

РОСЖЕЛДОР

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Ростовский государственный университет путей сообщения»

(ФГБОУ ВО РГУПС)

О. В. Игнатъева

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Учебно-методическое пособие
для курсовой работы

Ростов-на-Дону
2021

УДК 004.4(07) + 06

Рецензент – кандидат технических наук, доцент В. В. Жуков

Игнатъева, О. В.

Алгоритмизация и программирование: учебно-методическое пособие для курсовой работы / О. В. Игнатъева; ФГБОУ ВО РГУПС. – Ростов-на-Дону, 2021. – 84 с.

Содержит требования к оформлению и указания по организации выполнения курсовой работы по дисциплине «Алгоритмизация и программирование», а также задания и методику выполнения курсовой работы.

Предназначено для студентов направлений «Информатика и вычислительная техника» и «Информационные системы и технологии» для углубленного изучения программирования на аудиторных занятиях и самостоятельного изучения материала по дисциплине «Алгоритмизация и программирование», а также для всех студентов магистратуры, бакалавриата и специалитета различных направлений, изучающих дисциплины по программированию и спецкурсы.

Одобрено к изданию кафедрой «Вычислительная техника и автоматизированные системы управления».

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 ОБЩИЕ ПОЛОЖЕНИЯ.....	5
2 ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	7
2.1. Цели и задачи курсовой работы	7
2.2 Тематика курсовой работы	7
2.3 Основные этапы выполнения курсовой работы	8
2.4 Руководство и контроль выполнения курсовой работы	8
2.5 Рекомендации по изучению источников	9
2.6 Требования к разрабатываемому программному приложению	9
2.7 Требования к структуре и содержанию пояснительной записки курсовой работы	9
2.8. Планирование времени.....	11
2.9 Исследование предметной области.....	12
2.10 Проектирование приложения	12
2.11 Разработка программного приложения	18
3 ТЕМЫ И ЗАДАНИЯ ДЛЯ КУРСОВОЙ РАБОТЫ.....	20
4 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ	57
4.1 Стандарты, используемые при оформлении пояснительной записки.....	57
5 КОМПЬЮТЕРНАЯ ПРЕЗЕНТАЦИЯ	74
6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ЗАЩИТЕ КУРСОВОЙ РАБОТЫ.....	76
6.1 Критерии оценки курсовой работы.....	76
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	78
Приложение А	80
Приложение Б	82
Приложение В	83

ВВЕДЕНИЕ

Методические указания по организации выполнения и защиты курсовой работы разработаны с учетом требований Федерального государственного образовательного стандарта высшего образования для бакалавриата по направлениям подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.02 «Информационные системы и технологии», Правилами оформления текста курсовой работы (проекта) и дипломной работы (проекта) (РГУПС, 2020).

Методические указания определяют требования к составу, содержанию и оформлению курсовой работы, содержат рекомендации по организации ее выполнения и защиты, включают перечень тем и примерный план курсовой работы, а так же список рекомендуемой литературы.

Выполнение курсовой работы является одним из основных этапов изучения дисциплины «Алгоритмизация и программирование» и имеет целью закрепить и систематизировать знания, полученные студентами в процессе лекционных и практических занятий, развить навыки самостоятельной работы со специальной литературой и методическими материалами.

На выполнение курсовой работы в соответствии с учебным планом отводится 35 часов и 70 часов самостоятельной работы.

Разработка курсовой работы должен производиться с использованием современных средств и методов программирования.

1 ОБЩИЕ ПОЛОЖЕНИЯ

Курсовая работа является основным этапом изучения дисциплины «Алгоритмизация и программирование» и призвана служить систематизации, углублению и закреплению знаний студентом, а также способствовать формированию общепрофессиональных и профессиональных компетенций.

Курсовая работа позволяет дать оценку освоения следующих обще- и профессиональных компетенций

для направления 09.03.01:

- ОПК-1 – Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;
- ОПК-2 – Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;
- ОПК-8 – Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

для направления 09.03.02:

- ОПК-6 – Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
- ОПК-7 – Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем;
- ПК-1 – Способен разрабатывать требования и проектировать программное обеспечение.

Курсовая работа выполняется с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений по дисциплинам «Информатика и программирование» и «Алгоритмизация и программирование»;
- углубления теоретических знаний в соответствии с выбранной темой;
- отработки практических навыков проектирования и программирования программных приложений;
- развития творческой инициативы, самостоятельности, ответственности и организованности;
- формирования умения использовать справочную, нормативную и правовую документацию.

Курсовая работа рассматривается как обязательный вид аудиторной и самостоятельной работы студентов, направленной на формирование соответствующих компетенций. При выполнении и защите курсовой работы студент должен продемонстрировать полноту, глубину, оперативность, системность, осмысленность, прочность и действенность приобретенных знаний. Курсовая работа предусматривает учебные занятия под руководством преподавателя, консультации защиты выполненной работы. Курсовая работа является обязательным элементом процесса подготовки бакалавров. Системой курсовых работ

и проектов студент готовится к выполнению выпускной квалификационной работы.

Основной задачей курсовой работы является проектирование, разработка и тестирование программного обеспечения для информационных и вычислительных систем, имеющего самостоятельное назначение и пояснительную записку к нему.

2 ОРГАНИЗАЦИЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

2.1. Цели и задачи курсовой работы

Целями курсовой работы являются:

- закрепление знаний, полученных в ходе теоретического и практического изучения дисциплины «Алгоритмизация и программирование»;
- приобретение навыков практического программирования с использованием парадигмы структурного и модульного программирования;
- изучение современных систем программирования и сред для разработки программ и систем;
- приобретение навыков использования современных CASE-средств и языков моделирования, используемых при проектировании программ и систем;
- изучение отдельных разделов предметной области, не вошедших в программу теоретического обучения, формирование навыка поиска информации по конкретной теме, ее анализа и использования для решения задачи;
- подготовка к выполнению выпускной квалификационной работы.

Курсовая работа позволяет сформировать способности будущего специалиста к самостоятельному решению практических задач и проблем с использованием теоретических положений, а также знаний и умений, полученных в ходе обучения алгоритмизации и программированию.

Задачей курсовой работы является разработка программного приложения для обработки динамических структур данных в заданной предметной области, включающая в себя:

- описание технического задания;
- алгоритмизацию и проектирование программы;
- генерацию программного кода;
- разработку документации;
- подготовку презентации.

2.2 Тематика курсовой работы

Темы курсовой работы разрабатываются преподавателем с учетом развития современных технологий программирования и утверждаются кафедрой.

Тема курсовой работы выбирается студентом из предложенного преподавателем перечня или предлагается им самостоятельно при условии обоснования ее целесообразности.

Примерная тематика курсовых работ:

- 1) Разработка программного приложения на C++ для обработки динамического двусвязного списка;
- 2) Разработка программного приложения на C++ для обработки бинарного дерева;
- 3) Разработка программного приложения на C++ для обработки направленного графа потока данных.
- 4) Разработка программного приложения на C++ для обработки фрактальных объектов.

В состав курсовой работы могут быть включены научно-исследовательские работы, в которых студент принимал участие по линии учебно-исследовательской работы студентов ФГБОУ ВО РГУПС, в научных учреждениях и на производстве при прохождении практики. Объем этих работ устанавливается руководителями курсовой работы.

В случаях, когда сложность курсовой работы достаточно велика и (или) необходима подробная разработка тем, кафедра имеет право выдавать комплексные задания на курсовую работу группе студентов. Такими заданиями могут быть проекты по созданию перспективных разработок в области языков программирования и сред разработки программного обеспечения.

Темы и задания курсовых работ приведены в разделе 3.

2.3 Основные этапы выполнения курсовой работы

Курсовая работа выполняется и защищается в сроки, определенные учебным графиком. Выполнение курсовой работы состоит из следующих этапов:

- выбор темы и написание рабочего плана курсовой работы;
- оформление и утверждение задания на курсовую работу;
- ознакомление с выбранной темой;
- изучение целей создания программного приложения;
- выявление функциональных требований к приложению;
- составление блок-схемы приложения;
- кодирование и отладка приложения;
- тестирование приложения;
- оптимизация приложения;
- разработка справочной и технической документации;
- написание и оформление пояснительной записки;
- представление завершенной работы и пояснительной записки на проверку преподавателю;
- получение устного отзыва;
- устранение указанных недостатков;
- подготовка курсовой работы к защите;
- защита курсовой работы.

2.4 Руководство и контроль выполнения курсовой работы

Выполнение курсовой работы предполагает консультационную помощь со стороны преподавателя и творческое развитие студентом темы и разделов курсовой работы.

Руководство и контроль хода выполнения студентами курсовой работы осуществляет преподаватель, основными функциями которого являются:

- консультирование по вопросам содержания и последовательности выполнения курсовой работы;
- оказание помощи студенту в ходе выполнения работы;
- контроль хода выполнения курсовой работы;
- проверка курсовой работы и подготовка на него устного отзыва;

- прием курсовой работы.

Консультации по курсовой работе проводятся в пределах объема времени, установленного рабочим учебным планом бакалавра. В ходе консультаций студентам разъясняются назначение и задачи, структура и объем, принципы разработки и оформления пояснительной записки, примерное распределение времени на выполнение отдельных частей курсовой работы, даются ответы на вопросы.

2.5 Рекомендации по изучению источников

Написанию курсовой работы должен предшествовать активный поиск источников, подбор и изучение научно-практической литературы по избранной теме: нормативных документов, материалов веб-сайтов, справочных материалов. Для этого рекомендуется использовать библиотечный каталог университета <http://lib.rgups.ru>, библиографический справочный материал, справочные электронные системы. Изучать источники лучше в порядке обратном хронологическому, т. е. в начале целесообразно изучить самые свежие публикации, затем – прошлых лет.

2.6 Требования к разрабатываемому программному приложению

Для разработки программного приложения должны быть составлены спецификация и разработана блок-схема. В спецификации формулируются основные функциональные требования к приложению, входные и выходные данные, требования к пользовательскому интерфейсу, а также особые требования, обусловленные спецификой приложения либо условиями его эксплуатации.

Программное приложение разрабатывается на языке программирования C++, с использованием универсальной интегрированной среды разработки Visual Studio.

Программное приложение должно быть функционально законченным, отлаженным, протестированным и содержать справочную систему.

Комплект документации на программный продукт должен включать руководство пользователя.

Для приложения должен быть разработан комплект тестов, позволяющий продемонстрировать соответствие разработанной программы.

Программная реализация курсовой работы обязательно прилагается на носителе информации.

2.7 Требования к структуре и содержанию пояснительной записки курсовой работы

Курсовая работа включает в себя:

- 1) демонстрационную версию программного продукта;
- 2) пояснительную записку;
- 3) компьютерную презентацию.

Составными частями пояснительной записки являются:

- 1) титульный лист;

- 2) задание на курсовую работу;
- 3) содержание;
- 4) введение;
- 5) основной текст;
- 6) заключение;
- 7) список использованных источников;
- 8) приложения.

Пояснительная записка выполняется в программе Microsoft Word. Объем пояснительной записки составляет от 20 до 40 страниц текста (с учетом приложений), выполненного в соответствии с требованиями ЕСКД.

Основной текст пояснительной записки состоит из введения, разделов основной части и заключения.

Во введении обосновывается актуальность работы, раскрываются его цель и задачи, приводится краткое описание содержания последующей основной части пояснительной записки. Рекомендуемый объем введения 1–2 страницы.

Основная часть текста включает в себя описание предметной области и постановку задачи, проектную и практическую части. Объем основной части составляет примерно 60% текста пояснительной записки. В ней приводится подробное описание выполненной работы, структурированное по разделам и подразделам. Каждый раздел и подраздел должен иметь свое наименование и отражаться в содержании.

Рекомендуемая структура разделов и подразделов основной части:

- 1 Предметная область
 - 1.1 Анализ задачи
 - 1.2 Требования к разработке
- 2 Проектная часть
 - 2.1 Средства проектирования
 - 2.2 Разработка алгоритма и блок-схемы программы
- 3 Практическая часть
 - 3.1 Структура программного проекта
 - 3.2 Разработка и описание приложения
 - 3.3 Тестирование приложения
 - 3.4 Инструкция пользователя

В разделе «Предметная область» приводится анализ задачи на основании выданного задания, описывается исследование предметной области. На основе изучения литературных источников отечественных и зарубежных авторов рассматривается сущность исследуемой проблемы, анализируются различные подходы их решения, дается их критический анализ, излагается собственная позиция исследователя.

Выявляются требования к разработке, а именно необходимо выбрать среду разработки, описать ее преимущества и недостатки, грамотно, аргументиро-

вано обосновать свой выбор. Разработать структуру будущей программы, выделить ее основные части и продумать правила взаимодействия.

Рекомендуемый объем этого раздела 7–10 страниц.

Раздел «Проектная часть» содержит описание процесса проектирования программного приложения на основе разработки алгоритма ее работы и графического его представления в виде блок-схемы.

Рекомендуемый объем этого раздела 5–7 страниц.

Раздел «Практическая часть» содержит описание разработанного приложения, процесса разработки пользовательского интерфейса, инструкции пользователя и тестирование программного приложения.

Рекомендуемый объем этого раздела до 10 страниц.

Заключение должно содержать выводы по выполнению задания на курсовую работу и соответствовать введению в смысле достижения указанных в нем поставленной цели и задач проектирования. Следует отметить преимущества, связанные с реализацией проектных предложений, отметить недостатки работы, дать практические рекомендации по совершенствованию объекта проектирования, охарактеризовать перспективы дальнейшего развития работы. Рекомендуемый объем раздела 1 страница.

Список использованных источников и количество приложений формально не ограничены. Рекомендуемый объем используемых литературных источников от 5 до 20 наименований. Сведения об использованных источниках должны быть выполнены в соответствии с требованиями ГОСТ 7.1-2003.

Приложение к курсовой работе должно включать листинг программного кода разработанного приложения.

Демонстрационная версия программного продукта предоставляется на компакт-диске, который прилагается к пояснительной записке. Длительность демонстрации программного продукта при защите курсовой работы составляет 5-7 минут.

