


**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»
Декан естественнонаучного факультета
Лесникович А.И.
« 1 » Сентября 2026 г.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине (модулю)
ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ
Направление подготовки – 10.03.01 «Информационная безопасность»
Профиль – Безопасность компьютерных систем
(по отрасли или в сфере профессиональной деятельности)
Форма подготовки - очная
Уровень подготовки – бакалавриат

ДУШАНБЕ 2026

**ПАСПОРТ
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ
«ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»**

Код компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения	Вид оценочного знания
УК-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	<p>ИУК-3.1. Определяет свою роль в социальном взаимодействии и командной работе, исходя из знания социологии и социальной психологии, методов развития личности этических норм профессионального взаимодействия с коллективом</p> <p>ИУК-3.2. При реализации своей роли в социальном взаимодействии и командной работе учитывает особенности поведения и интересы других участников</p> <p>ИУК-3.3. Осуществляет обмен информацией, знаниями и опытом с членами команды; оценивает статусные позиции других членов команды для достижения поставленной цели</p> <p>ИУК-3.4. Соблюдает нормы и установленные правила внутригруппового взаимодействия; несет личную ответственность за результат</p>	Отчеты по практическим работам. Устный опрос. Презентация
ПК-1	Способен проводить обследование организаций, выявлять информационные потребности пользователей, формировать требования к информационной системе.	<p>ИПК-1.1. Использует методику проведения обследования организации и выявления информационных потребностей пользователей</p> <p>ИПК-1.2. Анализирует деятельности предприятий, и выявляет участки производства, нуждающиеся в автоматизации</p> <p>ИПК-1.3. Осуществляет широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий; теоретическими знаниями о роли компьютерных систем управления информационными потоками; типовыми разработанными средствами защиты информации и возможностями их использования в реальных задачах создания и внедрения информационных систем; навыками выбора класса ИС для автоматизации предприятия в соответствии с требованиями к ИС и ограничениями; способами автоматизации для конкретного</p>	Отчеты по практическим работам. Устный опрос. Презентация

		предприятия; способами выбора ИС на основании преимуществ и недостатков существующих способов; расчета совокупной стоимости владения ИС; способами организации стратегического и оперативного планирования ИС.	
--	--	--	--

ТЕМЫ РЕФЕРАТОВ И ПИСЬМЕННЫХ РАБОТ (рефератов, письменных работ)

1. Основные этапы разработки программного обеспечения (ПО).
2. Жизненный цикл программ: этапы цикла и их характеристики.
3. Методы проектирования программного обеспечения.
4. Структурное проектирование: особенности и примеры применения.
5. Информационное моделирование: сущности, атрибуты, связи.
6. Объектно-ориентированное проектирование (ООП).
7. Подход Rapid Application Development (RAD).
8. Ограничения и недостатки применения RAD.
9. Логическое программирование. Особенности языка Пролог.
10. Функциональное программирование. Работа со списками.
11. Функциональное программирование на языке Лисп.
12. Моделирование потоков данных: DFD-диаграммы.
13. Моделирование данных: сущность, связь, атрибут, экземпляр атрибута.
14. Метод функционального моделирования – SADT.
15. Принципиальное различие между структурным и объектно-ориентированным подходом.
16. Основные понятия объектно-ориентированного программирования: классы, объекты, наследование, инкапсуляция, полиморфизм.
17. UML-нотация: основные типы диаграмм.
18. Современные CASE-средства проектирования: классификация и примеры.
19. Сравнение традиционных и гибких методологий разработки ПО (Waterfall, Agile, Scrum).
20. Роль прототипирования в проектировании программных систем.
21. Тестирование программного обеспечения: виды, методы, уровни.
22. Автоматизация процессов разработки: CI/CD и DevOps-практики.
23. Архитектурные стили программных систем (монолитная архитектура, клиент–сервер, микросервисы).
24. Паттерны проектирования: классификация и примеры (GoF-паттерны).
25. Управление требованиями: виды требований, трассировка и приоритизация.
26. Документация программного обеспечения: стандарты и типы документации.
27. Проектирование интерфейсов пользователя (UI/UX).
28. Инженерия качества ПО: атрибуты качества (ISO/IEC 25010).
29. Управление конфигурациями программного обеспечения (Git и системы контроля версий).
30. Методы управления проектами в разработке ПО (PMBOK, PRINCE2, Kanban).
31. Рефакторинг: цели, приёмы и инструменты.
32. Управление рисками при разработке ПО.
33. Технический долг и его влияние на проект.
34. Применение искусственного интеллекта и машинного обучения в проектировании ПО.
35. Принципы безопасной разработки (DevSecOps).
36. Технологии контейнеризации и оркестрации (Docker, Kubernetes).
37. Средства контроля версий и совместной разработки (GitHub, GitLab, Bitbucket).
38. Автоматизация тестирования: юнит-тестирование, интеграционное и нагрузочное тестирование.
39. Программные фреймворки: назначение, классификация, примеры (Django, Spring, .NET).

