотренные программой.
- По окончании стирки сливает воду и запускает отжим.

Во время работы на индикаторе высвечивается время, прошедшее от начала стирки (минуты и секунды), текущий режим работы (стирка или отжим), номер текущей программы стирки. В целях безопасности дверца бака блокируется до окончания стирки. Машина не воспринимает нажатий на кнопки, за исключением одной – пользователь имеет возможность в любой момент нажать на кнопку «Останов», чтобы принудительно остановить стирку и слить воду.

№ 7 Таксофон

Программное обеспечение встроенной системы управления работой таксофона городской телефонной сети.

Таксофон предназначен для оказания платных услуг телефонной связи. Он подключен к линии связи. В нем имеется кнопочная панель, дисплей, трубка со встроенным микрофоном и громкоговорителем, приемник карт – устройство для считывания телефонных карт, используемых для оплаты разговора.

В начальном состоянии трубка таксофона повешена, дисплей потушен, таксофон не реагирует на нажатия кнопок и какие-либо сигналы из линии. При

снятии трубки таксофон выдает на дисплей сообщение «Вставьте карту» и ожидает, когда пользователь вставит карту в приемник. Дальнейшее функционирование таксофона осуществляется только при вставленной карте. Если карту вынимают, таксофон возвращается к началу и выдает сообщение о необходимости вставить карту. При попадании карты в приемник производится считывание информации с карты. Если кредит исчерпан или карта не пригодна (не удастся узнать кредит), то таксофон выдает соответствующее сообщение на дисплей таксофона. Если карта может быть использована для оплаты, то на дисплей выдается количество «единиц» на карте, и на телефонную станцию (АТС) подается сигнал «Трубка». При получении ответного сигнала «Тон» из линии таксофон воспроизводит звуковой тон «Готов» (длинный непрекращающийся гудок) в трубку. При получении сигнала «Занято», в трубке воспроизводится тон «Занято» (короткие гудки).

После получения от АТС сигнала «Тон» от пользователя принимаются семизначный номер вызываемого абонента, остальные нажатия на кнопки игнорируются. Когда пользователь нажимает на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Во время набора номера введенные цифры отображаются на дисплее. В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» таксофон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Таксофон воспроизводит данные, передаваемые с сигналом, в трубку. При получении данных из трубки, аппарат преобразует их в сигнал «Данные» и передает их АТС. Во время разговора на дисплее ведется отсчет времени и уменьшается кредит на телефонной карте - каждые 15 секунд вычитается четверть «единицы». Обмен данными прерывается, в следующих случаях:

- исчерпан кредит;
- карта вынута из приемника;
- от АТС пришел сигнал «Занято»;
- повешена трубка таксофона.

Если трубка была повешена, аппарат посылает в линию сигнал «Конец» и выдает на дисплей сообщение «Выньте карту». После извлечения карты из приемника таксофон переходит в начальное состояние.

№ 8 Банкомат

Программное обеспечение банкомата.

Банкомат – это автомат для выдачи наличных денег по кредитным пластиковым карточкам. В его состав входят следующие устройства: дисплей, панель управления с кнопками, приемник кредитных карт, хранилище денег и лоток для их выдачи, хранилище конфискованных кредитных карт, принтер для печати справок.

Банкомат подключен к линии связи для обмена данными с банковским компьютером, хранящим сведения о счетах клиентов.

Обслуживание клиента начинается с момента помещения пластиковой карточки в банкомат. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Персональный код представляет собой четырехзначное число. Затем банкомат проверяет правильность введенного кода. Если код указан неверно, пользователю предоставляются еще две попытки для ввода правильного кода. В случае повторных неудач карта перемещается в хранилище карт, и сеанс обслуживания заканчивается. После ввода правильного кода банкомат предлагает пользователю выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету.

При снятии наличных со счета банкомат предлагает указать сумму (10, 50, 100, 200, 500, 1000 рублей). После выбора клиентом суммы банкомат запрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы центральному компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег. Если он может выдать деньги, то на дисплей выводится сообщение «Выньте карту». После удаления карточки из приемника, банкомат выдает указанную сумму в лоток выдачи. Банкомат печатает справку по произведенной операции, если она была затребована клиентом.