Для защиты курсовой работы рекомендуется использовать компьютерную презентацию в виде последовательности слайдов. Демонстрация графического и иллюстративного материала в этом случае выполняется на мультимедийном проекторе.

2.8. Планирование времени

Перед началом проектирования следует проанализировать, из каких этапов состоит курсовая работа, оценить сложность этапов и правильно распределить время, отводимое для выполнения курсовой работы, между отдельными этапами.

Можно выделить следующие основные отдельные составляющие части курсовой работы:

- а) исследование предметной области;
- б) разработка алгоритма работы программы;
- в) разработка и оформление блок-схемы по алгоритму программы;
- г) разработка программного приложения;
- д) тестирование программного приложения;
- е) оформление пояснительной записки;
- ж) подготовка презентации и доклада.

На разработку курсовой работы отводится 35 учебных часов.

2.9 Исследование предметной области

В ходе анализа предметной области следует определить цели разработки и круг решаемых задач. Используя литературные и иные источники информации по предметной области, нужно выявить:

- чем является предметная область, какие понятия и термины используются в ее рамках, какие объекты и субъекты составляют ее основу, как взаимодействуют субъекты, как используются объекты и т. п.
- функциональные требования к разрабатываемому программному продукту. В их основе лежат потребности условного заказчика, которые необходимо соотнести с техническими возможностями.

Результатом анализа является составление технического задания, подписываемого заказчиком и исполнителем. В рамках курсовой работы такой документ не подписывается, и составляется как подраздел раздела «Предметная область» пояснительной записки. Этот подраздел должен описывать требования к конечному продукту, очерчивать границы реализуемых функций, определять состав конечного продукта.

2.10 Проектирование приложения

Следующий этап работы соответствует разделу «Проектная часть» пояснительной записки. При проектировании программы одним из важнейших действий является построение алгоритма решения. Используя принципы структурного программирования, здесь следует с помощью декомпозиции и абстрагирования выделить действующие объекты и пути данных при решении задачи, этапы обработки, а также различные применяемые носители данных.

На этом этапе разрабатывается блок-схема программного приложения согласно каскадной модели разработки ПО. Блок-схема – распространенный тип схем (графических моделей), описывающих алгоритмы или процессы, в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой линиями, указывающими направление последовательности. Блок-схема должна отображать взаимодействие программного модуля с внешней средой, например, с пользователем.

Правила выполнения регламентируются ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения». Стандарт в частности регулирует способы построения схем и внешний вид их элементов.

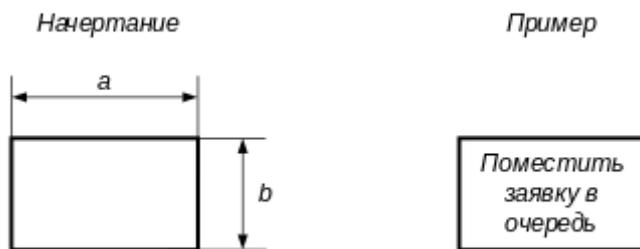
Результатом этапа проектирования является документированное описание разрабатываемого продукта с помощью блок-схемы.

Основные элементы схем алгоритма

При начертании элементов рекомендуется придерживаться строгих размеров, определяемых двумя значениями a и b . Значение a выбирается из ряда 15, 20, 25.. мм. Определение размеров несет рекомендательный характер, однако, стоит отметить, что при соблюдении выполнения размеров блок-схемы имеют более аккуратный вид.

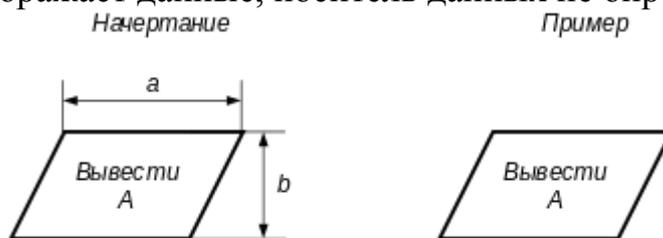
Действие

Символ отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или размещения информации или к определению, по которому из нескольких направлений потока следует двигаться).



Ввод/вывод

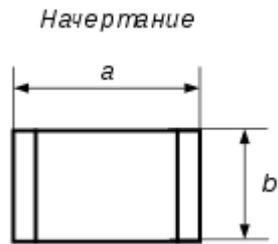
Символ отображает данные, носитель данных не определен.



Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод). Данный символ не определяет носителя данных (для указания типа носителя данных используются специфические символы).

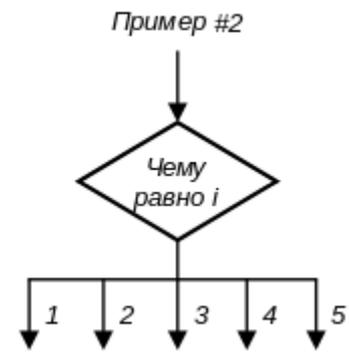
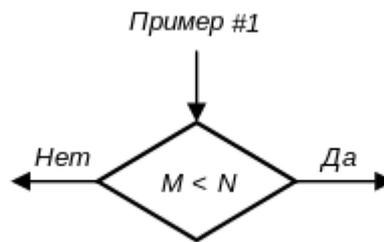
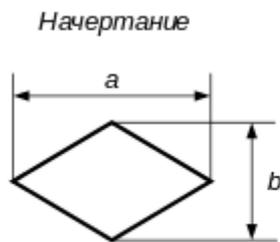
Предопределенный процесс (функция)

Символ отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (в подпрограмме, модуле). Например, в программировании – вызов процедуры или функции.



Условие

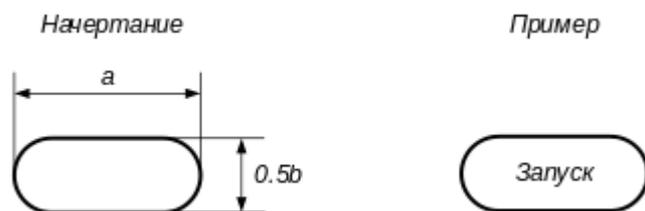
Символ отображает решение или функцию переключательного типа, имеющую один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных внутри этого символа. Соответствующие результаты вычисления могут быть записаны по соседству с линиями, отображающими эти пути.



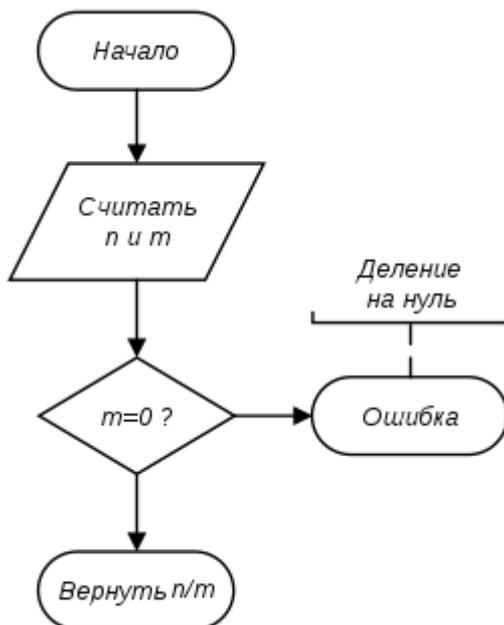
Отображает решение или функцию переключательного типа с одним входом и двумя или более альтернативными выходами, из которых только один может быть выбран после вычисления условий, определенных внутри этого элемента. Вход в элемент обозначается линией, входящей обычно в верхнюю вершину элемента. Если выходов два или три, то обычно каждый выход обозначается линией, выходящей из оставшихся вершин (боковых и нижней). Если выходов больше трех, то их следует показывать одной линией, выходящей из вершины (чаще нижней) элемента, которая затем разветвляется. Соответствующие результаты вычислений могут записываться рядом с линиями, отображающими эти пути. Примеры решения: в общем случае – сравнение (три выхода: $>$, $<$, $=$); в программировании – условные операторы `if` (два выхода: `true`, `false`) и `case` (множество выходов).

Ограничитель

Символ отображает вход из внешней среды и выход во внешнюю среду (начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).

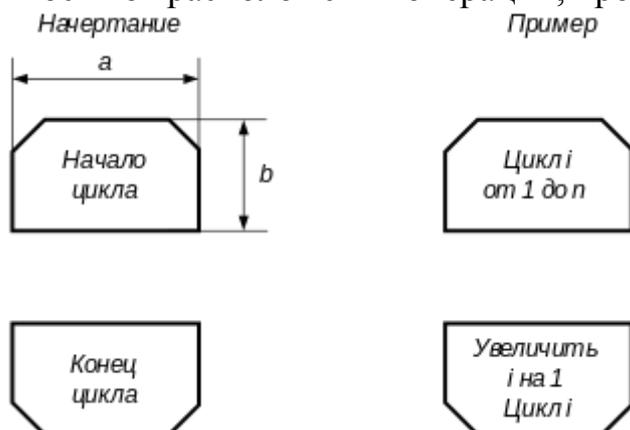


На практике имеют смысл следующие описания ограничителей: начало/конец, запуск/останов, перезапуск (подразумевает перезапуск данной блок-схемы), ошибка (подразумевает завершение алгоритма с ошибкой), исключение (подразумевает исполнение программного исключения)

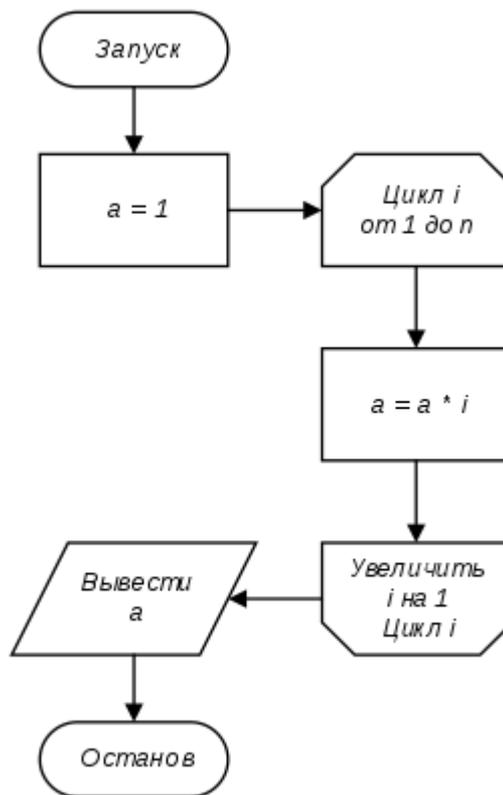


Цикл

Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещаются внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие.

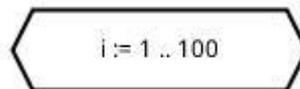


Пример блок-схемы расчета факториала с использованием цикла



Цикл со счетчиком

Символ «подготовка данных» в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком.



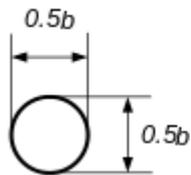
Пример использования блока цикла со счетчиком



Соединитель

Символ отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.

Начертание



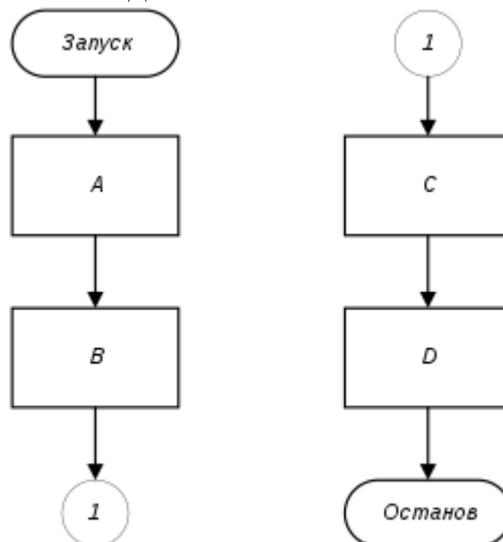
Пример #1



Пример #2



Разделение алгоритма на две части с использованием соединителей.



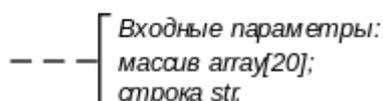
Комментарий

Символ используют для добавления описательных комментариев или пояснительных записей в целях объяснения или примечаний. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обходить группу символов. Текст комментариев или примечаний должен быть помещен около ограничивающей фигуры.

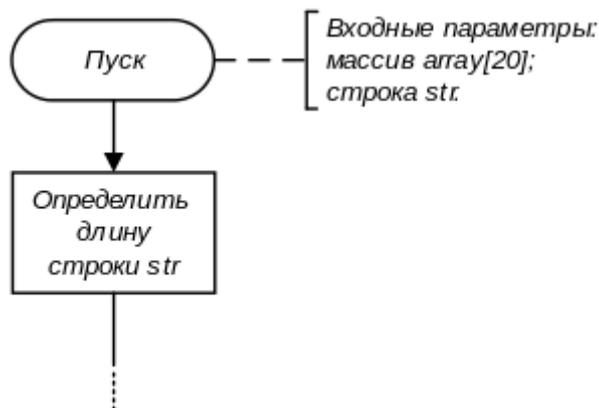
Начертание



Пример



Также символ комментария следует использовать в тех случаях, когда объем текста, помещаемого внутри некоего символа (например, символ процесса, символ данных и др.), превышает размер самого этого символа. Комментарии используют совместно с терминаторами для описания входных аргументов алгоритма при описании функций



Блок-схемы следуют разрабатывать в одной из специальных программ, а именно – MS Visio, Dia, Draw.io, Flowchart Maker или других.

2.11 Разработка программного приложения

Этот этап работы соответствует разделу «Практическая часть» пояснительной записки. В рамках этапа следует написать программу в соответствии с поставленной задачей на языке программирования C++ с получением основных структур данных, сформировать архитектуру программного приложения, определить алгоритмы методов, разработать интерфейс программного продукта, описать инструкцию пользователя.