40. Инженерия программных процессов (CMMI, SPICE).
41. Роль метрик и КРІ в разработке ПО.
42. Документирование архитектуры программных систем (4+1 View Model).
43. Будущие тенденции разработки ПО: low-code / no-code платформы, генеративный ИИ.

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 выставляются 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 доклада; - 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.

**МОУ ВО «Российско-Таджикский (Славянский) университет»
Кафедра Информатики и ИТ
КОМПЛЕКТ ЗАДАНИЙ ДЛЯ КОНТРОЛЬНОЙ РАБОТЫ по дисциплине
«Проектирование информационных систем»**

Задание 1. (списки и обработка списков)

Задания

1. Составить программу, 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 автобу-

сов, 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.

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 обес-

печивать 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 абонентную плату и сумму за междугородние переговоры, формируемую на основании квитанций. Для начисления абонентной 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 сведениях 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 оплату за жилплощадь, 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 нормативы для расчетов 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 ан-

глийское 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 ввод 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; 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.

6. Составить программу и построить дерево диаграмм (с использованием методов структурного программирования), которая содержит текущую информацию о книгах в библиотеке. Сведения о

7. 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; 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.

8. 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.

9. 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.

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 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.

11. 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 optr: // 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 выезде 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, 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.

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 помещается 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 «TLoggy»- грузовой автомобиль и далее «TTipperLoggy» - 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 «TCardiosurgeon» - 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 со-

здания 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 ежеме-

сячную 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 сведениях 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 задачи выдачи 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 синхровход 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 $g-1$, где g — 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 (g изменяется от 0 до $w-1$, где w — длина более короткой строки) меньше кода символа в той же позиции во второй строке, длины строк могут не совпадать. Раз-

работчик 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 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. 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 +, 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, 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;

1. 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.

2. 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.

3. 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.

4. Переопределить следующие операции:

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

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

5. 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: 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. 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 зарплаты. 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 магазины или списание) товара со склада, обслуживающего три магазина. Определите основные сообщения и их типы.

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

Российско-Таджикский (славянский) университет
Кафедра Информатики и информационных технологий
Экзаменационные билеты по дисциплине «Проектирование ИС»

Билет № 1

1. Основные этапы разработки программного обеспечения (ПО).
2. Взаимодействие объекта с окружающим миром - интерфейс объекта.
3. Задача

Утверждено на заседании кафедры, протокол № 9 от «26» апрель 2025г
Зав. кафедрой и ИТ /Лешукович А.И./

Итоговые оценки студентов Буквенное обозначение итоговых оценок студентов и их цифровые эквиваленты:

Оценка по буквенной системе	Диапазон соответствующих наборных баллов	Численное выражение оценочного балла	Оценка по традиционной системе
A	10	95-100	Отлично
A-	9	90-94	
B+	8	85-89	Хорошо
B	7	80-84	
B-	6	75-79	
C+	5	70-74	Удовлетворительно
C	4	65-69	
C-	3	60-64	
D+	2	55-59	
D	1	50-54	Неудовлетворительно
Fx	0	45-49	
F	0	0-44	

Критерии выведения итоговой оценки промежуточной аттестации:

«Отлично» - средняя оценка $\geq 3,67$.

«Хорошо» - средняя оценка $\geq 2,67$ и $\leq 3,33$.

«Удовлетворительно» - средняя оценка $\geq 1,0$ и $\leq 2,3$

«Неудовлетворительно» - средняя оценка < 0 .