Если клиент хочет узнать остаток на счету, то банкомат посылает запрос центральному компьютеру банка и выводит сумму на дисплей. По требованию клиента печатается и выдается соответствующая справка.

В специальном отделении банкомата, закрываемом замком, есть "секретная кнопка", которая используется обслуживающим персоналом для загрузки денег. При нажатии на эту кнопку открывается доступ к хранилищу денег и конфискованным кредитным картам.

№ 9 Холодильник

Программное обеспечение встроенного процессора холодильника.

Холодильник состоит из нескольких холодильных камер для хранения продуктов. В каждой холодильной камере имеется регулятор температуры, мотор, термометр, индикатор, таймер, датчик открытия двери камеры и устройство для подачи звуковых сигналов.

При помощи терморегулятора устанавливается максимально допустимая температура в данной камере. Мотор предназначен для поддержания низкой температуры. Термометр постоянно измеряет температуру внутри камеры, а индикатор температуры, расположенный на дверце, постоянно высвечивает ее значение. При повышении температуры выше предела, определяемого текущим положением регулятора, включается мотор. При снижении температуры ниже некоторого другого значения, связанного с первым, мотор отключается.

Доступ в камеру осуществляется через дверцу. Если дверь холодильной камеры открыта в течение слишком долгого времени, подается звуковой сиг-

нал. Звуковой сигнал также подается в любых нештатных ситуациях (например, при поломке мотора). Холодильник ведет электронный журнал, в котором отмечаются все происходящие события:

- изменение положения терморегулятора камеры;
- включение и отключение мотора;
- доступ в камеру;
- внештатные ситуации.

№ 10 Замок

Программное обеспечение встроенного процессора кодового замка, регулирующего доступ в помещение.

Кодовый замок состоит из панели с кнопками (цифры «0» ... «9», кнопка «Вызов», кнопка «Контроль»), цифрового дисплея, электромеханического замка, звонка. Панель с кнопками устанавливается с наружной стороны двери, замок устанавливается с внутренней стороны двери, звонок устанавливается внутри охраняемого помещения.

В обычном состоянии замок закрыт. Доступ в помещение осуществляется после набора кода доступа, состоящего из четырех цифр. Во время набора кода введенные цифры отображаются на экране дисплея. Если код набран правильно, то замок открывается на некоторое время, после чего дверь снова закрывается. Содержимое дисплея очищается.

Кнопка «Вызов» используется для подачи звукового сигнала внутри помещения. Кнопка «Контроль» используется для смены кодов. Смена кода доступа осуществляется следующим образом. При открытой двери нужно набрать код контроля, состоящий из четырех цифр, и новый код доступа. Для смены кода контроля нужно при открытой двери и нажатой кнопке «Вызов» набрать код контроля, после чего - новый код контроля.

№ 11 Турникет метро

Программное обеспечение встроенного процессора турникета для метрополитена.

При помощи турникета контролируется проход пассажиров в метро и взимается входная плата. Турникет имеет приемник карт, устройство для перекрытия доступа, таймер, три оптических датчика для определения прохода пассажира, устройство подачи звуковых сигналов, индикаторы «Проход» и «Стоп».

В начальном состоянии турникета зажжен индикатор «Стоп», индикатор «Проход» потушен. Если один из датчиков посылает сигнал, то проход через турникет сразу же перекрывается, и подается предупредительный звуковой сигнал. Для прохода пассажир должен поместить карту в приемник карт. Турникет считывает с нее данные: срок годности карты и количество «единиц» на

ней. Если данные не удается считать, или карта просрочена, или заблокирована, то карта возвращается пассажиру, и турникет остается в исходном состоянии. В другом случае с карты списывается одна «единица», карта возвращается из приемника, индикатор «Стоп» гаснет, зажигается индикатор «Проход», и пассажир может пройти через турникет. Получив от одного из датчиков сигнал, турникет ожидает время, отведенное на проход пассажира (5 секунд), после чего он возвращается в начальное состояние.

Наличие трех датчиков в турникете гарантирует, что при проходе пассажира хотя бы один из них подаст сигнал (датчики невозможно перешагнуть, перепрыгнуть и т. д.). Во время прохода пассажира возможна ситуация, когда все три датчика посылают сигналы. В этом случае принимается только первый сигнал и от момента его приема отсчитывается положенное время. Остальные сигналы игнорируются. Турникет заносит в свою память время всех оплаченных проходов. В конце рабочего дня он передает всю информацию, накопленную за день, в АСУ метрополитена.