Следует заметить, что раздел «Практическая часть» пояснительной записки является наиболее объемным и практически важным. Здесь описывается выполнение основной части работы, связанное непосредственно с программированием и формированием структуры программного продукта. Каждое действие в рамках разработки программного приложения должно быть обосновано и задокументировано.

Все действия разработчика в этой части должны выполняться в соответствии с результатами проектирования, а описание разработки должно показывать связь между элементами блок-схемы и элементами программного приложения.

Тексты программного приложения должны быть приведены в приложении. Также программный проект записывается на компакт-диск, который прилагается к пояснительной записке.

3 ТЕМЫ И ЗАДАНИЯ ДЛЯ КУРСОВОЙ РАБОТЫ

Общая тема курсовой работы – «**Программирование динамических структур данных**».

Индивидуальная тема соответствует порядковому номеру студента в академическом журнале группы.

Вначале перечислены индивидуальные темы по порядку, а затем ниже приведены задания для каждой темы.

***Примечание:** для программирования динамических структур данных использовать только парадигму структурного программирования, т.е. применять только структуры данных, а не классы объектно-ориентированного программирования!*

Список индивидуальных тем курсовых работ:

Тема 1. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о телефонном справочнике.

Тема 2. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о автобусах в автобусном парке.

Тема 3. Разработка программного приложения на С++ для обработки динамического односвязного списка, содержащего информацию о кафедрах.

Тема 4. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о прайс-листе электронных товаров.

Тема 5. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о книгах в библиотеке.

Тема 6. Разработка программного приложения на С++ для обработки динамического односвязного списка, содержащего информацию о такси.

Тема 7. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о автомобильном справочнике.

Тема 8. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о заявках на авиабилеты.

Тема 9. Разработка программного приложения на С++ для обработки динамического односвязного списка, содержащего информацию о автобусных перевозках.

Тема 10. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о расписании поездов.

Тема 11. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о товарах в газетном магазине.

Тема 12. Разработка программного приложения на С++ для обработки динамического односвязного списка, содержащего информацию о языках программирования.

Тема 13. Проектирование и разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о книжном архиве.

Тема 14. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о заказах такси.

Тема 15. Разработка программного приложения на С++ для обработки динамического односвязного списка, содержащего информацию о вакцинации.

Тема 16. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о универсальном классификаторе.

Тема 17. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о каталогах файлов в файловой системе.

Тема 18. Разработка программного приложения на С++ для обработки динамического односвязного списка на избирательных участках.

Тема 19. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о списке товаров в магазине.

Тема 20. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию картотек в бюро обмена квартир.

Тема 21. Разработка программного приложения на С++ для обработки динамического односвязного списка – реестра доменных зон.

Тема 22. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о ведомости успеваемости школьников.

Тема 23. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о результатах анкетирования населения.

Тема 24. Разработка программного приложения на С++ для обработки динамического односвязного списка кинофильмов на интернет-портале.

Тема 25. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о справочнике почтовых индексов.

Тема 26. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о предметном указателе.

Тема 27. Разработка программного приложения на С++ для обработки динамического односвязного списка клиентов мобильной сети.

Тема 28. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о прайс-листе канцелярских товаров.

Тема 29. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о лекарствах в аптеке.

Тема 30. Разработка программного приложения на С++ для обработки динамического односвязного списка характеристик промышленных роботов.

Тема 31. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о меню кулинарных блюд.

Тема 32. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию об отправлении поездов дальнего следования.

Тема 33. Разработка программного приложения на С++ для обработки динамического односвязного списка показателей температуры местности.

Тема 34. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о картотеке пользователей.

Тема 35. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию об ассортименте игрушек в магазине.

Тема 36. Разработка программного приложения на С++ для обработки динамического односвязного списка музыкальных произведений.

Тема 37. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о ведомости успеваемости студентов.

Тема 38. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о спортсменах.

Тема 39. Разработка программного приложения на С++ для обработки динамического односвязного списка периферийных устройств.

Тема 40. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о справочнике лекарственных средств.

Тема 41. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о факультетах в университете.

Тема 42. Разработка программного приложения на С++ для обработки динамического односвязного списка учебных лабораторий.

Тема 43. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о расписании авиаперелетов.

Тема 44. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о животных зоопарка.

Тема 45. Разработка программного приложения на С++ для обработки динамического односвязного списка кадрового состава направлений обучения.

Тема 46. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о товарах.

Тема 47. Разработка программного приложения на С++ для обработки динамического двусвязного списка, содержащего информацию о успеваемости студентов.

Тема 48. Разработка программного приложения на С++ для обработки динамического односвязного списка владельцев животных.

Тема 49. Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о продажах товаров.

Тема 50. Разработка программного приложения на C++ для обработки динамического двусвязного списка, содержащего информацию о сотрудниках организации.

Задания к индивидуальным темам курсовой работы:

Задание к теме 1

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это телефонный справочник: номер телефона и фамилия владельца. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или

корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) добавление левого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 10) исключение поддерева из его правой ветви заданного узла исходного дерева.
- 11) выполнить поиск всех телефонов, у которых владельцы однофамильцы.
- 12) вывести количество вершин дерева, значение которых равно числу, введенному пользователем.

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

Задание к теме 2

Составить программу, которая содержит динамическую информацию о наличии автобусов в автобусном парке в виде двусвязного списка. Сведения о каждом автобусе включают:

- номер автобуса;
- фамилию и инициалы водителя;
- номер маршрута.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех автобусов;
- 3) добавление автобуса в начало списка;
- 4) добавление автобуса в конец списка;
- 5) добавление автобуса после определенного автобуса;
- 6) добавление автобуса перед определенным автобусом;
- 7) по запросу выдаются сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте.
- 8) 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;
- 9) 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;
- 10) сохранение данных списка автобусов в текстовый файл.
- 11) извлечение данных из текстового файла и помещение новых автобусов в парк.

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

Задание к теме 3

Составить программу, которая содержит динамическую информацию о кафедрах в виде односвязного списка. Сведения о каждой кафедре включают:

- название кафедры;
- фио заведующего;
- количество преподавателей.
- аудитория.
- корпус.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех кафедр;
- 3) добавление кафедры в начало списка;
- 4) добавление кафедры в конец списка;
- 5) добавление кафедры после определенной кафедры;
- 6) добавление кафедры перед определенной кафедрой;
- 7) удаление кафедры из списка;
- 8) по запросу выдаются сведения о кафедрах определенного корпуса.
- 9) по запросу выдаются сведения о кафедрах в диапазоне введенных пользователей аудиторий.
- 10) по запросу выдаются сведения о кафедре с максимальным количеством преподавателей.

- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла и помещение новых кафедр в список.

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

Задание к теме 4

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это прайс-лист электронных товаров: номер товара и название. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде корень или

левое поддерево правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-корень-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) вывод данных дерева в отсортированном виде.
- 11) вывод дерева в текстовый файл.
- 12) извлечение данных из текстового файла и построение бинарного дерева.
- 13) выполнить поиск всех товаров, название которых начинаются на определенную букву.
- 14) вывести сумму значений всех вершин данного дерева.

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

Задание к теме 5

Составить программу, которая содержит текущую информацию о книгах в библиотеке в виде двусвязного списка. Сведения о книгах включают:

- номер УДК;
- фамилию и инициалы автора;

- название;
- год издания;
- количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление книги в начало списка;
- 3) добавление книги в конец списка;
- 4) добавление книги в позицию, отсортированную по имени в алфавитном порядке;
- 5) добавление книги после определенной книги;
- 6) добавление книги перед определенной книги;
- 7) удаление данных о списываемых книгах;
- 8) сохранение данных списка книг в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых книг.
- 10) по запросу выдаются сведения о наличии книг в библиотеке, упорядоченные по годам издания.
- 11) поиск книг определенного автора и года издания.

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

Задание к теме 6

Составить программу, которая содержит динамическую информацию о наличии такси в виде односвязного списка. Сведения о каждом такси включают:

- номер машины;
- фамилию и инициалы водителя;
- адрес выезда и адрес приезда.
- стоимость поездки.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех такси;
- 3) добавление такси в начало списка;
- 4) добавление такси в конец списка;
- 5) добавление такси после определенного такси;
- 6) добавление такси перед определенным такси;
- 7) удаление такси из списка;
- 8) по запросу выдаются сведения о такси и откуда и куда направляется.
- 9) по запросу выдаются сведения по каждому такси и их выручки.
- 10) по запросу выдаются сведения о такси с максимальным количеством выездов.
- 11) сохранение данных списка такси в бинарный файл.
- 12) извлечение данных из бинарного файла и помещение новых такси в список.

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

Задание к теме 7

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это автомобильный справочник: номер автомобиля и фамилия владельца. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-правое-корень).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) добавление правого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 11) исключение поддерева из левой ветви заданного узла исходного дерева.
- 12) выполнить поиск автомобиля, фамилию владельца которого вводит пользователь.
- 13) вывести количество вершин дерева, являющихся левыми дочерними вершинами (корень дерева не учитывать).

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

Задание к теме 8

Составить программу, которая содержит текущую информацию о заявках на авиабилеты в виде двусвязного списка. Каждая заявка включает:

- пункт назначения;
- номер рейса;
- фамилию и инициалы пассажира;
- желаемую дату вылета.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;

- 2) добавление заявки в начало списка;
- 3) добавление заявки в конец списка;
- 4) добавление заявки после определенной заявки;
- 5) добавление заявки перед определенной заявкой;
- 6) удаление определенной заявки;
- 7) вывод всех заявок на авиабилеты;
- 8) сохранение данных списка заявок в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых заявок в список.
- 10) вывод заявок по заданному номеру рейса и дате вылета;
- 11) вывод заявок по заданному клиенту;
- 12) поиск заявок на полеты в Москву за указанный год.

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

Задание к теме 9

Составить программу, которая содержит динамическую информацию о автобусных перевозках в виде односвязного списка. Сведения о каждой перевозке включают:

- номер автобуса;
- фамилию и инициалы водителя;
- пункт убытия
- пункт прибытия
- время в пути

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех автобусов;
- 3) добавление автобуса в начало списка;
- 4) добавление автобуса в конец списка;
- 5) добавление автобуса после определенного автобуса;
- 6) добавление автобуса перед определенным автобусом;
- 7) по запросу выдаются сведения об автобусах, отправляющихся в определенный город.
- 8) по запросу выдаются сведения об автобусах в отсортированном виде по времени в пути.
- 9) по запросу выдаются сведения об автобусах в интервале времени, введенного пользователем;
- 10) сохранение данных списка автобусов в текстовый файл.
- 11) извлечение данных из текстового файла и помещение новых автобусов в список.

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

Задание к теме 10

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это расписание поездов: дата выезда, пункт отправления, пункт прибытия. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
 - 5) вставка нового узла в дерево.
 - 6) удаление выбранного узла пользователем из дерева.
 - 7) удаление всех узлов дерева,
 - 8) изменение выбранного узла пользователем из дерева.
 - 9) определение числа узлов дерева.
 - 10) вывод данных дерева в отсортированном виде.
 - 11) вывод дерева в текстовый файл.
 - 12) извлечение данных из текстового файла и построение бинарного дерева.
 - 13) выполнить поиск всех поездов, которые приезжают в город, введенный пользователем.
 - 14) вывести количество листьев дерева (листом дерева называется его вершина, не имеющая дочерних вершин).
- Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание к теме 11

Составить программу, которая содержит текущую информацию о товарах в магазине в виде двусвязного списка. Сведения о товарах включают:

- уникальный код товара;
- название товара;
- категорию;
- год производства;
- производитель;
- количество экземпляров данного товара в магазине.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;

- 2) добавление товара в начало списка;
- 3) добавление товара в конец списка;
- 4) добавление нового товара, отсортированного по производителю (при добавлении товара в магазин вводится номер, и программа увеличивает значение количества товара на единицу);
- 5) добавление товара перед указанного товара;
- 6) добавление товара после указанного товара.
- 7) удаление (покупка) выбранного товара.
- 8) вывод всех товаров на экран;
- 9) при покупке каждого товара вводится уникальный номер, и программа уменьшает значение количества товара на единицу или выдает сообщение о том, что требуемого товара в магазине нет;
- 10) по запросу выдаются сведения о наличии товаров в магазине.
- 11) поиск товаров по названию.
- 12) поиск товаров по производителю.
- 13) сохранение данных списка товаров в текстовый файл.
- 14) извлечение данных из текстового файла и помещение новых товаров в список.

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

Задание к теме 12

Составить программу, которая содержит динамическую информацию о языках программирования в виде односвязного списка. Сведения о языке включают:

- название языка программирования;
- поддерживаемая парадигма;
- типизированность.
- рейтинг языка.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех языков;
- 3) добавление языка в начало списка;
- 4) добавление языка в конец списка;
- 5) добавление языка после определенного языка;
- 6) добавление языка перед определенным языком;
- 7) удаление языка из списка;
- 8) по запросу выдаются сведения о языках определенной парадигмы.
- 9) по запросу выдаются сведения о кафедрах отсортированного по рейтингу.
- 10) по запросу создается второй новый односвязный список, содержащий только объектно-ориентированные языки.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 13

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это книжный архив: номер издательства и название книги. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-корень-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) добавление левого поддерева в дерево (для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 10) исключение поддерева из его правой ветви заданного узла исходного дерева.
- 11) выполнить поиск книг по названию.
- 12) вывести сумму значений всех листьев данного дерева (листом дерева называется его вершина, не имеющая дочерних вершин).

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

Задание к теме 14

Составить программу, которая содержит динамическую информацию о такси в виде двусвязного списка. Сведения о каждом такси включают:

- номер такси;
- марка автомобиля;
- фамилию и инициалы водителя;
- признак того, где находится такси – на вызове или в свободное.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление такси в начало списка;
- 3) добавление такси в конец списка;
- 4) добавление такси перед определенным такси;
- 5) добавление такси после определенного такси;
- 6) удаление выбранного такси.
- 7) вывод всех такси;
- 8) сохранение данных списка такси в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых такси в список.
- 10) при выезде каждого такси вводится номер такси, и программа устанавливает значение признака «такси на вызове»;
- 11) при освобождении такси вводится номер такси, и программа устанавливает значение признака «такси свободное»;
- 12) по запросу выдаются сведения о свободных такси, или о такси, находящихся на выезде.

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

Задание к теме 15

Составить программу, которая содержит динамическую информацию о вакцинации в виде односвязного списка. Сведения о языке включают:

- ФИО человека;
- возраст;
- пол.
- название вакцины.
- дата вакцинирования.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех людей;
- 3) добавление человека в начало списка;
- 4) добавление человека в конец списка;
- 5) добавление человека после определенного человека;
- 6) добавление человека перед определенным человеком;
- 7) удаление человека из списка;
- 8) по запросу выдаются сведения о мужчинах, сгруппированных по возрастным группам, прошедших вакцинацию (младше 20 лет, от 20 до 40 лет, от 40 до 60 лет, старше 60 лет).
- 9) по запросу выдаются сведения о людях, прошедших вакцинацию в текущем году.
- 10) по запросу создается второй новый односвязный список, содержащий только информацию о вакцинированных женщинах.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 16

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это универсальный классификатор: код удк и название области науки. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-правое-корень).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) изменение выбранного узла пользователем из дерева.
- 8) определение числа узлов дерева.
- 9) вывод данных дерева в отсортированном виде.
- 10) вывод дерева в текстовый файл.
- 11) извлечение данных из текстового файла и построение бинарного дерева.
- 12) выполнить поиск определенной области по коду удк, введенному пользователем.
- 13) вывести количество листьев дерева, являющихся правыми дочерними вершинами.

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

Задание к теме 17

В файловой системе каталог файлов организован в виде двусвязного списка. Для каждого файла в каталоге содержатся следующие сведения:

- имя файла;
- дата создания;
- количество обращений к файлу.

Написать программу, которая обеспечивает:

- 1) вывод меню для работы со списком;
- 2) добавление файла перед указанным.

- 3) добавление файла после указанного.
- 4) вывод каталога файлов;
- 5) удаление файлов, дата создания которых меньше заданной;
- 6) добавление файла в начало списка;
- 7) сортировка списка по именам файлов;
- 8) сохранение данных списка файлов в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых файлов в список.
- 10) выборку файла с наибольшим количеством обращений.
- 11) выборку файлов с именами начинающихся на определенную комбинацию букв.
- 12) поиск файлов за определенный период времени.

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

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

Задание к теме 18

Составить программу, которая содержит динамическую информацию о избирателях в виде односвязного списка. Сведения о избирателях включают:

- ФИО избирателя;
- адрес проживания
- возраст;
- номер участка.
- год голосования
- голосование (да / нет/ воздерживаюсь).

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех людей;
- 3) добавление человека в начало списка;
- 4) добавление человека в конец списка;
- 5) добавление человека после определенного человека;
- 6) добавление человека перед определенным человеком;
- 7) удаление человека из списка;
- 8) по запросу выдаются сведения об избирателях ответивших положительно, сгруппированных по возрастным группам (младше 30 лет, от 30 до 50 лет, старше 50 лет).
- 9) по запросу выдаются сведения о избирателях, ответивших отрицательно или воздержались в текущем году.
- 10) по запросу создается второй новый односвязный список, содержащий только информацию о избирателях определенного участка.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 19

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это список товаров в магазине: шифр товара и название товара. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) вывод данных дерева в отсортированном виде.
- 9) добавление правого поддерева в дерево (для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 10) исключение поддерева из левой ветви заданного узла исходного дерева.
- 11) выполнить поиск товаров по названию.
- 12) вывести глубину дерева, то есть значение его максимального уровня (например, глубина дерева, состоящего только из корня, равна 0, корень дерева находится на нулевом уровне, его дочерние вершины – на первом уровне и т. д.).

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

Задание к теме 20

Картотека в бюро обмена квартир организована в виде двусвязного списка. Сведения о каждой квартире включают:

- количество комнат;
- этаж;
- площадь;
- адрес.

Написать программу, которая обеспечивает:

- 1) вывод меню для работы со списком;

- 2) добавление квартиры в начало списка;
- 3) добавление квартиры в конец списка;
- 4) добавление новой квартиры перед указанной в список.
- 5) добавление новой квартиры после указанной.
- 6) удаление квартиры по адресу.
- 7) добавление квартиры отсортированной по адресу.
- 8) вывод всех квартир;
- 9) сохранение данных списка квартир в бинарный файл.
- 10) извлечение данных из бинарного файла и помещение новых квартир в список.
- 11) обработка заявки на обмен квартирами.
- 12) поиск в картотеке подходящего варианта: при равенстве количества комнат и этажа и различии площадей в пределах 10% соответствующая карточка выводится и удаляется из списка, в противном случае поступившая заявка включается в список;

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

Задание к теме 21

Составить программу, которая содержит динамическую информацию о реестре доменных имен в виде односвязного списка. Сведения о доменах включают:

- имя домена;
- тип домена;
- уровень домена.
- доменная зона.

Написать программу, которая обеспечивает:

- 1) вывод меню для работы со списком;
- 2) добавление домена в начало списка;
- 3) добавление домена в конец списка;
- 4) добавление нового домена перед указанного в список.
- 5) добавление нового домена после указанного.
- 6) удаление домена.
- 7) добавление домена отсортированной по названию.
- 8) вывод всех доменов;
- 9) по запросу выдаются сведения о доменах, определенной зоны.
- 10) по запросу создается второй новый односвязный список, содержащий только информацию о образовательных доменах edu российской зоны ru.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 22

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это ведомость успеваемости школьников: номер класса, фамилия школьника, суммарная успеваемость в баллах. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-корень-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) вывод данных дерева в отсортированном виде.
- 11) вывод дерева в текстовый файл.
- 12) извлечение данных из текстового файла и построение бинарного дерева.
- 13) выполнить поиск школьников с наибольшей успеваемостью.
- 14) для каждого из уровней данного дерева, начиная с нулевого, вывести количество вершин, находящихся на этом уровне. Считать, что глубина дерева не превосходит 10.

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

Задание к теме 23

Результаты анкетирования населения организованы в виде двусвязного списка. 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, ответом на который может являться либо ДА, либо НЕТ.

Написать программу, которая:

- 1) вывод меню для работы со списком;
- 2) добавление анкеты в начало списка;
- 3) добавление анкеты в конец списка;
- 4) добавление анкеты перед определенной анкетой;
- 5) добавление анкеты после определенной анкеты;
- 6) удаление выбранной анкеты.
- 7) вывод всех анкет;
- 8) сохранение данных списка анкет в бинарный файл.
- 9) извлечение данных из бинарного файла и помещение новых анкет в список.
- 10) на основе анализа анкет выдает ответы на следующие вопросы:
 - а) сколько мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос анкеты;
 - б) сколько женщин моложе 30 лет, имеющих среднее образование, ответили НЕТ на вопрос анкеты;
 - в) сколько мужчин моложе 25 лет, имеющих начальное образование, ответили ДА на вопрос анкеты;
 - г) производит вывод всех анкет и ответов на вопросы.

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

Задание к теме 24

Составить программу, которая содержит текущую информацию о наличии кинофильмов на интернет-портале в виде односвязного списка. Сведения о фильмах включают:

- название фильма;
- жанр;
- год издания;
- количество просмотров.
- продолжительность (мин)

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление фильма в начало списка;
- 3) добавление фильма в конец списка;
- 4) добавление фильма в позицию, отсортированную по имени в алфавитном порядке;
- 5) добавление фильма после определенного фильма;
- 6) добавление фильма перед определенным фильмом;
- 7) удаление данных о фильмах с минимальным количеством просмотров;
- 8) сохранение данных списка фильмов в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых фильмов в конец списка.
- 10) по запросу выдаются сведения о самых популярных фильмах.
- 11) по запросу выдаются сведения о фильмах, сгруппированных по жанрам.

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

Задание к теме 25

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это справочник почтовых индексов: номер индекса и название населенного пункта. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или

корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) добавление левого поддерева в дерево (для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 11) исключение поддерева из его правой ветви заданного узла исходного дерева.
- 12) поиск всех индексов названия населенных пунктов которых начинаются на определенную букву, введенную пользователем.
- 13) для каждого из уровней данного дерева, начиная с нулевого, вывести сумму значений вершин, находящихся на этом уровне. Считать, что глубина дерева не превосходит 10.

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

Задание к теме 26

Предметный указатель организован в виде двусвязного списка. Каждая компонента указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, лежит в диапазоне от одного до десяти.

Написать программу, которая обеспечивает:

- 1) вывод меню для работы со списком;
- 2) добавление нового слова в начало списка.
- 3) добавление нового слова в конец списка.
- 4) добавление нового слова отсортированного по алфавиту.
- 5) добавление нового слова перед указанным словом.
- 6) добавление нового слова после указанного слова.
- 7) удаление указанного слова.
- 8) сортировка всего списка по возрастанию.
- 9) сортировка всего списка по убыванию.
- 10) вывод предметного указателя;
- 11) вывод номеров страниц для заданного слова.
- 12) поиск слова с максимальным количеством номером страниц.
- 13) поиск слов содержащих определенную комбинацию букв (в любой части слова).

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

Задание к теме 27

Составить программу, которая содержит текущую информацию о клиентах мобильной сети в виде односвязного списка. Сведения о клиентах включают:

- ФИО клиента;
- номер мобильного телефона;
- название тарифа;
- сумма денежных средств на телефоне.
- средняя продолжительность (мин) звонков в месяц.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление клиента в начало списка;
- 3) добавление клиента в конец списка;
- 4) добавление клиента в позицию, отсортированного по фамилии в алфавитном порядке;
- 5) добавление клиента после определенного клиента;
- 6) добавление клиента перед определенным клиентом;
- 7) удаление данных о клиентах с минимальным количеством денежных средств;
- 8) сохранение данных списка клиентов в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых клиентов в конец списка.
- 10) по запросу выдаются сведения о клиенте с максимальной продолжительностью звонков.
- 11) по запросу выдаются сведения о клиентах, сгруппированных по тарифам.

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

Задание к теме 28

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это прайс-лист канцелярских товаров: номер товара, название товара и цена. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-корень-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) вывод данных дерева в отсортированном виде.
- 11) добавление нового узла в дерево.
- 12) поиск всех товаров с максимальной ценой.
- 13) вывести значения всех вершин дерева в инфиксном порядке (вначале выводится содержимое левого поддерева, затем выводится значение корня, затем – содержимое правого поддерева).

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

Задание к теме 29

Написать программу, которая содержит текущую информацию о лекарствах в аптеке в виде двусвязного списка. Сведения о лекарствах включают:

- название;
- страна-производитель;
- тип лекарства (таблетки, микстура, уколы и др.)
- год производства;
- количество экземпляров данного лекарства в аптеке.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление нового лекарства в начало списка.
- 3) добавление нового лекарства в конец списка.

- 4) добавление нового лекарства отсортированного по названию в алфавитном порядке.
- 5) добавление нового лекарства перед указанным.
- 6) добавление нового лекарства после указанного.
- 7) удаление данных о лекарствах (при осуществлении покупки);
- 8) вывод всего списка лекарств;
- 9) поиск лекарства с максимальным количеством экземпляров в аптеке.
- 10) по запросу выдаются сведения о наличии лекарств в аптеке, упорядоченные по стране-производителю.
- 11) вывод лекарств сгруппированных по типам.

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

Задание к теме 30

Составить программу, которая содержит текущую информацию о технических характеристиках промышленных роботов в виде односвязного списка. Сведения о роботах включают:

- название промышленного робота;
- грузоподъемность;
- число степеней подвижности;
- линейная скорость.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление робота в начало списка;
- 3) добавление робота в конец списка;
- 4) добавление робота в позицию, отсортированного по названию в алфавитном порядке;
- 5) добавление робота после определенного робота;
- 6) добавление робота перед определенным роботом;
- 7) удаление данных о роботах с минимальным числом степеней подвижности;
- 8) сохранение данных списка роботов в бинарный файл.
- 9) извлечение данных из бинарного файла и помещение новых роботов в начало списка.
- 10) по запросу выдаются сведения о роботе с максимальной скоростью работы.
- 11) по запросу выдаются сведения о роботах, сгруппированных по числу степеней подвижности.

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

Задание к теме 31

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это меню кулинарных блюд: номер блюда,

название, цена. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое – правое - корень).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) добавление правого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 10) исключение поддерева из левой ветви заданного узла исходного дерева.
- 11) добавление нового узла в дерево.
- 12) поиск блюда с минимальной ценой.
- 13) вывести значения всех вершин дерева в префиксном порядке (вначале выводится значение корня, 12 затем – содержимое левого поддерева, затем – содержимое правого поддерева).

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

Задание к теме 32

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 двусвязного списка. Написать программу, которая:

- 1) вывод меню для работы со списком;
- 2) добавление информации об отправлении поезда в начало списка;

- 3) добавление информации об отправлении поезда в конец списка;
- 4) добавление информации об отправлении поезда перед определенным поездом;
- 5) добавление информации об отправлении поезда после определенного поезда;
- 6) удаление выбранного поезда.
- 7) вывод всего списка;
- 8) сохранение данных списка в бинарный файл.
- 9) извлечение данных из бинарного файла и помещение новых поездов в список.
- 10) вводит номер поезда и выводит все данные об этом поезде;
- 11) вводит название станции назначения и выводит данные обо всех поездах, следующих до этой станции.
- 12) выводит отсортированный список по станции назначения в алфавитном порядке.

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

Задание к теме 33

Составить программу, которая содержит текущую информацию о показателях температуры местности в виде односвязного списка. Сведения о показателях включают:

- название местности;
- дата;
- время;
- температура.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление показателя в начало списка;
- 3) добавление показателя в конец списка;
- 4) добавление показателя в позицию, отсортированного по названию местности в алфавитном порядке;
- 5) добавление показателя после определенного показателя;
- 6) добавление показателя перед определенным показателем;
- 7) удаление данных о показателях с минимальным значением температуры;
- 8) сохранение данных списка показателей в бинарный файл.
- 9) извлечение данных из бинарного файла и помещение новых показателей в начало списка.
- 10) по запросу выдаются сведения о показателях с максимальной и минимальной температурой.
- 11) по запросу выдаются сведения о температуре в определенной местности за указанный промежуток времени.

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

Задание к теме 34

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это картотека пользователей: дата рождения, фамилия, место рождения. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) вывод данных дерева в отсортированном виде.
- 11) вывод дерева в текстовый файл.
- 12) извлечение данных из текстового файла и построение бинарного дерева.
- 13) выполнить поиск всех однофамильцев.
- 14) вывести максимальное из значений его вершин и количество вершин, имеющих это максимальное значение.

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

Задание к теме 35

Дан файл записей, содержащий сведения об ассортименте игрушек в магазине в виде двусвязного линейного списка.

Структура записи:

- название игрушки,
- цена,
- количество,
- возрастные границы, например от 2 до 5 лет.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление новой игрушки в начало списка;
- 3) добавление новой игрушки в конец списка;

- 4) добавление игрушки перед определенной игрушкой;
- 5) добавление игрушки после определенной игрушки;
- 6) удаление выбранной игрушки.
- 7) вывод всех игрушек магазина;
- 8) сохранение данных списка игрушек в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых игрушек в список.
- 10) вывести названия игрушек, которые подходят детям от 3 до 5 лет и стоят не больше X рублей.
- 11) получить суммарную стоимость всех имеющихся в магазине кукол.
- 12) найти игрушки количество которых больше числа введенного пользователем.

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

Задание к теме 36

Составить программу, которая содержит о музыкальных произведениях в виде односвязного списка. Сведения о произведениях включают:

- название музыкального произведения;
- композитор;
- год издания;
- продолжительность (мин).

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление произведения в начало списка;
- 3) добавление произведения в конец списка;
- 4) добавление произведения в позицию, отсортированную по имени в алфавитном порядке;
- 5) добавление произведения после определенного произведения;
- 6) добавление произведения перед определенным произведением;
- 7) удаление данных произведения с минимальной продолжительностью;
- 8) сохранение данных списка произведений в текстовый файл.
- 9) извлечение данных из текстового файла и помещение новых произведений в начало списка.
- 10) по запросу выдаются сведения о произведениях определенного композитора.
- 11) по запросу выдаются сведения о фильмах, сгруппированных по авторам.

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

Задание к теме 37

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это ведомость успеваемости студентов: номер студента в группе, фамилия студента, оценка. Данные дерева формировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-корень-правое).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) определение числа узлов дерева.
- 8) вывод данных дерева в отсортированном виде.
- 9) добавление левого поддерева в дерево (для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
- 10) исключение поддерева из его правой ветви заданного узла исходного дерева.
- 11) выполнить поиск студентов, которые сдали экзамен на отлично.
- 12) вывести минимальное из значений всех его вершин и количество листьев, имеющих это минимальное значение (данное количество может быть равно 0).

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

Задание к теме 38

Составить программу, которая содержит информацию о спортсменах в виде двусвязного списка. Структура записи:

- фамилия,
- пол,
- вид спорта,
- год рождения,
- рост.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление данных о новом спортсмене в начало списка;
- 3) добавление данных о новом спортсмене в конец списка;
- 4) добавление данных о новом спортсмене в позицию, отсортированную по имени в алфавитном порядке;

- 5) добавление данных о новом спортсмене после определенного спортсмена;
 - 6) добавление данных о новом спортсмене перед определенным спортсменом;
 - 7) удаление данных о спортсмене;
 - 8) сохранение данных списка данных о спортсменах в текстовый файл.
 - 9) извлечение данных из текстового файла и помещение новых данных о спортсменах в список.
 - 10) найти самого молодого спортсмена, занимающегося плаванием, среди мужчин.
 - 11) вывести сведения о спортсменках – женщинах, выступающих в возрастной категории от 20 до 30 лет.
 - 12) по запросу выдаются сведения о спортсменах, упорядоченные по фамилиям.
- Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание к теме 39

Составить программу, которая содержит динамическую информацию о периферийных устройствах в виде односвязного списка. Сведения о устройстве включают:

- название устройства;
- тип устройства;
- страна производитель;
- год производства.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех периферийных устройств;
- 3) добавление устройства в начало списка;
- 4) добавление устройства в конец списка;
- 5) добавление устройства после определенного устройства;
- 6) добавление устройства перед определенным устройством;
- 7) удаление устройства из списка;
- 8) по запросу выдаются сведения о устройствах определенного типа
- 9) по запросу выдаются сведения о устройствах, отсортированных по годам производства.
- 10) по запросу создается второй новый односвязный список, содержащий только мониторы.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 40

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это справочник лекарственных средств: цена лекарства и название лекарства. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева минимальной высоты;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (левое-правое-корень).
- 5) вставка нового узла в дерево.
- 6) удаление выбранного узла пользователем из дерева.
- 7) удаление всех узлов дерева,
- 8) изменение выбранного узла пользователем из дерева.
- 9) определение числа узлов дерева.
- 10) вывод данных дерева в отсортированном виде.
- 11) вывод дерева в текстовый файл.
- 12) извлечение данных из текстового файла и построение бинарного дерева.
- 13) выполнить поиск лекарства, название которого вводит пользователь.
- 14) вывести количество вершин дерева, являющихся левыми дочерними вершинами (корень дерева не учитывать).

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

Задание к теме 41

Составить программу, которая содержит информацию о факультетах университета в виде двусвязного списка.

Структура записи:

- название факультета,
- номер аудитории,
- номер корпуса,
- количество студентов,
- ФИО декана.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление нового факультета в начало списка;

- 3) добавление нового факультета в конец списка;
- 4) добавление нового факультета после определенного факультета;
- 5) добавление нового факультета перед определенным факультетом;
- 6) удаление данных о выбранном факультете;
- 7) сохранение данных списка факультетов в бинарный файл;
- 8) извлечение данных из бинарного файла и помещение новых факультетов в список;
- 9) по запросу выдаются сведения о факультетах расположенных в первом корпусе на третьем этаже;
- 10) по запросу рассчитывается общее количество студентов со всех факультетов;
- 11) по запросу выдаются сведения о факультете с наименьшим и наибольшим количеством студентов.

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

Задание к теме 42

Составить программу, которая содержит динамическую информацию о учебных лабораториях в виде односвязного списка. Сведения о лаборатории включают:

- название лаборатории;
- аудитория;
- перечень дисциплин, читаемых в лаборатории;
- перечень оборудования.
- количество посадочных мест.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех лабораторий;
- 3) добавление лаборатории в начало списка;
- 4) добавление лаборатории в конец списка;
- 5) добавление лаборатории после определенной лаборатории;
- 6) добавление лаборатории перед определенной лабораторией;
- 7) удаление лаборатории из списка;
- 8) по запросу выдаются сведения о лабораториях определенного типа;
- 9) по запросу выдаются сведения о лабораториях, отсортированных по кол-ву посадочных мест.
- 10) по запросу создается второй новый односвязный список, содержащий только лаборатории, читаемые в определенных дисциплинах.
- 11) сохранение данных списка в бинарный файл.
- 12) извлечение данных из бинарного файла.

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

Задание к теме 43

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это расписание авиаперелетов: номер самолета, дата вылета, пункт отправления, пункт прибытия. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

- 1) построение бинарного дерева поиска;
- 2) создание пустого дерева,
- 3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

- 4) вывод данных путем обхода дерева в порядке (корень-левое-правое).
 - 5) вставка нового узла в дерево.
 - 6) удаление выбранного узла пользователем из дерева.
 - 7) удаление всех узлов дерева,
 - 8) изменение выбранного узла пользователем из дерева.
 - 9) определение числа узлов дерева.
 - 10) вывод данных дерева в отсортированном виде.
 - 11) добавление правого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).
 - 12) исключение поддерева из левой ветви заданного узла исходного дерева.
 - 13) вывод дерева в текстовый файл.
 - 14) извлечение данных из текстового файла и построение бинарного дерева.
 - 15) выполнить поиск всех самолетов, которые прилетают в город, введенный пользователем.
 - 16) вывести количество листьев дерева (листом дерева называется его вершина, не имеющая дочерних вершин).
- Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание к теме 44

Составить программу, которая содержит ведения о животных зоопарка в виде двусвязного списка.

Запись состоит из полей:

- название животного;

- природная зона;
- затраты на корм за один день.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
 - 2) добавление нового животного в начало списка;
 - 3) добавление нового животного в конец списка;
 - 4) добавление нового животного в позицию, отсортированную по имени в алфавитном порядке;
 - 5) добавление нового животного после определенного животного;
 - 6) добавление нового животного перед определенным животным;
 - 7) удаление данных о животном;
 - 8) сохранение данных списка животных в текстовый файл;
 - 9) извлечение данных из текстового файла и помещение новых животных в список;
 - 10) по запросу определить, какие животные степной зоны содержатся в зоопарке;
 - 11) по запросу определить, сколько надо суммарно затратить на пропитание всех животных в течение месяца (30 дней);
 - 12) вывести на экран отсортированный список животных по их названиям.
- Программа должна обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Задание к теме 45

Составить программу, которая содержит динамическую информацию о кадровом составе направлений обучения в виде односвязного списка. Сведения о лаборатории включают:

- название направления;
- общее количество преподавателей
- количество остепенённых преподавателей;
- количество нестепенных преподавателей;
- количество внешних преподавателей;
- количество профессоров и докторов наук.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех направлений;
- 3) добавление направления в начало списка;
- 4) добавление направления в конец списка;
- 5) добавление направления после определенного направления;
- 6) добавление направления перед определенным направлением;
- 7) удаление направления из списка;
- 8) по запросу выдаются сведения о направлении с наименьшим количеством остепенённых преподавателей;
- 9) по запросу выдаются сведения о направлении и ее показателях в процентом соотношении от общего количества работающих сотрудников.

10) по запросу создается второй новый односвязный список, содержащий только направления, у которых читают профессоры.

11) сохранение данных списка в бинарный файл.

12) извлечение данных из бинарного файла.

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

Задание к теме 46

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это справочник товара: код товара и название товара. Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

1) построение бинарного дерева минимальной высоты;

2) создание пустого дерева,

3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или

корень

левое поддерево

правое поддерево).

4) вывод данных путем обхода дерева в порядке (корень-левое-правое).

5) вставка нового узла в дерево.

6) удаление выбранного узла пользователем из дерева.

7) удаление всех узлов дерева,

8) изменение выбранного узла пользователем из дерева.

9) добавление левого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).

10) исключение поддерева из его правой ветви заданного узла исходного дерева.

11) выполнить поиск всех товаров, название которых начинаются на одну общую букву.

12) вывести количество вершин дерева, значение которых равно числу, введенному пользователем.

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

Задание к теме 47

Составить программу, которая содержит сведения о успеваемости студентов в виде двусвязного списка.

Запись состоит из полей:

- фамилия;
- имя студента;
- группа;
- 5 оценок по предметам.

Программа должна обеспечивать:

- вывод меню для работы со списком;
- добавление нового студента в начало списка;
- добавление нового студента в конец списка;
- добавление нового студента в позицию, отсортированную по имени в алфавитном порядке;
- добавление нового студента после определенного студента;
- добавление нового студента перед определенным студентом;
- удаление данных о студенте;
- сохранение данных списка студентов в текстовый файл;
- извлечение данных из текстового файла и помещение новых студентов в список;
- по запросу вывести среднюю успеваемость студентам сгруппированных по группам;
- по запросу определить наилучшего и наихудшего студента;
- вывести на экран отсортированный список студентов по среднему баллу успеваемости.

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

Задание к теме 48

Составить программу, которая содержит динамическую информацию о владельцах животных в виде односвязного списка. Сведения о владельцах включают:

- ФИО;
- адрес;
- животное;
- имя питомца;
- возраст питомца.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) вывод всех владельцев;
- 3) добавление владельца в начало списка;
- 4) добавление владельца в конец списка;
- 5) добавление владельца после определенного владельца;
- 6) добавление владельца перед определенным владельцем;
- 7) удаление владельца из списка;
- 8) по запросу выдаются сведения о владельцах сгруппированных по классам животных (кошки, собаки и т.д.);

9) по запросу выдаются сведения о владельца, отсортированных по фамилиям.

10) по запросу создается второй новый односвязный список, содержащий только владельцев кошек.

11) сохранение данных списка в бинарный файл.

12) извлечение данных из бинарного файла.

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

Задание к теме 49

Составить программу для построения бинарного дерева. Сведения, которые необходимо хранить в дереве – это продажа товаров (название товара, дата продажи, стоимость). Данные дерева сформировать по числовой последовательности. Программа должна выполнять:

1) построение бинарного дерева поиска;

2) создание пустого дерева,

3) вывод данных в виде дерева (использовать для разделения элементов

левое поддерево

корень

правое поддерево

табуляцию в виде или
корень

левое поддерево

правое поддерево).

4) вывод данных путем обхода дерева в порядке (корень-левое-правое).

5) вставка нового узла в дерево.

6) удаление выбранного узла пользователем из дерева.

7) удаление всех узлов дерева,

8) изменение выбранного узла пользователем из дерева.

9) определение числа узлов дерева.

10) вывод данных дерева в отсортированном виде.

11) добавление правого поддерева в дерево. (Для выполнения этой операции должны быть заданы: включаемое поддерево и узел исходного дерева, к которому поддерево подключается в качестве ветви. Данные нового поддерева задаются пользователем).

12) исключение поддерева из левой ветви заданного узла исходного дерева.

13) вывод дерева в текстовый файл.

14) извлечение данных из текстового файла и построение бинарного дерева.

15) выполнить поиск всех товаров с максимальной стоимостью.

16) вывести количество листьев дерева (листом дерева называется его вершина, не имеющая дочерних вершин).

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

Задание к теме 50

Составить программу, которая содержит ведения о сотрудниках организации в виде двусвязного списка.

Запись состоит из полей:

- ФИО;
- возраст;
- должность
- зарплата в месяц.

Программа должна обеспечивать:

- 1) вывод меню для работы со списком;
- 2) добавление нового сотрудника в начало списка;
- 3) добавление нового сотрудника в конец списка;
- 4) добавление нового сотрудника в позицию, отсортированную по имени в алфавитном порядке;
- 5) добавление нового сотрудника после определенного сотрудника;
- 6) добавление нового сотрудника перед определенным сотрудником;
- 7) удаление данных о сотруднике;
- 8) сохранение данных списка сотрудников в текстовый файл;
- 9) извлечение данных из текстового файла и помещение новых сотрудников в список;
- 10) по запросу определить, какие сотрудники программисты;
- 11) по запросу определить, сколько надо суммарно затратить на зарплату сотрудникам организации в течение года;
- 12) вывести на экран отсортированный список сотрудников по их возрасту.

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

4 ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

4.1 Стандарты, используемые при оформлении пояснительной записки

Пояснительная записка оформляется в соответствии с общеприменяемыми требованиями, в основу которых положены следующие стандарты:

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ Р 6.30-97. Унифицированная система организационно-распорядительной документации. Требования к оформлению документов.

ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

ГОСТ Р 7.0.5-2008 Библиографическая ссылка. Общие требования и правила составления.

ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.

ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы

ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения».

Основные положения курсовой работы необходимо иллюстрировать таблицами, рисунками, расчетами, схемами.

Объем пояснительной записки курсовой проекта должен быть не менее 20 – 25 страниц печатного текста (без приложений).

Пояснительная записка переплетается или сшивается в папку, все страницы, включая титульный лист и приложения, нумеруются порядковыми номерами.

Пояснительная записка должна быть выполнена в соответствии с требованиями ГОСТ 2.105 - 95 «Общие требования к текстовым документам», Правилами оформления текста курсовой работы (проекта) и дипломной работы (проекта), РГУПС, 2020 г.

Пример оформления первого листа пояснительной записки проекта см. приложение Б. Каждый раздел пояснительной записки рекомендуется начинать с новой страницы.

На титульном листе пояснительной записки студентом ставится подпись.

Пояснительная записка

Пояснительная записка выполняется с помощью табличного процессора MS Office и распечатывается на лазерном принтере. Оформление текста должно соответствовать ЕСКД и основному стандарту ГОСТ 7.32–2017, регламентирующий оформление любого текстового документа, в том числе и составление курсовой работы.

Текст располагается на одной стороне белого листа формата А4 (210×297 мм). На каждом листе должна быть выполнена рамка по ГОСТ 2.104 (размер рамки 185×287 мм, толщина линии рамки 0,8-1,2 мм). Рамка имеет отступ от левого края листа 20 мм, от верхнего края листа - 5 мм. Размер и толщина линии рамки на всех листах должны быть одинаковыми. Лист аннотации содержит основную надпись по форме 2, остальные листы, за исключением титульного, содержат основную надпись по сокращенной (неполной) форме 2а.

Текст на странице со всех сторон должен иметь отступ от рамки и от основной надписи, равный 4-5 мм.

Пояснительная записка представляется к защите переплетённой или сброшюрованной.

Пример оформления титульного листа приведен в приложении А на рисунке А.1.

Пример оформления листа задания приведен в приложении Б на рисунке Б.1.

Пример оформления листа содержания приведен в приложении В на рисунке В.1.

Основной текст

Текст выполняется шрифтом Times New Roman, размер шрифта 14 пунктов, междустрочный интервал – полторный (за исключением листинга программного кода). Выравнивание основного текста - «по ширине». Абзацы основного текста выполняются с отступом первой строки, равным 1,25 см (с «красной» строкой).

Не допускаются выделения текста при помощи полужирного и (или) курсивного начертания букв, а также подчеркивания. Переносы слов в документе должны быть разрешены.

Текст программного кода

Текст программного кода выносится в отдельное приложение и форматируется шрифтом Consolas, размер шрифта 11 пунктов, междустрочный интервал – одинарный. Выравнивание основного текста - «по левому краю». Абзацы текста выполняются с отступом первой строки, равным 1,25 см (с «красной» строкой).

Пример оформления программного кода приведен на рис. 3.1.

Приложение А

(обязательное)

Листинг кода

```
Config.php
<?php
//Название сайта
function SiteURL()
{
$protocol = (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] !==
'off' || $_SERVER['SERVER_PORT'] == 443) ? 'https://' : 'http://';
$domainName = $_SERVER['HTTP_HOST'].'/'.'//.'/'Shop/';
return $protocol.$domainName;
}
define('SITE_URL', SiteURL());
define('GOODS_IMAGE', 'Images/Goods/');
//Исходный запрос
function RequestURL()
{
if (isset($_GET['p']))
```

Рисунок 3.1 – Пример оформления программного кода

Заголовки

Заголовки разделов, подразделов, содержание, введение, заключение, приложения оформляются жирным шрифтом.

Заголовки разделов, подразделов, пунктов и подпунктов также выполняются с «красной» строкой и выравниваются по левому краю. Заголовки не должны содержать переносов слов. В конце заголовка не допускаются никакие знаки препинания.

Заголовки разделов и подразделов, пунктов и подпунктов должны быть по возможности краткими и раскрывать содержание изложенного материала. Разделы и подразделы должны иметь порядковые номера, обозначенные арабскими цифрами. После каждого номера должна стоять точка, за исключением последнего номера, после которого точка не ставится (рис. 3.2).

Разделы «Обозначения и сокращения», «Введение» и «Заключение» не нумеруются (рис. 3.4–3.6).

Каждый раздел следует начинать с новой страницы.

Расстояние между заголовком раздела, подраздела, пункта, подпункта и основным текстом визуально должно составлять одну пустую строку.

1 Анализ предметной области и формирование требований к АС

1.1 Магазин по продаже цветов ООО «Флора»

Рисунок 3.2 – Пример заголовков 1 и 2 уровня

Отступ красной строки 1,25 по линейке (рис. 3.3).

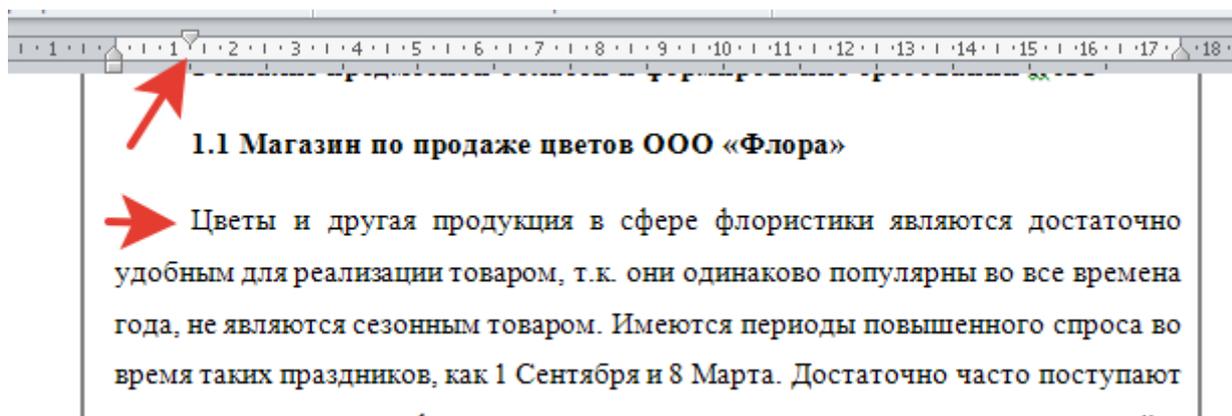


Рисунок 3.3 – Пример красной строки

Содержание	
Обозначения и сокращения	6
Введение.....	7
1 Исследование информационных систем диагностики и анализа технического состояния подвижного состава. Постановка проблемы	10

Рисунок 3.4 – Пример содержания

Обозначения и сокращения
АИС – автоматизированная информационная система
ИАС – информационно-аналитическая система
АРМ – автоматизированные рабочие места
СУБД – системы управления базами данных
БД – база данных
ADO.NET – универсальная технология доступа к данным
ПС – подвижной состав
ЖДС – железнодорожная станция
РЖД – российские железные дороги
ТО – техническое обслуживание подвижного состава
ТР – текущий деповский ремонт подвижного состава
КР – капитальный ремонт железнодорожной техники

Рисунок 3.5 – Пример обозначений и сокращений

Введение

Процессы автоматизации на современном этапе развития общества проникли во все сферы деятельности человека. Так, одним из приоритетных направлений развития железнодорожного транспорта Российской Федерации является широкое внедрение современных информационных технологий на всех уровнях.

Рисунок 3.6 – Пример введения

Списки

Перечисления оформляются так же, как основной текст. Пример оформления перечислений:

- пример элемента перечисления первого уровня;
- а) пример элемента перечисления второго уровня, который располагается на двух строчках;
 - 1) пример элемента перечисления третьего уровня, который располагается на двух строчках;
 - 2) пример элемента перечисления третьего уровня;
- б) пример элемента перечисления второго уровня;
- пример элемента перечисления первого уровня, который располагается на двух строчках.

Пример оформления маркированного списка показан на рис. 3.7.

Электронную коммерцию в настоящее время принято разделять на ряд направлений (рисунок 2.1), основными из которых считаются:

- «бизнес-бизнес» (Business-to-Business – B2B);
- «бизнес-потребитель» (Business-to-Customer или Business-to-Client – B2C);

Лист
18

- «потребитель-бизнес» (Consumer-to-Business – C2B);
- «потребитель-потребитель» (Consumer-to-Consumer – C2C).

Рисунок 3.7 – Маркированный список начинается с маленькой буквы

Если список состоит из нескольких больших предложений, то первое слово начинается с заглавной буквы и в конце каждого элемента списка ставится точка (рис. 3.8).

К системам B2C относятся:

- Web-витрины (Front Office) торговых компаний для привлечения возможных покупателей к продуктам данных компаний.
- Internet-магазины, которые занимаются только продажей товаров и содержат необходимую инфраструктуру (Back Office) для производства продаж и управление электронной торговлей через Internet.
- Торговые Internet-компании, в которых система электронных продаж (Back Office) полностью интегрирована со всеми торговыми бизнес-процессами.

Рисунок 3.8 – Маркированный список с заглавной буквы

Пример выравнивания маркированного списка показан на рис. 3.9.

При формировании площадок B2B необходимо учитывать ряд важных аспектов:

- доступность для новых участников;
- поддержка признанных стандартов разработки (EDI, Web-формы, XML-приложения);
- масштабируемость используемых платформ;
- возможность управления информацией и применения аналитических

Рисунок 3.9 – Пример маркированного списка

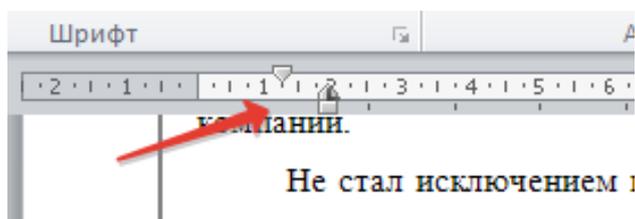


Рисунок 3.10 – Пример линейки для маркированного списка

Пример оформления нумерованного списка показан на рис. 3.11.

Основной успешный сценарий:

1 Покупатель с поисковых систем, через рекламу, закладки или по внешним ссылкам попадает на главную страницу (или другую страницу), откуда он может перемещаться в произвольный раздел магазина.

2 Выбрав в верхнем меню раздел «каталог», посетитель перемещается на страницу, содержащую информацию о всех имеющихся категориях геодезического оборудования, представленного в магазине.

3 Выбрав нужную категорию товаров, посетитель просматривает список товаров.

Рисунок 3.11 – Пример нумерованного списка

Рисунки

Документ может содержать таблицы, рисунки и формулы. На каждую таблицу, рисунок или формулу в документе непосредственно перед таблицей, рисунком или формулой должна быть ссылка, например, «рисунок 3.1», «(таблица 3.1)», «по формуле 3.2».

Рисунок размещается по центру страницы. Под рисунком размещается подпись, состоящая из слова «Рисунок» и порядкового номера. Если рисунок имеет название, оно размещается за номером рисунка через тире. Пример оформления рисунка приведен на рис. 3.12.



Рисунок 3.12 – Пример оформления рисунка

Структура сайта представлена на рисунке 5.1.



Рисунок 5.1 – Структура сайта

Рисунок 3.13 – Пример рисунка

Знаки препинания в конце названия рисунка недопустимы. Рисунок и подпись рисунка должны располагаться на одной странице. Расстояние между текстом и рисунком, между рисунком и подписью рисунка, а также между подписью рисунка и последующим текстом визуально должно составлять одну пустую строку.

Таблицы

Таблица выравнивается по левому краю. Перед таблицей должна находиться подпись, состоящая из слова «Таблица» и порядкового номера таблицы. Если таблица имеет название, оно размещается за номером рисунка через тире.

Знаки препинания в конце названия таблицы недопустимы. Расстояние между текстом и подписью таблицы, а также между таблицей и последующим текстом визуально должно составлять одну пустую строку. Расстояние между подписью таблицы и таблицей визуально должно составлять половину пустой строки. Подпись таблицы и начало таблицы должны располагаться на одной странице.

Если таблица не умещается на одном листе, она разрывается на две или более частей. Перед таблицей, которая является продолжением таблицы, начало которой находится на предыдущей странице, размещается подпись «Продолжение таблицы» или «Окончание таблицы» с указанием номера.

Рамки таблицы выполняются толщиной 0,75 пункта. Между рамкой таблицы и текстом внутри таблицы должно быть расстояние не менее 2 мм. Числовые данные внутри таблиц выравниваются по правому краю так, чтобы одноименные разряды чисел находились на одной вертикали. Текст внутри таблиц выравнивается по левому краю. Если текст имеет вид предложения, размещенного на нескольких строчках, то он должен иметь отступ «красной» строки, равный 1–1,5 см.

В заголовках граф таблицы не допускаются переносы.

Таблица нумеруется в зависимости от раздела, в котором она стоит.

Название таблицы указывается сверху и выравнивается по левому краю (рис. 3.14).

Используемые методы и классы описаны в таблице 5.1.

Таблица 5.1 – Используемые методы и классы

Наименование	Методы	Классы, с которыми взаимодействует	Краткое описание
1	2	3	4
Ядро сайта	LoadGoods LoadProducers LoadCountries SaveGood SaveProducer SaveCountry SaveOrder ShowPage FilterGoods AddToCart MakePurchase	PHP Data Objects Страница списка товаров Страница товара Страница заказа Страница входа Страница стран Страница производителей Страница товаров на складе Страница заказов	Выполняет загрузку основных библиотек для функционирования сайта, выполняет работу по взаимодействию с базой данных, формирует основной макет сайта, передает управление отдельным страницам для отрисовки их содержимого

Рисунок 3.14 – Пример таблицы

При продолжении таблицы название ее не пишется. Вместо названий столбцов переносятся их номера (рис. 3.15).

Продолжение Таблицы 5.1

1	2	3	4
MySQL	ExecuteQuery	PHP Data Objects	Класс СУБД MySQL
PHP Data Objects	ExecuteQuery	Ядро сайта	Расширение для PHP, предоставляющее простой и универсальный интерфейс для доступа к различным базам данных.
Страница списка товаров	ShowGoods	Ядро сайта	Отрисовывает страницу, содержащую список товаров
Страница товара	ShowGood	Ядро сайта	Отрисовывает страницу, содержащую информацию о товаре

Рисунок 3.15 – Пример продолжения таблицы

Фрагменты программного кода

В текст пояснительной записки могут быть включены небольшие фрагменты кода, если они помогают раскрыть содержание. При этом фрагменты кода должны оформляться как рисунок, и текст документа должен иметь ссылку на фрагмент кода, как на рисунок. Рекомендуемый шрифт для выполнения фрагмента кода - Courier New, размер шрифта 10 пунктов, междустрочное расстояние одинарное, начертание обычное. Каждая строка кода должна иметь отступ 1,25 см от левого края. Пример выполнения фрагмента кода приведен на рис. 3.16.

```
namespace WindowsDataBases
{
    public partial class Form2 : Form
    {
        SqlDataAdapter dataAdapterFacultet = new SqlDataAdapter();

        DataSet dsFacultet = new DataSet();

        BindingSource bsFacultet = new BindingSource();

        SqlCommand myCommand;
```

Рисунок 3.16 – Пример оформления фрагмента кода

Алгоритмы программы

Вместо фрагментов кода предпочтительнее вставлять в текст документа алгоритмы, выполненные на виде алгоритмической записи или в виде блок-схемы.

Алгоритмы, выполненные в виде алгоритмической записи, оформляются так же, как и фрагменты кода. Блок-схемы оформляются в соответствии с ГОСТ 19.701 90. Непосредственно в тексте допускается размещать блок-схемы, целиком уместающиеся на одной странице. При этом блок-схема помечается, как рисунок, и текст документа ссылается на нее, как на рисунок. Объемные блок-схемы следует размещать в приложениях, помеченных как обязательные, если они необходимы для правильного понимания текста. Пример блок-схемы приведен на рис. 3.17.

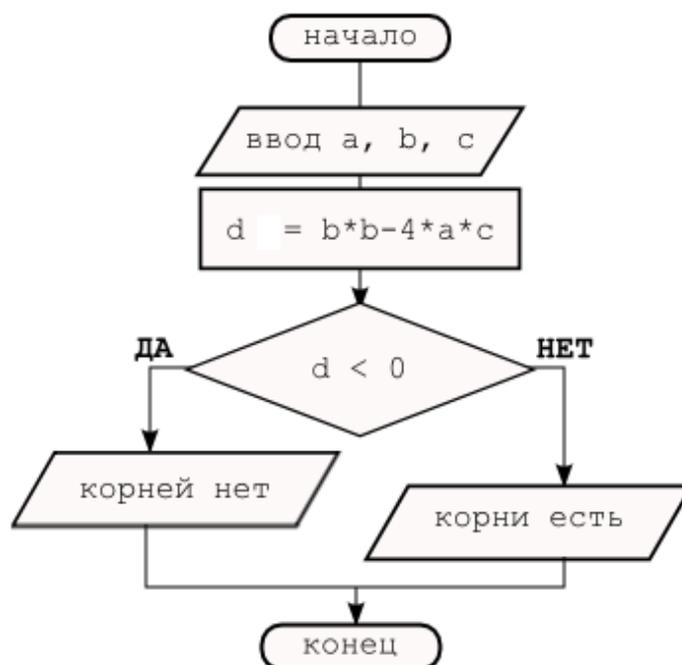


Рисунок 3.17 – Пример оформления блок-схемы

Ссылки на литературные источники

В основном тексте нецелесообразно использовать длинные цитаты. Запрещается воспроизведение фрагментов текста, фактов, данных публикаций тех или иных авторов без указания заимствованных источников. На материалы, взятые из литературы и других источников (утверждения, формулы, цитаты и т.п.) должны быть даны ссылки с указанием номера источника по списку использованной литературы. Номер ссылки проставляется арабскими цифрами в квадратных скобках.

Перспектива внедрения данного технического обслуживания и ремонта подвижного состава в значительной мере будет зависеть от возможности получения исчерпывающей и достоверной информации о состоянии оборудования и деталей каждого подвижного состава. Источником данной информации должны быть результаты анализа как статистической информации о неисправностях оборудования, так и результаты его диагностирования [1].

В настоящее время на железнодорожных станциях и ремонтных депо система сбора и обработки информации о техническом состоянии подвижного состава не удовлетворяет поставленной цели. Решение данной задачи видится в применении новых информационных технологий, обеспечивающих выполнение поставленной задачи системы учета и обработки информации [2, 3].

Рисунок 3.18 – Пример оформления ссылок

Список использованных источников

Список использованных источников должен содержать только те источники, которые непосредственно использованы студентом и на которую имеются ссылки в тексте. Список составляется в порядке появления ссылок.

Пример оформления списка используемых источников приведен на рис. 3.19.

Список используемых источников

- 1 **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5/ Р. Никсон. – СПб.: Питер, 2015. – 688 с.
- 2 **Вайсфельд, М.** Объектно-ориентированное мышление/ М. Вайсфельд. – СПб.: Питер, 2014. – 304 с.
- 3 **Симдянов, И.** PHP 7/ Симдянов, И., Котеров, Д. – СПб.: БХВ-Петербург, 2016. – 1073 с.
- 4 **Колисниченко, Д.** PHP и MySQL. Разработка Web-приложений/ Д. Колисниченко. – СПб.: БХВ-Петербург, 2015. – 593 с.
- 5 **Маклафлин, Б.** PHP и MySQL. Исчерпывающее руководство/ Б. Маклафлин. – СПб.: Питер, 2014. – 544 с.
- 6 **Петкович, Д.** SQL для начинающих/ Д. Петкович. – СПб.: Питер, 2013. – 816 с.
- 7 **Прохоренок, Н.** HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера/ Прохоренок, Н., Дронов, В. Д.; - СПб.: БХВ-Петербург, 2015. – 766 с.

Рисунок 3.19 – Пример списка используемых источников

- 10 Рекомендации по созданию web-сайта для интернет-магазинов. Web-студия «Tell different», создание и продвижение сайтов. [Электронный документ] <http://prach.poltava.ua/posts/web-internet/153.html>. Проверено 26.02.2018.
- 11 Частная коллекция качественных материалов для тех, кто делает сайты. [Электронный документ] <http://ruseller.com/lessons.php?id=2085&rub=28>. Проверено 26.02.2018.

Рисунок 3.20 – Пример оформления электронных документов

Примеры библиографических записей использованных источников, согласно Стандарту.

1 Книги

1.1 Однотомные издания

1.1.1 Один автор

Орлов, А.И. Эконометрика / А.И. Орлов. – М.: Изд-во «Экзамен», 2002. – 576 с.

1.1.2 Два автора

Иванов, Н.И. Основы виброакустики / Н.И. Иванов, А.С. Никифоров. – СПб.: Политехника, 2000. – 482 с.

1.1.3 Три автора

Фигурнов, Е.П. Релейная защита сетей тягового электроснабжения переменного тока / Е.П. Фигурнов, Ю.И. Жарков, Т.Е. Петрова. – М.: Маршрут, 2006. – 272 с.

1.1.4 Четыре и более авторов

Моделирование электромеханической системы электровоза с асинхронным тяговым приводом / Ю.А. Бахвалов [и др.]. – М.: Транспорт, 2001. – 286 с.

Неразрушающий контроль и диагностика: справочник / под ред. чл.-кор. РАН проф. В.В. Клюева. – М.: Машиностроение, 2005. – 656 с.

1.2 Многотомные издания в целом

Справочник по триботехнике : в 3 т. / под ред. М. Хебды, А.В. Чичинадзе. – М.: Машиностроение, 1989.

1.3 Отдельный том многотомного издания

Справочник по триботехнике : в 3 т. Т. 1 : Теоретические основы / под ред. М. Хебды, А.В. Чичинадзе. – М.: Машиностроение, 1989. – 397 с.

2 Законодательные материалы

Российская Федерация. Конституция (1993). Конституция Российской Федерации : офиц. текст. – М.: Маркетинг, 2001. – 39 с.

3 Сборники научных трудов, материалы конгрессов, конференций, совещаний

Вопросы организации грузовой и коммерческой работы : Тр. ин-тов инж. ж.-д. трансп. / Ред. А.А. Смехов ; Моск. ин-т ж.-д. трансп. – М., 1978. – Вып. 588. – 123 с.

Механика и трибология транспортных систем – 2003, сентябрь 2003 г. : Сб. докл. междунар. конгр. В 2 т. / РАН; МПС РФ, РГУПС. – Ростов н/Д : РГУПС, 2003.

Сборник тезисов докладов 65-й студенческой научно-практической конференции / РГУПС. – Ростов н/Д, 2006. – 207 с.

4 Статьи из сборников научных трудов и журналов

Мильков, Ю.А. Поляризационный оптический переключатель на волоконном световоде / Ю.А. Мильков, Е.З. Савин // Оптика кристаллов : Сб. науч. тр.; под ред. В.И. Строганова. – Хабаровск: Изд-во ДВГУПС, 2004. – С. 18-20.

Мугинштейн, Л.А. О выборе типа тягового электропривода электроподвижного состава / Л.А. Мугинштейн, Л.А. Кучумов, О.Н. Назаров // Железнодорожный транспорт. – 2005. - № 5. – С. 42-48.

Марков, Д.П. Приработка и повреждаемость элементов колесно-рельсовой трибосистемы / Д.П. Марков // Вестник РГУПС. - Ростов н/Д, - 2004. - № 3. – С. 25-33.

Вуколов, Л.А. Сравнительные характеристики тормозных колодок различных поставщиков / Л.А. Вуколов, В.А. Жаров // Вестник ВНИИЖТ. - 2005. - № 2. – С. 16-20.

Роль диффузионных и сегрегационных процессов в контактно-усталостном разрушении рабочей поверхности железнодорожного колеса в металлополимерном сопряжении / В.И. Колесников [и др.] // Трение и смазка в машинах и механизмах. – 2006. - № 8. – С. 22-32.

5 Доклады, тезисы докладов на семинарах и конференциях

Гуда, А.Н. Сбор и анализ эмпирических данных в телекоммуникационных сетях на транспорте / А.Н. Гуда, М.А. Бутакова // IV Междунар. науч.-практич. конф. «Телекоммуникационные технологии на транспорте России»: сб. докл. – Ростов н/Д, 2005. – С. 200-206.

Распознавание процессов в зоне трения по спектру акустической эмиссии / А.П. Брагинский [и др.] // Тез. докл. междунар. конф. «Трение, износ и смазочные материалы». – Ташкент, 1985. – Т. 5. – С. 83-84.

6 Стандарты, правила, нормы, методики, инструкции

ГОСТ Р 51771 – 2001. Аппаратура радиоэлектронная бытовая. Входные и выходные параметры и типы соединений. Технические требования. – Введ. 2002 – 01 – 01. – М.: Госстандарт России : Изд-во стандартов, 2001. – 27 с.

ГОСТ 10749.1-80. Спирт этиловый технический. Методы анализа: [Сборник]. - М.: Изд-во стандартов, 1981. – 47 с.

Правила устройства и безопасной эксплуатации грузоподъемных кранов.: утв. Госгортехнадзором РФ 30.12.92. – СПб.: Деан, 2000. – 224 с.

Нормы оснащения объектов и подвижного состава федерального железнодорожного транспорта первичными средствами пожаротушения : Утв. : 31.03.2000 г. № Г – 822у. – М., 2000. – 103 с.

Методика определения норм удельного расхода и лимита потребления электроэнергии для объектов грузового хозяйства железных дорог (ЦММ – 32) : Утв. : 27.08.2001 / МПС РФ. Департамент грузовой и коммерческой работы. – М., 2001. – 48 с.

Инструкция по формированию, ремонту и содержанию колесных пар тягового подвижного состава железных дорог колеи 1520 мм (ЦТ329). – М.: Техинформ, 2000. – 136 с.

7 Патентные документы

Пат. 2224147 РФ, МПК F16C 33/04. Способ получения антифрикционных высокопрочных изделий / В.А. Лапицкий, В.И. Колесников, В.Б. Воробьев, А.П. Сычев. - № 2002127862/04 ; заявл. 17.10.02 ; опубл. 20.02.04, Бюл. № 5.

Пат. РФ на полезную модель № 42484. Устройство для электроснабжения нетяговых потребителей на электрофицированных участках железных дорог переменного тока / А.С. Бочев, Т.Э. Финоченко. Бюл. «Изобретения. Полезные модели» № 34. – 2004.

А. с. 1093959 СССР, МКИ³ G 01 N 27/83. Устройство для магнитошумовой структуроскопии / А.В. Попов, Б.И. Морозов, В.А. Ткаченко (СССР). - № 3411327/25-28 ; заявл. 25.03.82 ; опубл. 23.05.84, Бюл. № 19.

8 Промышленные каталоги

Машина специальная листогибочная ИО 217М : листок-каталог : разработчик и изготовитель Кемер. з-д электромонтаж. изделий.–М., 2002.– 3 л.

9 Депонированные труды

Панов, В.Ф. Модели частиц в сильной гравитации / В.Ф. Панов ; Ред. журн. «Изв. вузов. Физика». – Томск, 1982. – 7 с. – Деп. в ВИНТИ 27.05.82, № 2641.

10 Неопубликованные документы

10.1 Отчет о научно-исследовательской работе

Повышение эксплуатационных характеристик строительных машин путем их модернизации в ходе ремонтных работ: отчет о НИР (заключ.) / Рост. ин-т инж. ж.д. трансп. (РИИЖТ); рук. Шаповалов В.В. – Ростов н/Д, 1986. – 65 с. - № ГР 01870005362. -Инв. № 02870008770.

10.2 Диссертация

Петров, П.Ю. Быстродействующая система управления тяговым электроприводом для улучшения сцепных свойств электроподвижного состава с асинхронными тяговыми двигателями : дис. ... канд. техн. наук : 05.09.03 / П.Ю. Петров ; Моск. гос. ун-т путей сообщения (МИИТ). – М., 1998. – 197 с.

Приложения

В качестве приложений к курсовой работе может быть иллюстрированный фактический материал, служащий для подтверждения тех или иных положений автора и занимающий определенный объем: листинг программы, большие таблицы и др.

Каждое приложение начинается с нового листа.

Приложения нумеруются буквами русского алфавита в порядке появления ссылок на них в основном тексте документа. Заголовок приложения состоит из трех строк, составляющих один абзац, выравнивание текста по центру. Первая строка содержит слово «Приложение» и прописную букву, обозначающую номер приложения, например, «Приложение А». Вторая строка содержит заключенное в круглые скобки слово «обязательное», «справочное» или «рекомендуемое». Третья строка содержит название приложения с прописной первой буквы.

Примеры приложений приведены на рис. 3.21–3.23.

Приложения

Рисунок 3.21 – Приложения

Приложение А

(обязательное)

Листинг кода

```
Config.php
<?php
//Название сайта
function SiteURL()
{
$protocol = (!empty($_SERVER['HTTPS']) && $_SERVER['HTTPS'] !==
'off' || $_SERVER['SERVER_PORT'] == 443) ? 'https://' : 'http://';
$domainName = $_SERVER['HTTP_HOST'].'/'.'//.'/'Shop/';
return $protocol.$domainName;
}
define('SITE_URL', SiteURL());
define('GOODS_IMAGE', 'Images/Goods/');
//Исходный запрос
function RequestURL()
{
if (isset($_GET['p']))
return SITE_URL.$GET['p'];
}
```

Рисунок 3.22 – Пример приложения А

Приложение Б

(обязательное)

Антиплагиат к бакалаврской работе

Отчет о проверке на заимствования №1

Автор: Бутаква Мария Александровна butakova@rups.ru / ID: 39
Проверщик: Бутаква Мария Александровна butakova@rups.ru / ID: 39
Организация: Ростовский Государственный Университет Путей Сообщения
Отчет предоставлен сервисом «Антиплагиат» - <http://rups.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 781
Начало загрузки: 14.06.2018 21:36:33
Длительность загрузки: 00:00:10
Имя исходного файла: Бочалгова
нормоконтроль
Размер текста: 3516 кБ
Тип документа: Выпускная
квалификационная работа
Символов в тексте: 153344
Слов в тексте: 18464
Число предложений: 912

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 14.06.2018 21:36:44
Длительность проверки: 00:00:11
Комментарии: не указано
Модули поиска: Кольцо вузов, Модуль поиска общепотребительных выражений,
Модуль поиска "РГУПС", Модуль поиска рефразирований Интернет, Модуль
поиска Интернет, Цитирование, Коллекция РГБ, Сводная коллекция ЭБС

заимствования	цитирования	ОРИГИНАЛЬНОСТЬ
21,96%	0,3%	77,74%



Рисунок 3.23 – Пример приложения Б

5 КОМПЬЮТЕРНАЯ ПРЕЗЕНТАЦИЯ

Компьютерная презентация выполняется в Microsoft PowerPoint. Она состоит из последовательности слайдов. Рекомендуемое количество слайдов 8–10.

Презентация сопровождает выступление студента и поэтому строится в соответствии с его докладом. Выступление необходимо продумать таким образом, чтобы сформировать 10–15 фрагментов длительностью 30–40 секунд (ориентировочно), на которых будет обращено внимание комиссии. Каждой такой фрагмент выступления должен быть связан соответствующим слайдом презентации.

Каждый слайд должен иметь заголовок из одной строки.

На слайдах следует размещать либо пункты, которые подчеркивают главные моменты фрагмента выступления, либо графический материал в виде рисунков, диаграмм, схем, графиков, формул и т. п. Недопустимо размещать на слайдах текст выступления. Следует помнить о том, что если все слайды презентации содержат только текстовый материал, дублирующий выступление, надобность в презентации отпадает и эффект, который от нее можно было бы получить, утрачивается.

Если слайд содержит пункты, они не должны дословно цитировать выступление. Вместо этого каждый пункт должен являться кратким содержанием предложения, которое выступающий проговаривает полностью. Как правило, пункт слайда содержит не более одной строки, а в идеале – одно слово, являющееся ключевым.

Если слайд содержит графический материал, количество отдельных элементов этого материала должно совпадать с тем, что намеревается показать выступающий. Если докладчик рассказывает о трех составляющих системы, которая отображена на слайде, рисунок при этом должен содержать три элемента, и количество подписей к ним также должно быть равно трем.

Следует помнить о том, что во время выступления докладчик стоит спиной к экрану (и лицом к комиссии). Если во время выступления необходимо показать на какой-то элемент графического материала, следует показать указкой в его сторону, не сходя при этом с места. Чтобы приемная комиссия могла понять, на какой элемент указывает докладчик, этот элемент при необходимости должен быть каким-либо образом выделен. При необходимости выделения нескольких элементов изображения следует подготовить несколько подряд идущих слайдов.

При размещении текста и иллюстративного материала следует помнить о том, что изображение на экране монитора и на экране проектора могут сильно различаться. Поэтому следует избегать мелких надписей, затрудняющих их прочтение, а также сочетания цвета плана и фона, имеющих один цветовой тон. Необходимо обеспечить достаточный контраст так, чтобы все надписи, а также мелкие элементы рисунков, графиков, диаграмм и т.п. были хорошо видны (различимы).

Не рекомендуется использовать для написания текста экзотические шрифты – их может не оказаться на том компьютере, с помощью которого презентация будет демонстрироваться.

При использовании в качестве иллюстративного материала скриншотов экранных форм необходимо помнить о том, что на формах используются мелкие шрифты. Поэтому полученные рисунки при размещении на слайдах лучше увеличить до масштаба 120–150 %. А для этого, вероятно, реальные размеры форм во время получения скриншота нужно уменьшить.

На слайдах недопустимо размещать рекламные элементы изображения, такие, как логотипы.

Анимация слайдов должна быть отключена.

Первый слайд содержит тему курсовой работы, фамилию руководителя и фамилию студента.

Следующий слайд раскрывает предметную область и подводит повествование к цели курсового проектирования. Для представления целей и задач курсовой работы отводится один слайд. Далее следует разместить слайд, на котором описываются средства разработки. Оставшиеся слайды посвящаются разработке проекта. В конце могут быть приведены слайды, посвященные результатам проектирования и перспективам проекта. Не следует заключать последовательность слайдом

«Спасибо за внимание» или подобным.

Во время выступления следует избегать ситуаций, когда слайд отображается на экране время, недостаточное для его полного прочтения, или отображается дольше необходимого. Время отображения слайда должно в точности соответствовать времени, необходимому для рассказа о его содержимом. Недопустимо рассказывать во время показа слайда о предметах, которые не нашли отображение на слайде, равно как и рассказывать о содержимом слишком долго.

6 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ЗАЩИТЕ КУРСОВОЙ РАБОТЫ

Курсовая работа, не соответствующая вышеуказанным требованиям, к защите не допускается.

Выполненная курсовая работа (разработанное приложение и пояснительная записка) перед сдачей руководителю должна быть тщательно выверена. Преподаватель проверяет курсовую работу, подписывает титульный лист.

Подготовленную курсовую работу студент представляет на проверку руководителю не позднее срока, указанного в задании.

Отзыв руководителя включает:

- заключение о соответствии курсовой работы заявленной теме и составленной спецификации;
- оценку полноты разработки программного приложения по выбранной теме и практической значимости курсовой работы;
- оценку качества оформления пояснительной записки курсовой работы;
- оценку курсовой работы в балльной системе.

При несоблюдении требований к профессиональному уровню, содержанию и оформлению курсовой работы руководитель возвращает ее студенту для доработки и устранения недостатков.

Нарушение правил оформления пояснительной записки курсовой работы может быть основанием для возврата ее на доработку, недопуска к защите или снижения оценки.

Курсовая работа защищается перед научным руководителем следующим образом: предварительно ознакомившись с отзывом, студент дает пояснения по существу критических замечаний по работе, отвечает на вопросы, поставленные руководителем, обосновывает свои выводы. При оценке курсовой работы учитывается не только ее выполнение, но и результаты защиты, владение студентом знаниями по данной проблеме. Научный руководитель определяет оценку и заносит ее в зачетную книжку студента, а также в ведомость. Защищенные курсовые работы студентам не возвращаются.

В пределах объема времени, предусмотренного на выполнение курсовой работы по дисциплине, предусматривается защита курсовой работы (8-10 минут), которая состоит из демонстрации разработанной программы, доклада студента и ответов на поставленные вопросы. В докладе в сжатом виде описываются анализ выбранной темы, принципы разработки программы, полученный результат. По окончании доклада студент должен быть готов ответить на ряд вопросов. В процессе беседы со студентом преподаватель выясняет уровень его теоретической подготовки по данной теме, умение автора излагать и обосновывать результаты своей разработки.

Курсовая работа оценивается оценками «отлично» («5»), «хорошо» («4»), «удовлетворительно» («3»), «неудовлетворительно» («2»).

6.1 Критерии оценки курсовой работы

В таблице 1 приведены критерии оценки курсовой работы

Таблица 1 – Критерии оценки курсовой работы

Показатели оценки	Оценка (в баллах)
1. Содержание курсовой работы	
1.1 Обоснование актуальности темы работы	
1.2 Соответствие структуры работы заданию	
1.3 Соответствие содержания разделов пояснительной записки заданию	
1.4 Последовательность, полнота, логика изложения материала	
1.5 Наличие элементов исследовательской деятельности	
1.6 Применение технологий разработки	
1.7 Правильность выполнения программной части работы	
1.8 Наличие выводов по разделам (при необходимости)	
1.9 Правильность построения алгоритма приложения	
1.10 Соблюдение функциональных требований к приложению	
1.11 Отсутствие ошибок функционирования приложения	
1.12 Практическая значимость работы	
Оценка содержания курсовой работы	
2. Качество оформления курсовой работы	
2.1 Соблюдение графика выполнения работы	
2.2 Оформление пояснительной записки в соответствии с Правилами оформления текста курсовой работы (проекта) и дипломной работы (проекта)	
2.3 Соблюдение требований ЕСПД к оформлению пояснительной записки	
2.4 Соблюдение требований к объему частей пояснительной записки работы	
2.5 Соблюдение требований к объему пояснительной записки работы	
2.6 Наличие ссылок на использованные источники	
2.7 Использованные источники, в том числе нормативно-правовые документы: их современность (год издания), соответствие теме работы, количество.	
Оценка качества оформления курсовой работы	

Итоговая оценка за курсовую работу выставляется целым числом в соответствии с правилами математического округления как среднее арифметическое оценок по разделам 1 и 2.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Основная литература

- 1 **Гагарина, Л. Г.** Технология разработки программного обеспечения / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. – Форум, Инфра-М, 2009.
- 2 **Гуда, А. Н.** Алгоритмизация и программирование : учебное пособие / А. Н. Гуда, М. А. Бутакова. – Ростов-на-Дону : ФГБОУ ВО РГУПС, 2003. – 143 с.
- 3 **Дейтел, Х.** Как программировать на C++ / Х. Дейтел, П. Дейтел. – Москва : Вильямс. – 1002 с.
- 4 **Зыков, С. В.** Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. – Москва : Юрайт, 2020. – 155 с. – ISBN 978-5-534-00850-0.
- 5 **Кувшинов, Д. Р.** Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. – Москва : Юрайт, 2020. – 104 с. – ISBN 978-5-534-07559-5.
- 6 **Кудрина, Е. В.** Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. – Москва : Юрайт, 2020. – 322 с. – ISBN 978-5-534-09796-2.
- 7 **Лафоре, Р.** Объектно-ориентированное программирование в C++ / Р. Лафоре. – Питер, 2004. – 922 с.
- 8 **Огнева, М. В.** Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. – Москва : Юрайт, 2020. – 335 с. – ISBN 978-5-534-05123-0.
- 9 **Павловская, Т. А.** C/C++. Программирование на языке высокого уровня : учебник для вузов / Т. А. Павловская. – Москва : Питер, 2004. – 460 с.
- 10 **Павловская, Т. А.** C++. Объектно-ориентированное программирование: практикум : учебное пособие для вузов / Т. А. Павловская, Ю. А. Щупак. – Москва ; Санкт-Петербург : Питер, 2004. – 264 с.
- 11 **Струструп, Б.** Программирование: принципы и практика использования C++ / Б. Струструп. – Москва : Вильямс, 2011. – 1248 с.
- 12 **Струструп, Б.** Язык программирования C++ / Б. Струструп. – Москва : Бином. – 1054 с.
- 13 **Трофимов, В. В.** