

56

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
ТАДЖИКИСТАН
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКО-ТАДЖИКСКИЙ (СЛАВЯНСКИЙ) УНИВЕРСИТЕТ»**

ЕСТЕСТВЕННОНАУЧНЫЙ ФАКУЛЬТЕТ

Кафедра «Информатика и ИТ»

«Утверждаю»

«25» 10 2023 г.

Зав. кафедрой к.э.н., доцент

Лешукович А.И.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине (модулю)
ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ
09.03.03.- Прикладная информатика
профиль "Инженерия программного обеспечения"

ДУШАНБЕ 2023

ЦЕЛИ И ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Под проектированием информационных систем (ИС) понимается процесс создания в рамках обозначений, правил описания и совокупности документации, предусмотренной стандартами проекта ИС, т.е. однозначно понимаемого описания архитектуры, структуры и алгоритма функционирования. Проектирование ИС осуществляется в соответствии с определенной технологией проектирования, согласно масштабу и особенностям разрабатываемого проекта. Технология проектирования – это совокупность методов и средств проектирования и организации ИС, т.е. методы и средства, используемые в процессе создания и модернизации проекта ИС. В основе технологии проектирования лежит технологический процесс, который определяет действия, их последовательность, состав исполнителей, средства и ресурсы, необходимые для выполнения этих действий. Дисциплина «Проектирование информационных систем» изучает способы конструирования программного обеспечения, разработки структуры информационного обеспечения и эффективного распределения ресурсного обеспечения информационных систем с применением методов визуального моделирования на основе концепции объектно-ориентированной и структурной разработки.

1.1. Цели изучения дисциплины

Целью дисциплины является освоение способов конструирования программ с применением методов структурного и объектно-ориентированного программирования, а также изучение методов визуального моделирования проектируемых информационных систем.

1.2. Задачи изучения дисциплины

Задачи дисциплины формулируются в соответствии с требованиями ФГОС, предъявляемые к компетенциям обучающегося. В результате освоения дисциплины «Проектирование информационных систем» формируются следующие компетенции обучающегося:

Таблица 1.

Р1.3	Результаты освоения	Перечень планируемых результатов	Вид оценочного знания
УК-1	Способностью осуществлять анализ, критический анализ и синтез тех. информации, при выборе системной информации для решения поставленных задач	УК-1.1. Знать принципы сбора, выбора и обработки информации, специфику системного анализа для решения профессиональных задач.	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контроль работы. Устный
		УК-1.2. Уметь анализировать и систематизировать информацию, оценивать эффективность решения поставленных проблем и прогнозировать результаты профессиональной деятельности.	УК-1.3. Оценить возможности научного поиска и практической работы в профессиональной деятельности, методы поиска.

ОПК-4	Способностью участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1. Знать основные стандарты оформления технической документации на различных стадиях жизненного цикла	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.
		ОПК-4.2. Уметь применять стандарты оформления технической документации на различных стадиях жизненного цикла информационной системы.	
		ОПК-4.3. Владеть навыками составления технической документации на различных этапах жизненного цикла информационной системы.	
ОПК-6	Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования	ОПК-6.1. Знать основы теории систем и системного анализа, дискретной математики, теории вероятностей и математической статистики, методов оптимизации и исследования операций, нечетких вычислений, математического и	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.
		ОПК-6.2. Уметь применять методы теории систем и системного анализа, математического, статистического и имитационного моделирования для автоматизации задач принятия решений, анализа информационных потоков, расчета экономической эффективности и надежности	
		ОПК-6.3. Владеет навыками проведения инженерных расчетов основных показателей результативности создания и применения информационных систем и технологий.	
ОПК-8	Способностью принимать участие в управлении проектами создания информации	ОПК-8.1. Знать основные технологии создания и внедрения информационных систем, стандарты управления жизненным циклом информационной	Тестирование. Контроль самостоятельной работы. Отчеты по практическим

	онных систем на стадиях жизненного цикла	<p>ОПК-8.2. Уметь осуществлять организационное обеспечение выполнения работ на всех стадиях и в процессах жизненного цикла информационной</p> <p>ОПК-8.3. Владеть навыками составления плановой и отчетной документации по управлению проектами создания информационных систем на стадиях жизненного цикла.</p>	работам. Контрольная работа. Устный опрос.
ОПК-9	Способностью принимать участие и реализацию профессиональных коммуникаций с заинтересованными участниками проектной деятельности и в рамках проектных групп	<p>ОПК-9.1. Знать инструменты и методы коммуникаций в проектах; каналы коммуникаций в проектах; модели коммуникаций в проектах; технологии межличностной и групповой коммуникации в деловом взаимодействии, отработка</p> <p>ОПК-9.2. Уметь осуществлять взаимодействие с заказчиком и процесс реализации проекта, принимать участие в составлении</p> <p>ОПК-9.3. Владеть навыками проведения презентаций, переговоров, публичных выступлений.</p>	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.

ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
(по дисциплине (модулю))

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

№ п/п	Раздел дисциплины	Формы учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			
		Лек.	Пр.	Лаб.	КСР
	Раздел 1				
1.	Тема 1. Основы проектирования программного обеспечения Современный подход. Декомпозиция задачи. Абстракция через параметризацию. Основные этапы разработки программного обеспечения. ПИИ. Детальное проектирование. Построение модели. Дедуктивный подход. Индуктивный.	2			
2.	ПИИ. Основные этапы разработки программного обеспечения. Построение модели		2		

3.	Лабораторная работа №1. Технология проектирования ИС.			2	
4.	КСР-1. Основные этапы разработки программного обеспечения (ПО).				2
5.	Тема 2. Методы и технологии проектирования ИС. Классификация методов проектирования. Методы проектирования ИС. Концепции и теоретические основы. Нотации, используемые при построении статической структуры и динамического поведения ИС. Процедуры, определяющие практическое применение методов. Оценка результатов.	2			
6.	ПР2. Методы и технологии проектирования ИС.Классификация методов проектирования.		2		
7.	Лабораторная работа №2. Методы проектирования ИС			2	
8.	КСР-2. Жизненный цикл программ				2
9.	Тема 3. Методы быстрой разработки ПО. Подход RAD направлен на быструю разработку приложений)— концепция организации технологического процесса разработки программных продуктов, ориентированная на максимально быстрое получение результата в условиях сильных ограничений по срокам и бюджету и нечётко определённых требований к продукту. Эффект ускорения разработки достигается путём использования соответствующих технических средств и непрерывного, параллельного с ходом разработки, уточнения требований и оценки текущих результатов с привлечением заказчика.	2			
10.	ПР3. Методы быстрой разработки ПО. Подход RAD		2		
11.	Лабораторная работа №3. Методы быстрой разработки ПО-Подход RAD			2	
12.	КСР-3. Построение модели. Дедуктивный подход. Индуктивный подход				2
13.	Тема 4. Проектирование структуры программного обеспечения. операционный и декларативный Операционный подход. Основные операции, исполняемые компьютером. Декларативный подход. Логическое программирование. Особенности языка Пролог. Этапы логической программы. Функциональное программирование. Списки и обработка списков. Логический принцип построение функциональных программ. Функциональное программирование на язык Лисп	2			
14.	ПР4. Проектирование структуры программного обеспечения. Обработка списков		2		
15.	Лабораторная работа №4. Модели жизненного цикла			2	

16.	КСР-4. Инспекция и сквозной просмотр. граничные условия.				2
17.	Тема 5. Проектирование информационного обеспечения ИС. Определение сущности информационного, математического и программного обеспечения. Ознакомление с процессом классификации и кодирования информации. Рассмотрение и характеристика основных принципов проектирования унифицированной системы документации.	2			
18.	ПР5. Проектирование информационного обеспечения ИС. Моделирование данных		2		
19.	Лабораторная работа №5. Проектирование структуры программного обеспечения			2	
20.	КСР-5. Методы проектирования ПО.				2
21.	Тема 6. Структурный подход. Сущность, структурного подхода к разработке ИС заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. При разработке системы "снизу-вверх" от отдельных видов во всей системе целостность.	2			
22.	ПР6. Моделирование потоков данных.		2		
23.	Лабораторная работа №6. Моделирование потоков данных			2	
24.	КСР-6. Структурное проектирование				2
25.	Тема 7. Объектно-ориентированный подход. Методологии проектирования, включающая в себя процесс объектной декомпозиции и приема представления как доменной и физической, так и статической и динамической проектируемой системы. Объектно-ориентированный анализ направлен на создание моделей, более близких к реальности, а методология объектно-ориентированного подхода.	2			
26.	ПР7. Структурный анализ		2		
27.	Лабораторная работа №7. Структурный анализ			2	
28.	КСР-7. Информационное моделирование				2
29.	Тема 8. Классы. Создание классов. Наследование, ассоциативные и полиморфизм	2			

30.	ПР8. Метод функционального моделирования – SADT		2		
31.	Лабораторная работа №8. Метод SADT			2	
32.	КСР-8. Объектно-ориентированное проектирование.				2
	ИТОГО:	16	16	16	16

№ п/п	Раздел дисциплины	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			
		Лек.	Пр.	Лаб.	КСР
6 семестр					
1.	Тема 1. Объектные модели Объектная модель Object Management Group (OMG). Модель объектно-ориентированных систем управления базами данных – Object Database Management Group. Основные этапы создания информационных систем. Вертикальное и горизонтальное проектирование. Функциональность приложения. Масштабность приложения. Модель, реализованная в среде CORBA. Брокер запросов. Система типов CORBA. Создание и уничтожение объектов. Идентификация объектов. Метаданные. Определение динамики поведения объектов. Модули или компоненты	2			
2.	ПР1. Основные понятия объектно - ориентированного подхода		2		
3.	Лабораторная работа №1. Введение в UML			2	
4.	КСР-1. Подход Rapid Application Development (RAD).				2
5.	Тема 2. Универсальный язык моделирования Универсальный язык моделирования Unified Modeling Language – UML. Цели разработки. Стандартизация UML. CASE – средства: Microsoft Visual Modeler. Нотация UML. Диаграммы UML для моделирования. Варианты использования. Последовательность транзакций. Взаимодействие между компонентами системы и взаимодействие между пользователем и системой.	2			
6.	ПР2. Классы. Создание классов. Наследование, встраивание и олиморфизм		2		
7.	Лабораторная работа №2. Диаграмма Use-Case			2	
8.	КСР-2. Диаграммы коммуникации и последовательности				2
9.	Тема 3. Диаграммы UML. В распоряжение проектировщика системы Rational Rose предоставляет следующие типы диаграмм, последовательное создание которых позволяет	2			

	получить полное представление о всей проектируемой системе и об отдельных ее компонентах: Диаграмма прецедентов (Use-case diagram); Диаграмма классов (Class diagram); Диаграмма активностей (Activity diagram); Диаграмма последовательности (Sequence diagram); Диаграмма развертывания (Deployment diagram); Диаграмма сотрудничества (Collaboration diagram); Диаграмма объектов (Object diagram); Диаграмма состояний (Statechart diagram).				
10.	ПР3. Объектные модели UML.		2		
11.	Лабораторная работа №3. Диаграммы последовательности.			2	
12.	КСР-3. Диаграмма активности				2
13.	Тема 4. Статическая структура ИС. Структурные диаграммы. Диаграммы пакетов (Package); классов (Class); объектов (Object); компонентов (Component); распределения (Развертывания Deployment); композитных структур (Composite structure). Структурные диаграммы представляют статическую структуру системы в терминах классов объектно-ориентированного программирования, диаграмма классов относится к логическому представлению системы. Классы решают типы объектов, которыми манипулирует система. На диаграмме классов с помощью специальной системы обозначений изображается некоторый набор классов с описанием их внутренней структуры.	2			
14.	ПР4. Представление истинной формы Use Case View.		2		
15.	Лабораторная работа №4. Диаграмма активности			2	
16.	КСР-4. Диаграмма объекта взаимодействия				2
17.	Тема 5. Диаграммы для описания динамического поведения ИС. Диаграммы прецедентов, использования (Use Case), деятельности (Activity), конечных автоматов (State Machine), описывает функциональные возможности системы. Изображаются внешне субъектом (actor) и их связи с объектами системы (state object) (use case). Диаграмма использования жизни, внешне представляет поведение системы, и не имеет отношения к описанию реализации функциональных возможностей этой системы. С помощью диаграммы использования жизни описывается функциональные требования к конечному продукту. Диаграммы взаимодействия (Interaction), коммуникации (Communication), обмена сообщениями (Message flow diagram), сценариями (Scenarios). Необходимо, чтобы представлено взаимодействие сценария. Присутствие диаграммы взаимодействия сценария, что на ней отмечено	2			

	читается порядок посылки сообщений, а преимущество диаграммы объектов в том, что она лучше подходит для многих объектов со сложными вызовами и допускает включение другой информации: связи, значения атрибутов, роли, блок-схемы и видимость. Так как оба типа диаграмм имеют неоспоримые достоинства, мы пользуемся в нашем методе обоими				
18.	ПР5. Логическое представление (Logical View) .		2		
19.	Лабораторная работа №5. Диаграмма классов			2	
20.	КСР-5. Диаграмма синхронизации				2
	ИТОГО:	10	10	10	10

ВО «Российско-Таджикский (Славянский) университет» Кафедра Информатика и ИТ

**ТЕМЫ РЕФЕРАТОВ И ПИСЬМЕННЫХ РАБОТ
(рефератов, Эссе, письменных работ**

1. Основные этапы разработки программного обеспечения (ПО).
2. Жизненный цикл программ, этапы цикла.
3. Построение модели. Дедуктивный подход. Индуктивный подход.
4. Инспекция и сквозной просмотр. Граничные условия.
5. Методы проектирования ПО.
6. Структурное проектирование.
7. Информационное моделирование
8. Объектно-ориентированное проектирование.
9. Подход Rapid Application Development (RAD).
10. Ограничения в применении подхода RAD.
11. Логическое программирование. Особенности языка Пролог.
12. Функциональное программирование. Списки и обработка списков.
13. Функциональное программирование на язык Лисп.
14. Моделирование потоков данных.
15. Моделирование данных: сущность, связь, атрибут, экземпляр атрибута.
16. Метод функционального моделирования – SADT.
17. Принципиальное различие между структурным и объектно-ориентированным подходом.
18. Основные понятия объектно-ориентированного программирования.

Критерии оценки выполнения самостоятельной работы.

В основу разработки балльно рейтинговой системы положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется постоянно в процессе его обучения в университете. Настоящая система оценки успеваемости студентов основана на использовании совокупности контрольных точек, равномерно расположенных на всем временном интервале изучения дисциплины. При этом предполагается разделение всего курса на ряд более или менее самостоятельных, логически завершенных блоков и модулей и проведение по ним промежуточного контроля.

Студентам выставяются следующие баллы за выполнение задания к ПК:

- **оценка «отлично» (10 баллов):** контрольные тесты, а также самостоятельно выполненные семестровые задания, выполненные полностью и сданные в срок в соответствии с предъявляемыми требованиями;

- **оценка «хорошо» (8-9 баллов):** задание выполнено и в целом отвечает предъявляемым требованиям, но имеются отдельные замечания в его оформлении или сроке сдачи;

- **оценка «удовлетворительно» (6-7 баллов):** задание выполнено не до конца, отсутствуют ответы на отдельные вопросы, имеются отклонения в объеме, содержании, сроке выполнения;

- **оценка «неудовлетворительно» (5 и ниже):** отсутствует решение задачи, задание переписано (скачано) из других источников, не проявлена самостоятельность при его выполнении. Текущий контроль осуществляется в ходе учебного процесса по результатам выполнения самостоятельной работы и контрольной работы.

Основными формами текущего контроля знаний являются:

- обсуждение вынесенных в планах практических занятий лекционного материала и контрольных вопросов; - решение тестов и их обсуждение с точки зрения умения сформулировать выводы, вносить рекомендации и принимать адекватные управленческие решения;

- выполнение контрольной работы и обсуждение результатов;

- участие в дискуссиях в качестве участника и модератора групповой дискуссии по темам дисциплины;

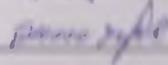
- написание и презентация докладов; - написание самостоятельной (контрольной) работы.

Для контроля усвоения данной дисциплины учебным планом предусмотрен экзамен.

Общее количество баллов по дисциплине - 100 баллов.

Распределение баллов на текущий и промежуточный контроль при освоении дисциплины, а также итоговой оценке представлено ниже.

Составитель:  преподаватель Мирюкаримов О.А.

« 20 »  2021

**МОУ ВО «Российско-Галлийской (Славянской) ушперситет» Кафедра
Информации и ИТ
КОМПЛЕКТ ЗАДАНИЙ ДЛЯ КОНТРОЛЬНОЙ РАБОТЫ по дисциплине
«Проектирование информационных систем»**

Задание 1. (списки и обработка списков)

Задание

1. Составить программу, которая сохраняет автоматическую информацию о наличии автобусов в заданном парке. В списке и каждом автобусе содержатся: номер автобуса; фамилия и инициалы водителя; номер маршрута. Программа должна обеспечивать: начальный форматированный список и файл автобусов и парка в виде списка; при выезде каждого автобуса из парка выводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте; приезде каждого автобуса в парк выводится номер автобуса, и программа удаляет данные об этом автобусе из списка

- автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке; по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.
2. Составить программу, которая содержит текущую информацию о заявках на ж/д билеты. Каждая заявка содержит: пункт назначения; номер рейса; фамилию и инициалы пассажира; желаемую дату отправки. Программа должна обеспечивать: хранение всех заявок в виде списка; добавление заявок в список; удаление заявок; вывод заявок по заданному номеру рейса и дате отправки; вывод всех заявок.
 3. Составить программу, которая содержит динамическую информацию о наличии такси в таксопарке. Сведения о каждом такси содержат: номер автомобиля; фамилию и инициалы водителя; признак того, где находится такси — на маршруте или в парке. Программа должна обеспечивать: начальное формирование данных о всех такси в виде списка; при выезде каждого такси из парка вводится его номер, и программа устанавливает значение признака «такси на рейсе»; при въезде каждого такси в парк вводится его номер автобуса, и программа устанавливает значение признака «в парке»; по запросу выдаются сведения об такси, находящихся в парке, или на рейсе.
 4. Составить программу, отыскивающую проход по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов. Для хранения пути использовать стек.
 5. Гаражная стоянка имеет одну стояночную полосу, причем въезд и выезд находятся в одном конце полосы. Если владелец автомашины приходит забрать свой автомобиль, который не является ближайшим к выходу, то все автомашины, загораживающие проезд, удаляются, машина данного владельца выводится со стоянки, а другие машины возвращаются на стоянку в исходном порядке. Написать программу, которая моделирует процесс прибытия и отъезда машин. Прибытие или отъезд автомашины задается командной строкой, которая содержит признак прибытия или отъезда и номер машины. Программа должна выводить сообщение при прибытии или выезде любой машины. При выезде автомашины со стоянки сообщение должно содержать число раз, которое машина удалялась со стоянки для обеспечения выезда других автомобилей.
 6. В файловой системе каталог файлов организован как линейный список. Для каждого файла в каталоге содержатся следующие сведения: имя файла; дата создания; количество обращений к файлу. Составить программу, которая обеспечивает: начальное формирование каталога файлов; вывод каталога файлов; удаление файлов, дата создания которых меньше заданной; выборку файла с наибольшим количеством обращений. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
 7. Текст помощи для некоторой программы организован как линейный список. Каждая компонента текста помощи содержит термин (слово) и текст, содержащий пояснения к этому термину. Количество строк текста, относящихся к одному термину, от одной до пяти. Составить программу, которая обеспечивает: начальное формирование текста помощи; вывод текста помощи; вывод поясняющего текста для заданного термина. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
 8. Картотека в бюро обмена квартир организована как линейный список. Сведения о каждой квартире содержат: количество комнат; этаж; площадь; адрес. Составить программу, которая обеспечивает: начальное формирование картотеки; ввод заявки на обмен; поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 10% выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в список;

вывод всего списка. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

9. Анкета для опроса населения содержит две группы вопросов. Первая группа содержит сведения о респонденте: возраст; пол; образование (начальное, среднее, высшее). Вторая группа содержит собственно вопрос анкеты, ответ на который либо ДА, либо НЕТ. Составить программу, которая: обеспечивает начальный ввод анкет и формирует из них линейный список; на основе анализа анкет выдает ответы на следующие вопросы: а) сколько мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос анкеты; а) сколько женщин моложе 30 лет, имеющих среднее образование, ответили НЕТ на вопрос анкеты; а) сколько мужчин моложе 25 лет, имеющих начальное образование, ответили ДА на вопрос анкеты; производит вывод всех анкет и ответов на вопросы. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
10. На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована как линейный список. Составить программу, которая: обеспечивает начальное формирование картотеки в виде линейного списка; производит вывод всей картотеки; вводит номер телефона и время разговора; выводит извещение на оплату телефонного разговора. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание 2.

1. Постройте диаграмму для выписки счета за телефон при структурном подходе. Счет выписывается абонентским отделом ГТС и включает начисленную ежемесячную абонентскую плату и сумму за междугородные переговоры, формируемую на основании квитанций. Для начисления абонентской платы используется справочный массив тарифов.
2. Постройте диаграмму для продажи железнодорожных билетов при структурном подходе. Билет выписывается в кассе по запросу клиента, используется информация о графике движения поездов, наличии свободных мест, стоимости билетов.
3. Постройте диаграмму для составления ведомости на выдачу стипендии при структурном подходе. Ведомость составляется в бухгалтерии учебного заведения на основании списков студентов и сведений об успеваемости. Для расчетов используются установленные размеры стипендии.
4. Постройте диаграмму для выписки счета за жилищно-коммунальные услуги при структурном подходе. Счет выписывается бухгалтерией жилищно-коммунального отдела и включает начисленную ежемесячную плату за жилищную площадь, отопление, воду. Для начисления используется справочный массив тарифов за коммунальные услуги и сведения о жильцах.
5. Постройте диаграмму для составления ведомости на выдачу зарплаты при структурном подходе. Ведомость составляется в бухгалтерии предприятия на основании списков сотрудников и их окладов, табеля отработанных каждым сотрудником дней, больничных и отпускных листов. Для расчетов используются установленные ставки налогов и коэффициенты для расчета больничных и отпусков.
6. Постройте диаграмму для выдачи книги в библиотеке. Книга может быть выдана в читальный зал или на абоненте. При этом составляется формуляр, в котором регистрируется состояние книги. Книга может находиться в книгохранилище, в читальном зале, на руках или на реставрации.
7. Постройте диаграмму для выписки счета за отгруженную продукцию при структурном подходе. Выписка счета осуществляется финансовым отделом предприятия на основании накладной. Для расчетов используются справочные массивы по ставкам налогов и ценам на продукцию.

Задание 3.

1. Постройте дерево диаграмм для расхода (выдача в магазины или списание) товара со склада, обслуживающего три магазина, при структурном подходе.
2. Постройте дерево диаграмм для работы кассы предприятия при структурном подходе. Касса осуществляет следующие расчеты:
 - выдача зарплаты;
 - расчеты с подотчетными лицами, включающие выдачу и прием денег от подотчетных лиц;
 - расчеты с банком, включающие перевод наличных денег из кассы в банк и получение наличных сумм из банка.
3. Постройте дерево диаграмм для работы отдела по расчетам за жилищно-коммунальные услуги при структурном подходе. Бухгалтерия осуществляет начисление оплаты за жилплощадь, отопление и воду, прием оплаты от населения за каждый из перечисленных видов коммунальных услуг.
4. Постройте дерево диаграмм для работы отделения Сбербанка при структурном подходе. Сбербанк осуществляет прием вкладов от населения и выдачу денег со счетов, а также выдачу пенсий и пособий.
5. Постройте дерево диаграмм для работы отдела по расчетам за телефон при структурном подходе. Абонентский отдел ГТС начисляет абонентскую плату, выписывает счета за междугородние переговоры, осуществляет оплату счетов за телефон и переговоры.
6. Постройте дерево диаграмм для организации, занимающейся продажей и обменом квартир, при структурном подходе. Организация осуществляет продажу квартир, равноценный обмен или с доплатой, расселение коммунальных квартир, сдачу квартир или комнат в аренду.
7. Постройте дерево диаграмм для задачи 1.7

Задание 4.

1. Постройте структуру SADT-модели для расхода товара со склада, обслуживающего три магазина.
2. Постройте структуру SADT-модели по заданию 2.2.
3. Постройте структуру SADT-модели по заданию 2.3.
4. Постройте структуру SADT-модели по заданию 2.4.
5. Постройте структуру SADT-модели по заданию 2.5.
6. Постройте структуру SADT-модели по заданию 2.6.
7. Постройте структуру SADT-модели по заданию 1.7.

Задание 5.

1. Предметный указатель организован как линейный список. Каждая компонента указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая обеспечивает: начальное формирование предметного указателя; вывод предметного указателя; вывод номеров страниц для заданного слова. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
2. Текст помощи для некоторой программы организован как линейный список. Каждая компонента текста помощи содержит термин (слово) и текст, содержащий пояснения к этому термину. Количество строк текста, относящихся к одному термину, от одной до пяти. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая обеспечивает: начальное формирование текста помощи; вывод текста помощи; вывод поясняющего текста для заданного термина. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

3. Картотека в бюро обмена квартир организована как линейный список. Сведения о каждой квартире содержат: количество комнат; этаж; площадь; адрес. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая обеспечивает: начальное формирование картотеки; ввод заявки на обмен; поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 10% выводится соответствующая карточка и удаляется из списка, в противном случае поступившая заявка включается в список; вывод всего списка. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
4. Англо-русский словарь построен как двоичное дерево. Каждая компонента содержит английское слово, соответствующее ему русское слово и счетчик количества обращений к данной компоненте. Первоначально дерево формируется согласно английскому алфавиту. В процессе эксплуатации словаря при каждом обращении к компоненте в счетчик обращений добавляется единица. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает начальный вход словаря с конкретными значениями счетчиков обращений; формирует новое представление словаря в виде двоичного дерева по следующему алгоритму: а) в старом словаре ищется компонента с наибольшим значением счетчика обращений; б) найденная компонента заносится в новый словарь и удаляется из старого; в) переход к п. а) до исчерпания исходного словаря; производит вывод исходного и нового словарей. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
5. Анкета для опроса населения содержит две группы вопросов. Первая группа содержит сведения о респонденте: возраст, пол, образование (начальное, среднее, высшее). Вторая группа содержит собственно вопрос анкеты, ответ на который либо ДА, либо НЕТ. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает начальный ввод анкет и формирует из них линейный список; на основе анализа анкет выдает ответы на следующие вопросы: а) сколько мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос анкеты; в) сколько женщин моложе 30 лет, имеющих среднее образование, ответили НЕТ на вопрос анкеты; а) сколько мужчин моложе 25 лет, имеющих начальное образование, ответили ДА на вопрос анкеты; производит вывод всех анкет и ответов на вопросы. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
6. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая содержит текущую информацию о книгах в библиотеке. Сведения о книге содержат: номер УДК; фамилию и инициалы автора; название; год издания; количество экземпляров данной книги в библиотеке. Программа должна обеспечивать: начальное формирование данных о всех книгах в библиотеке в виде списка; добавление данных о книгах, вновь поступающих в библиотеку; удаление данных о списываемых книгах; по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.
7. На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована как линейный список. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает начальное формирование картотеки в виде линейного списка; производит вывод всей картотеки; вводит номер телефона и время разговора, выводит извещение на оплату телефонного разговора. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
8. На междугородной телефонной станции картотека абонентов, содержащая сведения о телефонах и их владельцах, организована как двоичное дерево. Составить программу

и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает начальное формирование картотеки в виде двоичного дерева; производит вывод всей картотеки; вводит номер телефона и время разговора; выводит извещение на оплату телефонного разговора. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

9. Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается: номер поезда; станция назначения; время отправления. Данные в информационной системе организованы в виде линейного списка. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает первоначальный ввод данных в информационную систему и формирование линейного списка; производит вывод всего списка; вводит номер поезда и выводит все данные об этом поезде; вводит название станции назначения и выводит данные обо всех поездах, следующих до этой станции. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.
10. Автоматизированная информационная система на железнодорожном вокзале содержит сведения об отправлении поездов дальнего следования. Для каждого поезда указывается: номер поезда; станция назначения; время отправления. Данные в информационной системе организованы в виде двоичного дерева. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая: обеспечивает первоначальный ввод данных в информационную систему и формирование двоичного дерева; производит вывод всего дерева; вводит номер поезда и выводит все данные об этом поезде; вводит название станции назначения и выводит данные о всех поездах, следующих до этой станции. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание 6.

1. Для хранения данных о планшетных сканерах описать запись (структура) вида:
- ```
struct scan_info{
char model[25]; // наименование модели
int price; // цена
double x_size; // горизонтальный размер области сканирования
double y_size; // вертикальный размер области сканирования
int opt; // оптическое разрешение
int grey; // число градаций серого
};
```
- Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Написать функцию, которая извлекает из этого файла данные о сканере в структуру типа `scaninfo`. Обязательный параметр — номер требуемой записи. Функция должна возвращать нулевое значение, если чтение прошло успешно, и -1 в противном случае. Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) — 6-8 записей и выводящей на дисплей данные о запрошенной записи. Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.
2. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит динамическую информацию о наличии автобусов в автобусном парке. Сведения о каждом автобусе содержат: номер автобуса; фамилию и инициалы водителя; номер маршрута. Программа должна обеспечивать: начальное формирование данных о всех автобусах в парке в виде списка; при выезде каждого автобуса из парка вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на

маршруте; при въезде каждого автобуса в парк вводится номер автобуса, и программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке; по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.

3. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах содержат: номер УДК; фамилию и инициалы автора; название; год издания; количество экземпляров данной книги в библиотеке. Программа должна обеспечивать: начальное формирование данных о всех книгах в библиотеке в виде двоичного дерева; добавление данных о книгах, вновь поступающих в библиотеку; удаление данных о списываемых книгах; по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.
4. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит текущую информацию о заявках на авиабилеты. Каждая заявка содержит: пункт назначения; номер рейса; фамилию и инициалы пассажира; желаемую дату вылета. Программа должна обеспечивать: хранение всех заявок в виде списка; добавление заявок в список; удаление заявок; вывод заявок по заданному номеру рейса и дате вылета; вывод всех заявок.
5. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит текущую информацию о заявках на авиабилеты. Каждая заявка содержит: пункт назначения; номер рейса; фамилию и инициалы пассажира; желаемую дату вылета. Программа должна обеспечивать: хранение всех заявок в виде двоичного дерева; добавление и удаление заявок; по заданному номеру рейса и дате вылета вывод заявок с их последующим удалением; вывод всех заявок.
6. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах содержат: номер УДК; фамилию и инициалы автора; название; год издания; количество экземпляров данной книги в библиотеке. Программа должна обеспечивать: начальное формирование данных о всех книгах в библиотеке в виде списка; при взятии каждой книги вводится номер УДК, и программа уменьшает значение количества книг на единицу или выдает сообщение о том, что требуемой книги в библиотеке нет, или требуемая книга находится на руках; при возвращении каждой книги вводится номер УДК, и программа увеличивает значение количества книг на единицу; по запросу выдаются сведения о наличии книг в библиотеке.
7. Составить программу и описать ее модель с помощью диаграммы SADT, которая содержит динамическую информацию о наличии автобусов в автобусном парке. Сведения о каждом автобусе содержат: номер автобуса; фамилию и инициалы водителя; номер маршрута; признак того, где находится автобус — на маршруте или в парке. Программа должна обеспечивать: начальное формирование данных о всех автобусах в виде списка; при въезде каждого автобуса из парка вводится номер автобуса, и программа устанавливает значение признака «автобус на маршруте»; при въезде каждого автобуса в парк вводится номер автобуса, и программа устанавливает значение признака «автобус в парке»; по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.
8. Составить программу и описать ее модель с помощью диаграммы SADT, отображающую проход по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. Программа находит проход через лабиринт, двигаясь от заданного входа. После отыскания прохода программа выводит найденный путь в виде координат квадратов. Для хранения пути

использовать стек.

9. Гаражная стоянка имеет одну стояночную полосу, причем въезд и выезд находятся в одном конце полосы. Если владелец автомашины приходит забрать свой автомобиль, который не является ближайшим к выходу, то все автомашины, загораживающие проезд, удаляются, машина данного владельца выводится со стоянки, а другие машины возвращаются на стоянку в исходном порядке. Написать программу и описать ее модель с помощью диаграммы SADT, которая моделирует процесс прибытия и отъезда машин. Прибытие или отъезд автомашины задается командной строкой, которая содержит признак прибытия или отъезда и номер машины. Программа должна выводить сообщение при прибытии или выезде любой машины. При выезде автомашины со стоянки сообщение должно содержать число раз, которое машина удалялась со стоянки для обеспечения выезда других автомобилей.
10. Составить программу и описать ее модель с помощью диаграммы SADT, моделирующую заполнение гибкого магнитного диска. Общий объем памяти на диске 360 Кбайт. Файлы имеют произвольную длину от 18 байт до 32 Кбайт. В процессе работы файлы либо записываются на диск, либо удаляются с него. В начале работы файлы записываются подряд друг за другом. После удаления файла на диске образуется свободный участок памяти, и вновь записываемый файл либо размещается на свободном участке, либо, если файл не вмещается в свободный участок, размещается после последнего записанного файла. В случае, когда файл превосходит длину самого большого свободного участка, выдается аварийное сообщение. Требование на запись или удаление файла задается в командной строке, которая содержит имя файла, его длину в байтах, признак записи или удаления. Программа должна выдавать по запросу сведения о занятых и свободных участках памяти на диске. Указание: следует создать список занятых участков и список свободных участков памяти на диске.
11. В файловой системе каталог файлов организован как линейный список. Для каждого файла в каталоге содержатся следующие сведения: имя файла; дата создания; количество обращений к файлу. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая обеспечивает: начальное формирование каталога файлов; вывод каталога файлов; удаление файлов, дата создания которых меньше заданной; выборку файла с наибольшим количеством обращений. Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

#### Задание 7.

1. Используя средства объектно-ориентированной среды Delphi, создайте абстрактный класс

2. «TPerson» - человек. Реализуйте инкапсуляцию, с помощью механизма наследования, создайте подклассы «TEmployer»- сотрудники и далее «TEngineer» - инженер. Определите методы.

3. Используя средства объектно-ориентированной среды Delphi, создайте абстрактный класс

4. «TCar» - автомобиль. Реализуйте инкапсуляцию, с помощью механизма наследования, создайте подклассы «TLorry»- грузовой автомобиль и далее «TTipperLorry» - самосвал. Определите методы.

5. Используя средства объектно-ориентированной среды Delphi, создайте абстрактный класс

6. «TDoctor» - врач. Реализуйте инкапсуляцию, с помощью механизма наследования, создайте подклассы «TSurgeon» - хирург и далее «TCardiologist» - кардиохирург. Определите методы.

7. Используя средства объектно-ориентированной среды Delphi, определите положение на шахматной доске. Положение на шахматной доске определяется классом «TPosition». Постройте дочерний класс «TFigure», описывающий шахматную фигуру. Класс должен

обладать методами отражения и удаления фигуры на шахматной доске. Далее создать класс «TKing» - король.

8. На основе объектов, описанных в задании 4.4, описать объект «TPawn» - пешка.

9. Используя средства объектно-ориентированной среды Delphi, создайте абстрактный класс

10. «TCurPos», определяющий позицию курсора на экране. На его основе создать класс для вывода символьной и текстовой информации.

11. Используя средства объектно-ориентированной среды Delphi, создайте абстрактный класс

12. «TGoods» - товары. Реализуйте инкапсуляцию, с помощью механизма наследования, создайте подклассы «TFoods»- продукты и далее «TMilks» - молочные продукты. Определите методы.

### **Задание 8.**

13. Описать класс с помощью нотации UML (диаграммы классов), реализующий стек. Написать программу, использующую этот класс для моделирования T-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

14. Описать класс с помощью нотации UML (диаграммы классов), реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

15. Построить систему классов с помощью нотации UML (диаграммы классов) для описания плоских геометрических фигур: круг, квадрат, прямоугольник. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

16. Построить описание класса с помощью нотации UML (диаграммы классов), содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

17. Составить описание класса с помощью нотации UML (диаграммы классов) для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

18. Составить описание класса с помощью нотации UML (диаграммы классов) для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

19. Составить описание класса с помощью нотации UML (диаграммы классов) прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и

прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

20. Составить описание класса с помощью нотации UML (диаграммы классов) для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

21. Составить описание класса с помощью нотации UML (диаграммы классов) для определения одномерных массивов строк фиксированной длины. Предусмотреть контроль выхода за пределы массива, возможность обращения к отдельным строкам массива по индексам, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, а также вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Составить описание класса с помощью нотации UML (диаграммы классов) многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Задание 9.**

22. Определите основные прецеденты и действующие лица (актеры) для задачи расхода (выдача в магазины или списание) товара со склада, обслуживающего три магазина. Постройте диаграмму UseCase.

23. Определите основные прецеденты и действующие лица (актеры) для задачи выписки счета за телефон. Счет выписывается абонентским отделом ГТС и включает начисленную ежемесячную абонентскую плату и сумму за междугородние переговоры, формируемую на основании квитанций. Для начисления абонентской платы используется справочный массив тарифов. Постройте диаграмму UseCase.

24. Определите основные прецеденты и действующие лица (актеры) для задачи продажи железнодорожных билетов. Билет выписывается в кассе по запросу клиента, на основании информации о графике движения поездов, наличии свободных мест, стоимости билетов. Постройте диаграмму UseCase.

25. Определите основные прецеденты и действующие лица (актеры) для задачи составления ведомости на выдачу стипендии. Ведомость составляется в бухгалтерии учебного заведения на основании списков студентов и сведений об успеваемости. Для расчетов используются установленные размеры стипендии. Постройте диаграмму UseCase.

26. Определите основные прецеденты и действующие лица (актеры) для задачи выписки счета за жилищно-коммунальные услуги. Счет выписывается бухгалтерией жилищно-коммунального отдела и включает начисленную ежемесячную оплату за жилплощадь, отопление, воду и т.д. Для данной операции используется справочный массив тарифов за коммунальные услуги и сведения о жильцах. Постройте диаграмму UseCase.

27. Определите основные прецеденты и действующие лица (актеры) для задачи составления ведомости на выдачу зарплаты при структурном подходе. Ведомость составляется в бухгалтерии предприятия на основании списков сотрудников и их окладов,

табеля отработанных каждым сотрудником дней, больничных и отпускных листов. Для расчетов используются установленные ставки налогов и нормативы для расчетов больничных и отпусков. Постройте диаграмму UseCase.

28. Определите основные прецеденты и действующие лица (актеры) для задачивыдачи книги в библиотеке. Книга может быть выдана в читальный зал или по абонементу. При этом составляется формуляр, в котором регистрируется состояние книги. Книга может находиться в книгохранилище, в читальном зале, на руках или на реставрации.

#### **Задание 10.**

29. Составить описание класса с помощью нотации UML (диаграммы классов) одномерных массивов строк, каждая строка задается длиной и указателем на выделенную для нее память. Предусмотреть контроль выхода за пределы массивов, возможность обращения к отдельным строкам массива по индексам, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, а также вывод на экран элемента массива и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

30. Составить описание класса с помощью нотации UML (диаграммы классов), обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Написать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их. Для этого необходимо реализовать:

- перегруженные операторы присваивания и конкатенации;
- операции сравнения и приведения типов;
- преобразование в число любого типа;
- форматный вывод строки.

31. Написать программу и описать ее функциональность с помощью диаграмм поведения, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

32. Описать класс «записная книжка» с помощью нотации UML (диаграммы классов). Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, и описать ее функциональность с помощью диаграмм поведения демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

33. Описать класс с помощью нотации UML (диаграммы классов) «студенческая группа». Предусмотреть возможность работы с переменным числом  $M$  студентов, поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу и описать ее функциональность с помощью диаграмм поведения, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

34. Описать класс с помощью нотации UML (диаграммы классов), реализующий

тип данных

35. «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами: сложение, вычитание, умножение, деление (+, -, \*, / ) (умножение и деление как на другую матрицу, так и на число); комбинированные операции присваивания (+=, -=, \*=, /=); операции сравнения на равенство/неравенство; операции вычисления обратной и транспонированной матрицы, операцию возведения в степень; методы вычисления детерминанта и нормы; методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметрическая, верхняя треугольная, нижняя треугольная); операции ввода/вывода в стандартные потоки. Написать программу и описать ее функциональность с помощью диаграмм поведения, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

36. Описать класс с помощью нотации UML (диаграммы классов) «множество», позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств. Написать программу и описать ее функциональность с помощью диаграмм поведения, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

37. Описать класс с помощью нотации UML (диаграммы классов), реализующий стек. Написать программу и описать ее функциональность с помощью диаграмм поведения, использующую этот класс для отыскания прохода по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.

38. Описать класс с помощью нотации UML (диаграммы классов) «предметный указатель». Каждая компонента указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя. Написать программу и

39. описать ее функциональность с помощью диаграмм поведения, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## **Задание 11.**

40. Создать класс CFile, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс CMyDataFile — файл, содержащий в себе данные некоторого определенного типа MyData, а также заголовок, облегчающий доступ к этому файлу. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

41. Создать класс CPoint — точка. На его основе создать классы CcoloredPoint и CLine. На основе класса CLine создать класс CColoredLine и класс CPolyLine — многоугольник. Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета. Написать демонстрационную программу, в которой будет использоваться список объектов этих классов в динамической памяти.

42. Создать абстрактный класс CVehicle. На его основе реализовать классы CPlane, CCar и CShip. Классы должны иметь возможность задавать и получать координаты, параметры средств передвижения (цена, скорость, год выпуска). Для самолета должна быть определена высота, для самолета и корабля — количество пассажиров. Для корабля — порт приписки. Написать программу, создающую список объектов этих классов в

динамической памяти. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

43. Описать базовый класс «Элемент». Поля:

- имя элемента (указатель на строку символов);
- количество входов элемента;
- количество выходов элемента. Методы:
- конструктор класса;
- деструктор класса;
- метод, задающий имя элемента.

44. На основе класса «Элемент» описать производный класс «Комбинационный», представляющий собой комбинационный элемент (двоичный вентиль), который может иметь несколько входов и один выход.

Поля:

- указатель, используемый для динамического размещения полей, содержащих значения входов. Методы:
- конструктор;
- конструктор копирования;
- деструктор;
- метод, задающий значение на входах экземпляра класса;
- метод, позволяющий опрашивать состояние отдельного входа экземпляра класса;
- метод, вычисляющий значение выхода (по варианту задания).

45. На основе класса «Элемент» описать производный класс «Память», представляющих собой триггер. Триггер имеет входы, соответствующие типу триггера, и входы установки и сброса. Все триггеры считаются синхронными, сам синхрорход в состав триггера не включается.

Поля:

- массив значений входов объекта класса (задается статически), в массиве учитываются все входы (управляющие и информационные);
- состояние на прямом выходе триггера;
- состояние на инверсном выходе триггера. Методы:
- конструктор (по умолчанию сбрасывает экземпляр класса);
- конструктор копирования;
- деструктор; метод, задающий значение на входах экземпляра класса;
- методы, позволяющие опрашивать состояния отдельного входа экземпляра класса;
- метод, вычисляющий состояние экземпляра класса (по варианту задания) в зависимости от текущего состояния и значений на входах;
- метод, переопределяющий операцию `==` для экземпляров класса.

46. Создать класс «Регистр», используя класс «Память» как включаемый класс. Поля:

- состояние входа «Сброс» — один для экземпляра класса;
- состояние входа «Установка» — один для экземпляра класса;
- статический массив типа «Память» заданной в варианте размерности;
- статический(е) массив(ы), содержащие значения на соответствующих входах элементов массива типа «Память».

Методы:

- метод, задающий значение на входах экземпляра класса (желательно в качестве параметров передавать методу указатели на массивы значений);
- метод, позволяющий опрашивать состояние отдельного выхода экземпляра класса;
- метод, вычисляющий значение нового состояния экземпляра класса.

Все поля классов «Элемент», «Комбинационный» и «Память» должны быть описаны с ключевым словом `private`, или `protected`.

В задании перечислены только обязательные члены и методы класса. Можно задавать дополнительные члены и методы, если они не отменяют обязательные и обеспечивают

дополнительные удобства при работе с данными классами, например, описать функции вычисления выхода/состояния как виртуальные.

47. Описать базовый класс СТРОКА. Обязательные поля класса:

- указатель на char — хранит адрес динамически выделенной памяти для размещения символов строки;
- значение типа int — хранит длину строки в байтах. Обязательные методы:
- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- получение длины строки;
- очистка строки (сделать строку пустой);
- деструктор. Описать производный от СТРОКА класс СТРОКА ИДЕНТИФИКАТОР.

Строки данного класса

строятся по правилам записи идентификаторов в языке C и могут включать в себя только те символы, которые могут входить в состав C-идентификаторов. Если исходные данные противоречат правилам записи идентификатора, то создается пустая СТРОКА\_ИДЕНТИФИКАТОР.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- перевод всех символов строки в верхний регистр;
- перевод всех символов строки в нижний регистр;
- поиск первого вхождения символа в строку;
- деструктор.

48. Переопределить следующие операции:

- присваивание (=);
- сложение (+) — операция конкатенации строк;
- вычитание (-) — из строки (первый операнд) удаляются все символы, входящие в строку — второй операнд, при этом может получиться пустая строка; операция (<) — проверка на больше. Строка считается больше другой, если код символа первой строки в g-й позиции (g изменяется от 0 до gg-1, где n — длина более короткой строки) больше кода символа в той же позиции во второй строке, длины строк могут не совпадать.
- операция (<) — проверка на меньше. Строка считается меньше другой, если код символа первой строки в g-й позиции (z изменяется от 0 до w-1, где n — длина более короткой строки) меньше кода символа в той же позиции во второй строке, длины строк могут не совпадать. Разработчик вправе вводить любое (с обоснованием необходимости) число дополнительных полей и методов.

## Задание 12

1. Написать тестовую программу, которая: динамически выделяет массив указателей на базовый класс; в режиме диалога заполняет этот массив указателями на производные классы, при этом экземпляры производных классов создаются динамически с заданием начальных значений; для созданных экземпляров производных классов выполняет проверку всех разработанных методов с выводом исходных данных и результатов на дисплей. Для конструкторов копирования каждого класса предусмотреть диагностическую печать количества его вызовов в определенное место дисплея (рекомендуется использовать статические члены класса). Режим диалога обеспечивается с

помощью иерархического меню.

2. Описать базовый класс СТРОКА. Обязательные поля класса:

- указатель на char — хранит адрес динамически выделенной памяти для размещения символов строки;
- значение типа int — хранит длину строки в байтах. Обязательные методы:
- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- получение длины строки;
- очистка строки (сделать строку пустой);
- деструктор.

Описать производный от СТРОКА класс БИТОВАЯ\_СТРОКА.

Строки данного класса могут содержать только символы '0' или '1'. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, БИТОВАЯ\_СТРОКА принимает нулевое значение. Содержимое данных строк рассматривается как двоичное число. Отрицательные числа хранятся в дополнительном коде.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом); а конструктор копирования;
- деструктор;
- изменение знака на противоположный (перевод числа в дополнительный код).

49. Переопределить следующие операции (длина строки результата равна длине большей из строк; в случае необходимости более короткая битовая строка расширяется влево знаковым разрядом): присваивание (=); сложение (+) — арифметическая сумма строк; операция (==) — проверка на равенство. Разработчик вправе вводить любое (с обоснованием необходимости) число дополнительных полей и методов.

50. Написать тестовую программу, которая: динамически выделяет массив указателей на базовый класс; в режиме диалога заполняет этот массив указателями на производные классы, при этом экземпляры производных классов создаются динамически с заданием начальных значений; для созданных экземпляров производных классов выполняется проверка всех разработанных методов с выводом исходных данных и результатов на дисплей. Для конструкторов копирования каждого класса предусмотреть диагностическую печать количества его вызовов в определенное место дисплея (рекомендуется использовать статические члены класса).

Режим диалога обеспечивается с помощью иерархического меню. Описать базовый класс СТРОКА. Обязательные поля класса: указатель на char — хранит адрес динамически выделенной памяти для размещения символов строки; значение типа int — хранит длину строки в байтах.

Обязательные методы:

- конструктор без параметров; конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом); конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- получение длины строки; очистка строки (сделать строку пустой); деструктор.

51. Описать производный от СТРОКА класс ДЕСЯТИЧНАЯ\_СТРОКА. Строки данного класса могут содержать только символы десятичных цифр и символы - и +, являющие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, ДЕСЯТИЧНАЯ\_СТРОКА принимает нулевое

значение. Содержимое данных строк рассматривается как десятичное число.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор копирования;
- деструктор;
- метод, определяющий, можно ли представить данное число в формате int;

3. Переопределить следующие операции:

- присваивание (=);
- вычитание (-) — арифметическая разность строк;
- операция (>) — проверка на больше (по значению);
- операция (<) — проверка на меньше (по значению).

Разработчик вправе вводить любое (с обоснованием необходимости) число дополнительных полей и методов.

4. Написать тестовую программу, которая: динамически выделяет массив указателей на базовый класс; в режиме диалога заполняет этот массив указателями на производные классы, при этом экземпляры производных классов создаются динамически с заданием начальных значений; для созданных экземпляров производных классов выполняет проверку всех разработанных методов с выводом исходных данных и результатов на дисплей. Для конструкторов копирования каждого класса предусмотреть диагностическую печать количества его вызовов в определенное место дисплея (рекомендуется использовать статические члены класса). Режим диалога обеспечивается с помощью иерархического меню.

5. Описать базовый класс СТРОКА. Обязательные поля класса: указатель на char — хранит адрес динамически выделенной памяти для размещения символов строки; значение типа int — хранит длину строки в байтах.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор, принимающий в качестве параметра символ;
- конструктор копирования;
- получение длины строки;
- очистка строки (сделать строку пустой);
- деструктор.

52. Описать производный от СТРОКА класс КОМПЛЕКСНОЕ ЧИСЛО. Строки данного класса состоят из двух полей, разделенных символом i. Первое поле задает значение реальной части числа, а второе — мнимой. Каждое из полей может содержать только символы десятичных цифр и символы - и +, задающие знак числа. Символы - или + могут находиться только в первой позиции числа, причем символ + может отсутствовать, в этом случае число считается положительным. Если в составе инициализирующей строки будут встречены любые символы, отличные от допустимых, КОМПЛЕКСНОЕ\_ЧИСЛО принимает нулевое значение. Примеры строк: 331, -7i100, +5i-21.

Обязательные методы:

- конструктор без параметров;
- конструктор, принимающий в качестве параметра C-строку (заканчивается нулевым байтом);
- конструктор копирования;
- деструктор.

6. Переопределить следующие операции:

- присваивание (=);
- операция (==) — проверка на равенство;
- умножение (\*) — умножение чисел.

Разработчик вправе вводить любое (с обоснованием необходимости) число дополнительных полей и методов.

7. Написать тестовую программу, которая: динамически выделяет массив указателей на базовый класс; в режиме диалога заполняет этот массив указателями на производные классы, при этом экземпляры производных классов создаются динамически с заданием начальных значений; для созданных экземпляров производных классов выполняет проверку всех разработанных методов с выводом исходных данных и результатов на дисплей. Для конструкторов копирования каждого класса предусмотреть диагностическую печать количества его вызовов в определенное место дисплея (рекомендуется использовать статические члены класса). Режим диалога обеспечивается с помощью иерархического меню.

### Задание 13

53. Определите основные классы и отношения между ними для задачи расхода (выдача в магазин или списание) товара со склада, обслуживающего три магазина. Постройте структурную схему задачи.

54. Определите основные классы и отношения между ними в деятельности кассы предприятия. Касса осуществляет следующие расчеты: выдачи зарплаты; расчеты с подотчетными лицами, включающие выдачу и прием денег от подотчетных лиц; расчеты с банком, включающие перевод наличных денег из кассы в банк и получение наличных сумм из банка. Постройте структурную схему задачи.

55. Определите основные классы и отношения между ними в деятельности отдела по расчетам за жилищно-коммунальные услуги. Бухгалтерия осуществляет начисление оплаты за жилплощадь, отопление и воду; прием оплаты от населения за каждый из перечисленных видов коммунальных услуг. Постройте структурную схему задачи.

56. Определите основные классы и отношения между ними в деятельности отделения Сбербанка при структурном подходе. Сбербанк осуществляет прием вкладов от населения и выдачу денег со счетов, а также выдачу пенсий и пособий. Постройте структурную схему задачи.

57. Определите основные классы и отношения между ними в деятельности отдела по расчетам за телефон. Абонентский отдел ГТС начисляет абонентскую плату, выписывает счета за междугородние переговоры, осуществляет оплату счетов за телефон и переговоры. Постройте структурную схему задачи.

58. Определите основные классы и отношения между ними в деятельности организации, занимающейся продажей и обменом квартир. Организация осуществляет продажу квартир, равноценный обмен или с доплатой, расселение коммунальных квартир, сдачу квартир или комнат в аренду. Постройте структурную схему задачи.

59. Определите основные классы и отношения между ними для задачи составления ведомости на выдачу зарплат. Ведомость составляется в бухгалтерии предприятия на основании списков сотрудников и их окладов, табеля отработанных каждым сотрудником дней, больничных и отпускных листов. Для расчетов используются установленные ставки налогов и нормативы для расчетов больничных и отпусков. Постройте структурную схему задачи.

### Задание 14

60. 1. Постройте диаграмму последовательности для задачи расхода (выдача в магазин или списание) товара со склада, обслуживающего три магазина. Определите основные сообщения и их типы.

61.

**ТЕСТЫ**  
**ДЛЯ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ ЗНАНИЙ ПО ДИСЦИПЛИНЕ**  
**(ДЛЯ ЗАЧЕТА – ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ)**

**@1. В основе информационной системы лежит:**

- \$A) среда хранения и доступа к данным;
- \$B) вычислительная мощность компьютера;
- \$C) компьютерная сеть для передачи данных;
- \$D) методы обработки информации;
- \$E) протокол передачи гипер;

**@2. Информационные системы ориентированы на:**

- \$A) конечного пользователя, не обладающего высокой квалификацией;
- \$B) программиста;
- \$C) специалиста в области СУБД;
- \$D) руководителя предприятия;
- \$E) сетевой график;

**@3. Неотъемлемой частью любой информационной системы является:**

- \$A) база данных;
- \$B) программа, созданная в среде разработки Delphi;
- \$C) возможность передавать информацию через Интернет;
- \$D) программа, созданная с помощью языка программирования высокого уровня;
- \$E) хранилища и потоки данных;

**@4. В настоящее время наиболее широко распространены системы управления базами данных:**

- \$A) реляционные;
- \$B) иерархические;
- \$C) сетевые;
- \$D) объектно-ориентированные;
- \$E) разностные схемы;

**@5. Более современными являются системы управления базами данных:**

- \$A) постреляционные;
- \$B) иерархические;
- \$C) сетевые;
- \$D) реляционные;
- \$E) разностные схемы;

**@6. СУБД Oracle, Informix, Subase, DB 2, MS SQL Server относятся к:**

- \$A) реляционным;
- \$B) сетевым;
- \$C) иерархическим;
- \$D) объектно-ориентированным;
- \$E) разностные схемы;

**@7. Традиционным методом организации информационных систем является:**

- \$A) архитектура клиент-сервер;
- \$B) архитектура клиент-клиент;
- \$C) архитектура сервер- сервер;
- \$D) размещение всей информации на одном компьютере;
- \$E) разностные схемы;

**@8. Первым шагом в проектировании ИС является:**

- \$A) формальное описание предметной области;
- \$B) построение полных и непротиворечивых моделей ИС;
- \$C) выбор языка программирования;

\$D) разработка интерфейса ИС;

\$E) разностные схемы;

**@9. Модели ИС описываются, как правило, с использованием:**

\$A) языка UML;

\$B) Delphi;

\$C) СУБД;

\$D) языка программирования высокого уровня;

\$E) разностные схемы;

**@10. Для повышения эффективности разработки программного обеспечения применяют:**

\$A) CASE –средства;

\$B) Delphi;

\$C) C++;

\$D) Pascal;

\$E) VBA;

**@11. Под CASE – средствами понимают:**

\$A) программные средства, поддерживающие процессы создания и сопровождения программного обеспечения;

\$B) языки программирования высокого уровня;

\$C) среды для разработки программного обеспечения;

\$D) прикладные программы;

\$E) сетевой график;

**@12. Средством визуальной разработки приложений является:**

\$A) QBASIC;

\$B) Visual Basic;

\$C) Pascal;

\$D) язык программирования высокого;

\$E) сетевой график;

**@13. Microsoft.Net является:**

\$A) платформой;

\$B) языком программирования;

\$C) системой управления базами данных;

\$D) прикладной программой;

\$E) работы и стрелки;

**@14. По масштабу ИС подразделяются на:**

\$A) одиночные, групповые, корпоративные;

\$B) малые, большие;

\$C) сложные, простые;

\$D) объектно- ориентированные и прочие;

\$E) хранилища и потоки данных;

**@15. СУБД Paradox, dBase, Fox Pro относятся к:**

\$A) сетевым;

\$B) локальным;

\$C) корпоративным;

\$D) групповым;

\$E) объектно- ориентированные и прочие;

**@16. СУБД Oracle, DB2, Microsoft SQL Server относятся к:**

\$A) серверам баз данных;

\$B) локальным;

\$C) сетевым;

\$D) посредническим;

\$E) работы и стрелки;

**@17. По сфере применения ИС подразделяются на:**

- \$A) системы обработки транзакций и принятия решений;
- \$B) системы поддержки принятия решений;
- \$C) системы для проведения сложных математических вычислений;
- \$D) экономические системы;
- \$E) мощный компьютер;

**@18. Транзакция это:**

- \$A) передача данных;
- \$B) обработка данных;
- \$C) совокупность операций;
- \$D) преобразование данных;
- \$E) офисные;

**@19. Составление сметы и бюджета проекта, определение потребности в ресурсах, разработка календарных планов и графиков работ относятся к фазе:**

- \$A) подготовки технического предложения;
- \$B) концептуальной;
- \$C) проектирования;
- \$D) разработки;
- \$E) концептуальной;

**@20. Наиболее часто на начальных фазах разработки ИС допускаются следующие ошибки:**

- \$A) ошибки в определении интересов заказчика;
- \$B) неправильный выбор языка программирования;
- \$C) неправильный выбор СУБД;
- \$D) неправильный подбор программистов;
- \$E) международная организация по стандартизации;

**@21. Жизненный цикл ИС регламентирует стандарт ISO/IEC 12207. IEC – это:**

- \$A) международная организация по стандартизации;
- \$B) неправильный выбор языка программирования;
- \$C) международная организация по информационным системам;
- \$D) международная организация по программному обеспечению;
- \$E) разработки и внедрения;

**@22. Согласно стандарту, структура жизненного цикла ИС состоит из процессов:**

- \$A) основных и вспомогательных процессов жизненного цикла и организационных процессов;
- \$B) разработки и внедрения;
- \$C) программирования и отладки;
- \$D) создания и использования ИС;
- \$E) модель параллельной разработки программных модулей;

**@23. Наиболее распространённой моделью жизненного цикла является:**

- \$A) каскадная модель;
- \$B) модель параллельной разработки программных модулей;
- \$C) объектно-ориентированная модель;
- \$D) модель комплексного подхода к разработке ИС;
- \$E) спиральная модель;

## **МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ И ТЕМАТИКА КУРСОВОГО ПРОЕКТИРОВАНИЯ**

Курсовой проект предусмотрен учебным планом изучения дисциплины «Проектирование информационных систем». Она является самостоятельной работой

студента, позволяет оценить качество знаний и отражает приобретенные студентом практические навыки.

Курсовой проект позволяет расширить объем знаний студентов в области проектирования ИС и включает в себя практически все этапы жизненного цикла ИС. Проводится планирование и анализ требований к создаваемой ИС, техническое проектирование, логическое проектирование и реализация (рабочее проектирование, физическое проектирование и программирование с использованием тех или иных технологий).

Курсовой проект является завершающим этапом изучения дисциплины «Проектирование информационных систем» направления подготовки «Прикладная информатика». Выполнение курсового проекта способствует систематизации и обобщению знаний, выработке умения правильно ориентироваться в современных ИС при выборе нужного средства для решения конкретной задачи автоматизации учета.

Защита курсового проекта должна выявить степень подготовленности студента к умению анализировать предметную область, строить модели, определять требования к разрабатываемой базе данных, выбирать и применять конкретное средство для автоматизации информационных процессов.

Цель курсового проекта - приобретение студентом практических навыков по формулированию требований к разрабатываемым информационным системам и построению их моделей, а также формирование навыков самостоятельного практического применения современных методов и средств проектирования программного обеспечения, основанных на использовании визуального проектирования и CASE- средств.

Для реализации данной цели студент должен:

- проявить способность к решению задач автоматизации при организации информационных процессов с целью повышения эффективности учета;
- овладеть навыками постановки задачи по автоматизации отдельных этапов учетного процесса предметной области;
- уметь проводить системный анализ объекта автоматизации учета и на его основе формулировать требования к разрабатываемой информационной системе, адекватные поставленным задачам;
- наиболее полно использовать возможности современных средств информационных технологий и обеспечить разработку прогрессивных организационных решений по поддержанию информационного обеспечения на высоком уровне;
- показать умение применять теоретические положения к решению практических задач, четко формулировать свои мысли и предложения.

В методических рекомендациях приведена тематика курсовых проектов. Вместе с тем тема может быть назначена руководителем курсового проекта или ведущим преподавателем. В том или ином случае тема утверждается на заседании учебно-методической комиссии кафедры.

Перед студентом ставится задача спроектировать модель информационной системы. Для решения поставленной задачи студенту необходимо предварительно ознакомиться с литературой, посвященной теме задания. При этом следует обратить внимание на средства, используемые для решения аналогичных задач или для решения каких-либо ключевых моментов задачи. Этап работы с литературой должен закончиться обзором, в котором собраны полученные сведения из литературы, дан их анализ с точки зрения приложения к поставленной задаче. Результатом выполнения курсового проекта является модель, описывающая различные аспекты проектируемой системы в терминах системы обозначений UML, оформленная в виде проекта системы (программы) и пояснительная записка.

Сформулированные в настоящих указаниях задания на курсовой проект предоставляют студенту простор для творчества. После ознакомления с литературой

студент должен построить концептуальную модель предметной области, для которой разрабатывается модель программы и проектируется ее структура.

При решении задачи необходимо придерживаться техники пошаговой детализации, использовать стандартные структуры, не забывая при этом о развитии инструментальных средств проектирования (CASE-технологии), расширяя возможности проекта за счет включения новых CASE-средств.

При разработке проекта необходимо предусмотреть средства проверки, возможности модификации и реинжиниринга

## **ВЫПОЛНЕНИЕ КУРСОВОГО ПРОЕКТА**

### **1.1 Выбор темы курсового проекта**

Курсовой проект разрабатывается на основе задания, полученного у преподавателя, закрепленного кафедрой в качестве научного руководителя. Студент может предложить свою формулировку темы, но она обязательно должна быть согласована с научным руководителем и одобрена заведующим кафедрой.

Внутри одного потока студентов не разрешается выбор одинаковой темы двумя и более студентами.

### **1.2. Этапы выполнения курсового проекта Подготовительный этап (разработка эскизного и технического проектов).**

1. Практическая работа (разработка рабочего проекта).
2. Оформление пояснительной записки.

При выполнении курсового проекта студенту предлагается:

- провести исследование предметной области (объекта исследования);
- на основе анализа предметной области выбрать подход к проектированию и осуществить постановку задачи.

При применении объектно-ориентированного подхода построить:

- диаграммы вариантов использования (Use-Case);
- диаграммы взаимодействия;
- построить логическую модель информационной системы;
- диаграммы классов;
- построить модель на физическом уровне;
- диаграммы компонентов;
- диаграммы размещения.

При использовании технологии прототипного проектирования (RAD- технология) построить таблицы и определить их взаимосвязи.

## **2. СТРУКТУРА И СОДЕРЖАНИЕ РАЗДЕЛОВ КУРСОВОГО ПРОЕКТА**

### **2.1. Структура курсового проекта**

Внутренняя структура работы должна состоять из введения, разработки модели информационной системы на различных уровнях представления, заключения, списка использованной литературы и приложений. Все этапы разработки проекта отражаются в пояснительной записке. Пояснительная записка состоит из следующих разделов:

- 1) Титульный лист;
- 2) Содержание;
- 3) Введение;
- 4) Теоретическая часть;
- 5) Практическая часть;
- 6) Заключение;
- 7) Список использованной литературы;
- 8) Приложения.

## 2.2. Основные требования к содержанию структурных элементов

Во введении необходимо: обосновать актуальность темы курсового проекта; сформулировать цель работы и поставить задачи, которые необходимо решить для ее достижения ее, кратко представить содержание работы. Здесь же оговаривается объект исследования и аннотация по главам. Введение не должно раскрывать темы курсового проекта, так как оно не является содержательной частью работы. Не следует во введении приводить определение, понятие, состав, роль анализируемых категорий и т.д. Разработку модели информационной системы на различных уровнях представления необходимо начать с постановки задачи. Полная формулировка задания обязательна.

При этом необходимо:

1. Описать предметную область (описание должно быть кратким, но оно должно обязательно содержать характеристику задачи, описание входной и выходной информации). Очень важно определить рамки системы и перечень выполняемых ей функций. Целью подобного исследования является выделение значимых функций для ридриабитнисмой ИС, их согласование, описание в терминах понятных как разработчику, так и будущему пользователю. На этом этапе важно понять смысловое значение данных, обрабатываемых в системе, отделить ключевые понятия предметной области от маловажных и вообще несущественных для рассматриваемого случая.
2. Составить глоссарий проекта, т.е. определить термины и их значения. Дать описание дополнительных спецификаций.
3. Создать модели вариантов использования (определить действующих лиц и вариантыиспользования), составить диаграммы вариантов использования (прецедентов), описать основной и альтернативный потоки событий.
4. Провести архитектурный анализ.
5. Создать диаграммы взаимодействия (диаграмма последовательности и /или кооперации).
6. Создать логическую модель информационной системы (пакеты, классы, диаграммы классов). Построить классы-сущности, управляющие классы, граничные классы. При проектировании граничных классов разобраться и представить их интерфейс.
7. Построить диаграмму компонентов и диаграмму размещения.

Осуществить проектирование реляционной базы данных, построить диаграммы «сущность- связь» и генерацию описания базы данных на тот или иной язык программирования.

Обязательно сделать ссылки на литературные источники, материал которых использовался.

Заключение завершает изложение курсового проекта. В них резюмируются итоги выполненной работы в виде обобщения самых существенных результатов. Выводы должны отражать только содержание проекта, быть краткими, ясно и четко сформулированными.

Список использованной литературы должен включать в себя не менее 5 источников, оформленных в соответствии с общепринятыми правилами оформления библиографического аппарата.

В списке литературы должно быть два раздела:

- I. Нормативно-правовые материалы и стандарты;
- II. Специальная литература (журнальные статьи, учебники, монографии в алфавитном порядке по фамилии авторов).

## 2.3. Требования к оформлению текста работы

Текст курсового проекта печатается на компьютере на одной стороне стандартного листа формата А4 белой писчей бумаги. Объем работы установлен в пределах 20-25 страниц машинописного текста (без учета списка литературы и

приложений).

Общепринятым является следующее расположение текста на странице: размер левого поля - 30 мм, правого - 20 мм, верхнего и нижнего - 25 мм. Текст и другие отпечатанные и вписанные элементы работы должны быть черными, контуры букв и знаков - четкими, без ореолов и затенения, шрифт Times New Roman - 14, интервал - 1,5. Курсив и подчеркивание в работе не допускаются. Названия разделов и параграфов выделяются полужирным шрифтом.

Каждая глава начинается с нового листа (страницы), а параграфы продолжают на той же странице, отступив от названия главы или текста предыдущего параграфа на 20 мм. Подзаголовки в параграфе не допускаются. Нумерация страниц текста проставляется в правом верхнем углу листа. Проставлять номер страницы необходимо с первой страницы введения, на которой ставится номер

«3». После этого нумеруются все страницы, включая приложения.

Все иллюстрации (схемы, графики, диаграммы) обозначаются словом «Рисунок». Рисунки нумеруются арабскими цифрами, начиная с первого. Название дается под рисунком в центре с номером рисунка и выделяется жирным шрифтом. Рисунки могут быть выполнены в цветном виде. Во всех рисунках должны быть проставлены единицы измерения.

Помещенный в курсовой работе цифровой материал рекомендуется оформлять в виде таблиц. Слово «Таблица» печатается вверху справа от текста. Таблицы нумеруются арабскими цифрами, причем также как и у рисунков, нумерация таблиц должна быть сквозной (например: Таблица 1; Таблица 2) Каждая таблица должна иметь название, которое помещается над соответствующей таблицей. Названия таблиц следует выделять полужирным шрифтом.

Цифровые данные в таблицах пишутся строго по классам и разрядам чисел (единицы под единицами, десятки под десятками и т.д.). Если таблица целиком заимствована из одного источника или составлена по нескольким источникам, то под таблицей следует ссылка на источник. Авторство не указывается, если таблица составлена исполнителем курсового проекта на основе первичных материалов. В таблицах можно использовать одинарный интервал, а размер шрифта сократить до 10-

12. Во всех таблицах должны быть проставлены единицы измерения.

При переносе таблицы на следующую страницу курсового проекта над ней размещают слова

«Продолжение таблицы» с указанием ее номера. При этом, пронумеровав графы в начале таблицы, повторяют их нумерацию на следующей странице. Заголовок таблицы при ее переносе не повторяют. Если размер таблицы превышает одну страницу, то она выносится в приложения. Пояснение символов, коэффициентов рекомендуется приводить непосредственно под формулой в той последовательности, в какой они даны в формуле. Значение каждого символа и коэффициента приводится с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия. Формулы должны нумероваться арабскими цифрами. Номер формулы заключают в круглые скобки и помещают на правом поле.

В работе должны быть ссылки на первоисточники. Ссылки приводятся в квадратных скобках после упоминания о них или в конце цитаты. После указания конкретных цифровых данных или цитат в ссылке указываются страницы, на которых помещается использованный материал, и номер источника в списке литературы. Например, [3, С.52], т.е. источник номер 3 в списке использованной литературы, страница 52.

### 3. ПОДГОТОВКА КУРСОВОГО ПРОЕКТА К ЗАЩИТЕ

Оформленный курсовой проект представляется студентом преподавателю в распечатанном и в электронном виде для просмотра не позднее двух недель до начала

сессии. Получив через некоторое время обратно свою работу с отзывом научного руководителя, студент очного отделения начинает готовиться к ее защите, то есть демонстрации знаний темы, умения отстаивать изложенный материал, аргументировать свои выводы и предложения.

Отзыв содержит предварительную оценку, которая может измениться в ту или иную сторону в зависимости от результатов защиты курсовой. При работе с отзывом студент особое внимание должен уделить анализу отмеченных недостатков, методическим советам преподавателя по их устранению, обратив внимание и на постраничные замечания руководителя.

При неудовлетворительной оценке курсовой проект не засчитывается, студент должен полностью переработать курсовой проект по выбранной теме.

#### 4. ОЦЕНКА КУРСОВОГО ПРОЕКТА

Критерии оценки курсового проекта:

- степень усвоения студентом понятий и категорий по теме исследования;
- умение работать с документальными и литературными источниками;
- умение формулировать основные выводы по результатам анализа конкретного материала;
- грамотность и стиль изложения;
- самостоятельность работы, оригинальность в осмыслении материала;
- правильность и аккуратность оформления;
- соответствие оформления курсового проекта установленным требованиям.

Критерии, при наличии хотя бы одного из которых работа оценивается только на "неудовлетворительно". К ним относятся:

- содержание работы не относится к предмету дисциплины;
- работа перепечатана из Интернета, CD-ROM или других носителей информации.

А при следующих недостатках снижаются оценка работы или предоставляется возможность доработать

- неструктурированный план курсового проекта;
- объем работы менее 15 листов машинописного текста;
- в работе отсутствуют ссылки и сноски на нормативные и другие источники;
- оформление курсовой работы не соответствует требованиям (отсутствует нумерация страниц, неверное или неполное оформление библиографии и т.д.).

При оценке письменных курсовых проектов преподаватель обращает также внимание на следующие распространенные ошибки в работах студентов:

- отсутствие четкости в определении основного содержания курсового проекта, убедительных доказательств, обоснований, выводов и рекомендаций;
- нарушение последовательности изложения, частые повторения, нечеткие формулировки, оговорки, грамматические ошибки;
  - излагаемые по тексту примеры не подкреплены смысловым содержанием, размышлениями автора;
- Курсовой проект представляет собой пересказ литературных источников, набор цитат, фраз.

#### ТЕМАТИКА КУРСОВОГО ПРОЕКТИРОВАНИЯ

Создать проект информационной системы по выбранной теме из списка:

1. Проектирование ИС «Автовокзал» с использованием CASE- технологий
2. Проектирование ИС «Адвокатская контора» с использованием CASE-технологий
3. Проектирование ИС «Ателье» с использованием CASE-технологий
4. Проектирование ИС «Фермерское хозяйство» с использованием CASE-технологий

5. Проектирование ИС «Поликлиника» с использованием CASE- технологий
6. Проектирование ИС «Аэропорт» с использованием CASE- технологий
7. Проектирование ИС «Автошкола» с использованием CASE-технологий
8. Проектирование ИС «Супермаркет: автоматизация учета место кассира» с использованием CASE-технологий
9. Проектирование ИС «Управления учебным процессом в режиме функционирования сети» с использованием CASE- технологий
10. Проектирование ИС «Автоматизация продуктового магазина» с использованием CASE- технологий
11. Проектирование ИС «Отдел кадров предприятия» с использованием CASE- технологий
12. Проектирование ИС «Санаторий» с использованием CASE- технологий
13. Проектирование ИС «Бухгалтерия предприятия» с использованием CASE-технологий
14. Проектирование ИС «Авторемонтная мастерская» с использованием CASE- технологий
15. Проектирование ИС «Биржа труда» с использованием CASE-технологий
16. Проектирование ИС «Кафе» с использованием CASE-технологий
17. Проектирование ИС «Гостиница» с использованием CASE- технологий
18. Проектирование ИС «Водозаборная станция» с использованием CASE- технологий
19. Проектирование ИС «Сеть стоянок автотранспорта» с использованием CASE- технологий
20. Проектирование ИС «Мебельный магазин» с использованием CASE-технологий
21. Проектирование ИС «Автоматизация угольного предприятия» с использованием CASE- технологий
22. Проектирование ИС «Магазин бытовой техники» с использованием CASE- технологий
23. Проектирование ИС «Электронный магазин» с использованием CASE-технологий
24. Проектирование ИС «Учёт поставок и продаж в торговой фирме» с использованием CASE- технологий
25. Проектирование ИС «Туристическая фирма» с использованием CASE-технологий
26. «Автоматизация выдачи кредитов в банке» с использованием CASE-технологий
27. Автоматизация рабочего место оператора автотранспортного предприятия с использованием CASE-технологий
28. Проектирование ИС «Компьютерный салон» с использованием CASE-технологий
29. Проектирование ИС «Управления сайтами» с использованием CASE-технологий
30. Проектирование ИС «Охранной пожарной безопасности» с использованием CASE- технологий
31. Проектирование ИС «Рекламное агентство» с использованием CASE- технологий
32. Автоматизация учета животных, птиц, рептилии в зоопарке с использованием CASE- технологий
33. Проектирование ИС «Провайдеры» с использованием CASE- технологий
34. Проектирование ИС «Склад» с использованием CASE-технологий
35. Проектирование ИС «Учёт труда и зарплаты сотрудников торговой фирмы» с использованием CASE-технологий
36. Проектирование ИС «Торговый центр» с использованием CASE- технологий
37. Проектирование ИС «Книжный магазин» с использованием CASE- технологий
38. Проектирование ИС «Больница» с использованием CASE- технологий
38. Проектирование ИС «Управление и автоматизация автопилота» с