№ 12 Система учета товаров

Программное обеспечение системы поддержки заказа и учета товаров в бакалейной лавке.

В бакалейной лавке для каждого товара фиксируется место хранения (определенная полка), количество товара и его поставщик. Система поддержки заказа и учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках, включающей в себя название фирмы, ее адрес и телефон. При помощи системы составляются заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой позиции указываются наименование товара и его количество в заказе. Система учета по требованию пользователя формирует и выдает на печать следующую справочную информацию:

- список всех товаров;
- список товаров, имеющихся в наличии;
- список товаров, количество которых необходимо пополнить;
- список товаров, поставляемых данным поставщиком.

№ 13 Интернет-магазин

Программное обеспечение Интернет-магазина.

Интернет-магазин позволяет делать покупки с доставкой на дом. Клиенты магазина при помощи программы-браузера имеют доступ к каталогу продаваемых товаров, поддержку которого осуществляет Интернет-магазин. В каталоге товары распределены по разделам. О каждом товаре доступна полная информация (название, вес, цена, изображение, дата изготовления и срок годности) Для удобства клиентов предусмотрена система поиска товаров в каталоге. Заполне-

ние каталога информацией происходит автоматически в начале рабочего дня, информация берется из системы автоматизации торговли.

При отборе клиентами товаров поддерживается виртуальная «торговая корзина». Любое наименование товара может быть добавлено в «корзину» или изъято в любой момент по желанию покупателя с последующим пересчетом общей стоимости покупки. Текущее содержимое «корзины» постоянно показывается клиенту.

По окончании выбора товаров производится оформление заказа и регистрация покупателя. Клиент указывает в регистрационной форме свою фамилию, имя и отчество, адрес доставки заказа и телефон, по которому с ним можно связаться для подтверждения сделанного заказа. Заказы передаются для обработки в систему автоматизации торговли. Проверка наличия товаров на складе и их резервирование Интернет-магазином не производятся.

№ 14 WWW-конференция

Программное обеспечение WWW-конференции.

WWW-конференция представляет собой хранилище сообщений в сети Интернет, доступ к которому осуществляется при помощи браузера. Для каждого сообщения конференции хранятся значения следующих полей: номер сообщения, автор, тема, текст сообщения, дата добавления сообщения, ссылка на родительское сообщение. Начальной страницей конференции является иерархический список сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на сообщения верхнего уровня. Сообщение-ответ всегда имеет ссылку на исходное сообщение. В списке отображаются только темы сообщений, их авторы и даты добавления. Просматривая список, пользователь выбирает сообщение и по гиперссылке открывает страницу с текстом сообщения. Помимо текста на этой странице отображается список (иерархический) сообщений являющихся ответами, ответами на ответы и т.д. Для удобства пользователей необходимо предусмотреть поиск сообщений по автору или по ключевым словам в теме или тексте сообщения.

Сообщения добавляются в конференцию зарегистрированными пользователями, которые при отправке сообщения должны указать своё имя и пароль. Регистрирует новых пользователей модератор конференции - её ведущий. При регистрации пользователь заполняет специальную форму, содержимое которой затем пересылается модератору и запоминается в базе пользователей. Модератор решает, регистрировать пользователя или нет, и отправляет свой ответ.

При добавлении сообщений пользователь имеет возможность начать новую тему или ответить на ранее добавленные сообщения. После добавления сообщения оно доступно для чтения всем пользователям (даже незарегистрированным), и список сообщений обновляется. Модератор имеет право по тем или иным причинам удалять сообщения любых авторов. Он также может наказывать пользователей, нарушающих правила поведения в конференции, лишая на

некоторое время пользователя возможности добавлять и редактировать сообщения.

№ 15 Каталог ресурсов Интернет

Программное обеспечение каталога ресурсов сети Интернет.

В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация.

Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен пользователям. Пользователям каталога должны быть предоставлены возможности по поиску ресурсов. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать список ресурсов по релевантности (соответствию ключевым словам из запроса) или по дате обновления.

Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая Web-сайты, URL которых хранятся в каталоге. Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

№ 16 Будильник

Программное обеспечение встроенного микропроцессора для будильника.

Будильник постоянно отображает текущее время (часы, минуты, например: 12 : 00). Управление будильником осуществляется следующими кнопками:

- кнопкой режима установки времени,
- кнопкой режима установки времени срабатывания,
- двумя отдельными кнопками для установки часов и минут,
- кнопкой сброса сигнала «СБРОС».

На будильнике имеется переключатель режима работы со следующими положениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР».

Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс. Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени сраба-

тивания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается.

Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению «ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

№17 Генеалогическое дерево

Программное обеспечение системы для поддержки генеалогических деревьев.

Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-жены», «дети-родители» и «братья-сестры». Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. Недопустимо, чтобы человек был собственным предком или потомком.

Пользователи системы могут осуществлять поиск полезной информации по дереву:

- находить для указанного члена семьи его детей;
- находить для указанного члена семьи его родителей;
- находить для указанной персоны братьев и сестер, если таковые есть;
- получать список всех предков персоны;
- получать список всех потомков персоны;
- получать список всех родственников персоны;
- проследивать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр – брат Ольги, Ольга – жена Ивана»).

№ 18 Телевизор

Программное обеспечение встроенной системы управления работой телевизора.

В телевизоре имеются следующие устройства: приемник телевизионного сигнала, устройство отображения картинки, память каналов, память настроек,

управляющие кнопки, пульт дистанционного управления (ДУ). Управление телевизором осуществляется при помощи кнопок на корпусе (их четыре: «ВКЛ / ВЫКЛ», « - », « + », кнопка начальной установки) и пульта ДУ. Кнопка «ВКЛ / ВЫКЛ» позволяет включать и выключать телевизор. После включения телевизора на экран отображается передача, идущая по каналу №1, при этом используются параметры изображения и значение громкости, сохраненные в памяти настроек.

Память каналов телевизора хранит до 60 каналов. Каналы нумеруются, начиная с нуля. Последовательное переключение каналов осуществляется при помощи кнопок « - » и « + ». Нажатие на « + » переключает телевизор на канал с номером, на единицу большим (с 59-го канала телевизор переключается на 0-ой). Нажатие на « - » переключает телевизор на канал с номером, на единицу меньшим (с 0-го канала телевизор переключается на 59-ый).

При нажатии на кнопку начальной установки очищается память каналов телевизора, после чего осуществляется поиск передач и сохранение их частот в памяти каналов. Поиск начинается с нижней границы рабочего диапазона телевизора. На экран телевизора выводится «синий экран». Рабочая частота постепенно увеличивается до тех пор, пока приемник не обнаружит телевизионный сигнал. Найденная передача выводится на экран. Также отображается номер, под которым найденный канал будет сохранен в памяти (начиная с 1). Затем поиск продолжается до тех пор, пока не будет достигнута верхняя граница диапазона, или пока не заполнена вся память каналов. Телевизор принимает управляющие сигналы с пульта ДУ. На пульте ДУ расположены следующие кнопки:

- кнопки с цифрами «0» ... «9» для прямого переключения канала (по номеру);
- кнопки «П-» и «П+» для последовательного переключения каналов;
- кнопки «Г-» и «Г+» для изменения громкости;
- кнопки «МЕНЮ», « < » и « > » для доступа к экранному меню.

Для прямого переключения на нужный канал его номер набирается с помощью кнопок с цифрами. После нажатия первой цифры в течение 5 секунд ожидается нажатие второй. Если вторая цифра не была нажата, то номер канала считается состоящим из одной цифры и осуществляется переключение на него.

Кнопки «П-» и «П+» на пульте имеют те же функции, что и кнопки « - » и « + » на корпусе телевизора. Кнопки «Г-» и «Г+» позволяют увеличивать или уменьшать громкость. Каждое нажатие на «Г-» уменьшает громкость на одну единицу, пока она больше нуля, «Г+» увеличивает громкость на единицу, пока не достигнуто максимальное значение. Текущее значение громкости сохраняется в памяти настроек. Кнопки «МЕНЮ», « < » и « > » позволяют устанавливать значения настроек, хранящихся в памяти телевизора. При нажатии на кнопку «МЕНЮ» внизу экрана возникает надпись «ЯРКОСТЬ» и полоса, отображающая текущее значение яркости. Кнопками « < » и « > » яркость можно уменьшить или увеличить. При работе с меню нажатия на все остальные кнопки иг-

норируются. После повторного нажатия на кнопку «МЕНЮ» значение яркости запоминается в памяти настроек, и осуществляется переход к настройке контрастности. Настройка контрастности и остальных параметров (четкости, цветовой гаммы) происходит аналогично. Нажатие на кнопку «МЕНЮ» по окончании настройки цветовой гаммы (последнего пункта меню) приводит к окончанию работы с меню.

№ 19 Система поддержки составления расписания занятий

Программное обеспечение системы поддержки составления расписания занятий.

Система обеспечивает составление расписания некоторого учебного заведения, внесение в расписание изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы на заданный день или неделю).

В расписании фиксируются время и место проведения занятия, предмет и преподаватель, проводящий занятие, а также номер группы, для которой это занятие проводится. Расписание не должно содержать коллизий (например, разные занятия не должны пересекаться друг с другом по месту и времени их проведения, один преподаватель не может вести одновременно два разных занятия, в одно и то же время у одной и той же группы не может быть два различных занятия и т. д.).

№ 20 Домофон

Программное обеспечение встроенного микропроцессора домофона.

Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена панель с кнопками на каждую квартиру, микрофон и динамик. В каждой квартире имеется кнопка «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, в квартире имеется микрофон и динамик.

Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры на внешней панели. При этом в квартире раздается звонок (если подача звонка в квартиру не заблокирована). Услышав звонок, жилец квартиры нажимает на кнопку «СВЯЗЬ» внутренней панели домофона, после чего домофон устанавливает звуковое сообщение между жильцом и посетителем. Звуки, произносимые посетителем в микрофон, установленный на внешней панели, воспроизводятся в динамике, установленном в квартире. Звуки из микрофона в квартире, передаются в динамик на внешней панели. После сеанса связи жилец может нажать на кнопку «ОТКРЫТЬ», чтобы замок на двери в подъезд открылся, и посетитель смог войти. По истечении минуты замок должен снова заблокировать вход в подъезд. Жилец, который желает, чтобы его не беспокоили, может отключить подачу звонка в свою квартиру, нажав на кнопку «БЛОКИРОВКА». Повторное нажатие на эту кнопку вновь включает подачу звонка.

Вопросы к экзамену

1. Общие понятия

- 1.1. Почему затраты на формирование требований и определении структуры системы могут превосходить стоимость продаваемых программ?
- 1.2. Какими четырьмя характеристиками должен обладать любой программный продукт?
- 1.3. Каково различие между моделью процесса создания ПО и самим процессом?

2. Создание ПО

- 2.1. Почему в процессе определения требований необходимо различать разработку пользовательских требований и разработку системных требований?
- 2.2. Каковы пять основных компонентов любых методов проектирования?
- 2.3. Разработайте модель процесса тестирования исполняемой программы.

3. Управление проектами

- 3.1. Объясните, почему нематериальность программных систем порождает особые проблемы в процессе управления программными проектами.
- 3.2. Объясните, почему хорошие программисты не всегда могут быть хорошими менеджера проектов.
- 3.3. Объясните, почему процесс планирования проекта является итерационным и почему план должен постоянно пересматриваться в течение всего срока выполнения проекта.
- 3.4. Опишите кратко каждый раздел плана выполнения программного проекта.
- 3.5. В табл. 4.9 показаны этапы некоего проекта, их длительность и взаимозависимость между ними. Нарисуйте сетевую и временную диаграммы этапов работ для данного проекта.

Таблица 4.9. Этапы проекта

Этап	Длительность (дни)	Зависимость
T1	10	
T2	15	T1
T3	10	T1, T2
T4	20	
T5	10	
T6	15	T3, T4
T7	20	T3
T8	35	T7
T9	15	T6
T10	5	T5, T9
T11	10	T9

T12	20	T10
T13	35	T3, T4
T14	10	T8, T9
T15	20	T12, T14
T16	10	T15

- 3.6. Менеджер проекта предупреждает о возможной задержке выполнения работ, которой можно избежать только за счет бесплатных сверхурочных работ команды разработчиков. Все члены команды имеют семьи, требующие определенной доли внимания. Обсудите возможность отклонения предложения менеджера о бесплатных сверхурочных работах либо согласия предпочесть интересы организации семейным интересам. Какие аргументы наиболее весомы в этой дискуссии?
- 3.7. Как опытному программисту, вам предложили возглавить управление проектом, но вы чувствуете, что больше пользы можете принести в качестве технического специалиста, а не менеджера проекта. Обсудите возможности принятия или отклонения предложения возглавить программный проект.

4. Формальные спецификации ПО

- 4.1. Объясните, почему архитектурное проектирование системы должно предшествовать разработке формальной спецификации.
- 4.2. Банковские автоматы используют информацию с карточки клиента, предоставляющую идентификатор банка, номер счета и персональный идентификатор клиента. Они также получают информацию из центральной базы данных и вносят в нее изменения по завершении транзакции. Используя ваши знания работы банкоматов, напишите Z-схему, определяющую состояния системы, проверку карточки клиента и снятие денежных средств.
- 4.3. Вы системный инженер и вас просят назвать наилучший способ разработки программного обеспечения для сердечного стимулятора, критического по обеспечению безопасности. Вы предлагаете разработать формальную спецификацию системы, но ваше предложение отвергнуто менеджером. Вы считаете, что его доводы не обоснованы и базируются на предубеждениях. Будет ли этичной разработка системы с использованием методов, которые вы считаете неподходящими?

5. Архитектурное проектирование

- 5.1. Объясните, почему архитектуру системы необходимо разработать до окончания создания спецификации.
- 5.2. Обсудите преимущества и недостатки модели потоков данных и объектной модели в предположении, что необходимо разработать как локальную, так и распределенную версии программного приложения.
- 5.3. Предположим, существует конкретная должность "архитектор про-

граммного обеспечения"; его роль состоит в проектировании системной архитектуры независимо от того, для какого заказчика выполняется данный проект. Такая должность может быть, например, в компании, занимающейся разработкой ПО. Какие трудности могут возникнуть при введении данной должности?

6. Объектно-ориентированное проектирование

- 6.1. Объясните, почему в проектировании систем применение подхода, который полагается на слабо связанные объекты, скрывающие информацию о своем представлении, приводит к созданию системной архитектуры, которую затем можно легко модифицировать.
- 6.2. Покажите на примерах разницу между объектом и классом объектов.
- 6.3. При каких условиях можно разрабатывать систему, в которой объекты выполняются параллельно?
- 6.4. Разработайте проект метеостанции, показывающий взаимодействие между подсистемой сбора данных и приборами, собирающими данные. Воспользуйтесь диаграммой последовательностей.
- 6.5. Определите возможные объекты в следующей системе, применяя при этом объектно-ориентированный подход:
 - Система "Дневник группы" поддерживает расписание собраний и встреч в группе сотрудников. Для организации встречи, в которой участвует группа людей, система находит общие для всех личных дневников свободные "окна" и назначает эту встречу на определенное время. Если система не находит общих "окон", то начинает взаимодействовать с пользователями, чтобы реорганизовать личные дневники и тем самым создать "окно" для встречи.
- 6.6. Определите возможные объекты в следующей системе, применяя при этом объектно-ориентированный подход:
 - Установлена полностью автоматизированная бензоколонка. Водитель вставляет кредитную карточку в считывающее устройство, связанное с насосом; карточка по линиям коммуникаций проверяется кредитной компанией, устанавливается требуемое количество бензина. Затем автомобиль заправляется горючим. Когда подача прекращается, с кредитной карточки водителя снимается стоимость полученного бензина. Кредитная карточка возвращается после вычета водителю. Если карточка неверна, она возвращается водителю перед подачей топлива.

7. Модели систем

- 7.1. Разработайте модель рабочего окружения для информационной системы больницы. Модель должна предусматривать ввод данных о новых пациентах и систему хранения рентгеновских снимков.
- 7.2. Создайте модель обработки данных в системе электронной почты. Необходимо отдельно смоделировать отправку почты и ее получение.
- 7.3. Нарисуйте модель конечного автомата управляющей системы для программного обеспечения проигрывателя компакт-дисков;

- 7.4. Используя подход "сущность-связь", опишите возможную модель данных для системы библиотечного каталога.
- 7.5. Разработайте объектную модель, включающую диаграммы иерархии классов и агрегирования, и показывающую основные элементы системы персонального компьютера и его программного обеспечения.
- 7.6. Разработайте диаграмму последовательностей, которая показывает действия студента, регистрирующегося на определенный курс в университете. Курс может иметь ограниченное число мест, поэтому процесс регистрации должен проверять количество доступных мест. Предположите, что студент обращается к электронному каталогу курсов, чтобы выяснить количество доступных мест.

8. Верификация и аттестация ПО

- 8.1. Обсудите различия между верификацией и аттестацией и объясните, почему аттестация является более сложным процессом.
- 8.2. Объясните, почему не нужно устранять все дефекты в программе перед ее поставкой заказчику. До каких пор следует тестировать программу, чтобы удостовериться, что она соответствует своему назначению?
- 8.3. Объясните, почему инспектирование программы является эффективным методом обнаружения в ней ошибок. Какие типы ошибок нельзя обнаружить методом инспектирования?
- 8.4. Составьте отчет, в котором бы приводились преимущества метода "чистая комната", а также связанные с ним расходы и риски.
- 8.5. Менеджер решил для оценки специалистов в качестве исходных данных воспользоваться отчетами о результатах инспектирования программ. В отчетах содержится информация о том, кто совершил и кто обнаружил ошибки в программе. Этичны ли действия менеджера? Этично ли заранее проинформировать персонал об этом? Как это решение может повлиять на процесс инспектирования?
- 8.6. Один из подходов, широко используемых при тестировании ПО, состоит в тестировании системы до тех пор, пока не будут израсходованы все средства, выделенные на тестирование. Затем система передается заказчику. Обсудите этичность такого подхода.

9. Тестирование программного обеспечения

- 9.1. Обсудите различия между тестированием методом черного ящика и структурным тестированием. Подумайте, каким образом можно совместно использовать эти методы в процессе тестирования дефектов.
- 9.2. Какие проблемы тестирования могут возникнуть в программах, которые обрабатывают как очень большие, так и очень малые числа?
- 9.3. На примере небольшой программы покажите, почему практически невозможно полностью протестировать программу.
- 9.4. Объясните, почему методы нисходящего и восходящего тестирования не подходят для объектно-ориентированных систем.

10. Оценка стоимости программного продукта

- 10.1. Опишите два подхода к определению производительности программиста. Отметьте преимущества и недостатки каждого подхода.
- 10.2. Приведите пять факторов, которые оказывают существенное влияние на производительность команды программистов по разработке больших встроенных систем реального времени.
- 10.3. Любой оценке стоимости присущ определенный риск, независимо от метода оценки. Предложите четыре способа снижения возможного риска при оценке стоимости.
- 10.4. Назовите три причины, по которым алгоритмические оценки стоимости, проведенные различными компаниями, не будут сопоставимыми.
- 10.5. Объясните, каким образом менеджеры проектов могут использовать алгоритмический подход к оценке стоимости для анализа проектных характеристик. Опишите ситуацию, когда менеджеры выбирают подход, не основанный на принципе наименьшей стоимости проекта.
- 10.6. Некоторые большие программные проекты требуют написания миллионов строк кода. Объясните, насколько полезными могут быть модели определения стоимости для таких систем. В каких случаях они могут быть неприменимы к большим системам?
- 10.7. Насколько этично назначить компанией-разработчиком относительно низкую цену для контракта, зная, что при столь неопределенных требованиях можно со временем повысить цену за дополнительные изменения в них, которые со временем обязательно будут сделаны заказчиком?
- 10.8. Следует ли менеджерам применять критерий производительности для определения деловой характеристики специалиста? Какие меры предосторожности необходимы, чтобы этот процесс определения профессиональных возможностей специалиста не влиял на качество его работы?

11. Проектирование интерфейса пользователя

- 11.1. Отмечалось, что объекты, которыми манипулирует пользователь, должны отображать его понятия предметной области приложения ПО (а не компьютерной предметной области). Предложите подходящие объекты манипулирования для следующих типов пользователей и систем.
- 11.2. Какие факторы следует учитывать при проектировании интерфейсов, использующих меню, для таких систем, как банкоматы? Опишите основные черты интерфейса банкомата, которым вы пользуетесь.
- 11.3. Предложите способы адаптации пользовательского интерфейса в системах электронной коммерции (например, виртуального книжного магазина или магазина музыкальных дисков) для пользователей, имеющих физические недостатки, например плохое зрение или проблемы опорно-двигательной системы.

- 11.4. Какими основными принципами следует руководствоваться при использовании цветов в интерфейсах пользователя? Предложите более эффективный способ использования цветов в интерфейсе любого известного вам приложения.
- 11.5. Составьте анкету по сбору данных об интерфейсе какой-либо известной вам программы (например, текстового редактора). Если есть возможность, распространите эту анкету среди других пользователей и попытайтесь оценить результаты анкетирования. Что вы узнали об интерфейсе программы из анкет?
- 11.6. С какими этическими проблемами сталкиваются разработчики интерфейсов, когда пытаются согласовать запросы конечных пользователей системы с требованиями организации, которая оплачивает разработку данной системы?

12. Проектирование систем реального времени

- 12.1. Почему системы реального времени обычно реализованы как множество параллельных процессов? Проиллюстрируйте свой ответ примерами.
- 12.2. Объясните, почему объектно-ориентированные методы разработки ПО не всегда подходят к системам реального времени.
- 12.3. Обсудите сильные и слабые стороны Java как языка программирования для реализации систем реального времени.

13. Разработка требований

- 13.1. Предложите, кто бы мог участвовать в формировании требований для университетской системы регистрации студентов. Объясните, почему почти неизбежно, что требования, сформулированные разными лицами, будут противоречивы.
- 13.2. Разрабатывается система ПО для автоматизации библиотечного каталога. Эта система будет содержать информацию относительно всех книг в библиотеке и будет полезна библиотечному персоналу, абонентам и читателям. Система должна иметь средства просмотра каталога, средства создания запросов и средства, позволяющие пользователям резервировать книги, находящиеся в данный момент на руках. Определите основные опорные точки зрения, которые необходимо учесть в спецификации системы, и покажите их взаимоотношения, используя диаграмму иерархии точек зрения.
- 13.3. Приведите пример типа системы, где социальные и политические факторы могут иметь влияние на системные требования.
- 13.4. Кто должен проводить обзор требований? Нарисуйте модель процесса обзора требований.
- 13.5. Существуют ситуации, когда изменения в системе вносятся прежде, чем изменения в требованиях будут одобрены. Предложите модель процесса внесения срочных изменений в систему, который гарантирует со-

гласованность системы и спецификации требований.

- 13.6. Ваша компания использует стандартный метод анализа требований. В процессе работы вы обнаружили, что этот метод не учитывает социальные факторы, важные для системы, которую вы анализируете. Ваш руководитель дал вам ясно понять, какому методу анализа нужно следовать. Обсудите, что вы должны делать в такой ситуации.

14. Управление персоналом

- 14.1. Дайте краткое описание иерархической структуры человеческой памяти. Объясните, почему данной структурой обеспечивается лучшее понимание объектно-ориентированных систем, чем систем, построенных на функциональной декомпозиции.
- 14.2. Каково различие между синтаксическими и семантическими знаниями? Исходя из вашего личного опыта, дайте несколько примеров этих двух типов знания.
- 14.3. Какие факторы прежде всего принимаются во внимание при подборе сотрудников для работы над программным проектом?
- 14.4. Объясните, каким образом доступность информации о ходе разработки проекта и тех технических решениях, которые имеют отношение ко всем членам группы, могут усилить сплоченность группы.
- 14.5. Дайте определение понятия "групповая мысль". Опишите, какие затруднения могут возникать в результате этого явления и как их можно избежать.
- 14.6. Представьте, что вы менеджер и вас попросили спасти проект, от которого зависит финансовый успех или неуспех всей компании. Руководство старшего уровня передало вам незакрытый бюджет, и вам предстоит набрать команду из пяти человек, работающих в данный момент над другими проектами компании. Кроме того, конкурирующая фирма, специализирующаяся в этой же области, активно набирает персонал, и некоторые ваши сотрудники перешли к конкурентам.
- 14.7. Почему открытые и общие помещения менее пригодны для работы команды программистов, чем индивидуальные кабинеты? В каких случаях, по вашему мнению, открытые офисы оказываются более подходящими?
- 14.8. Как вы думаете, порядочно ли схитрить и дать те ответы на вопросы в психологическом тесте, которые работодатель хочет от вас услышать, а не говорить того, что вы на самом деле думаете?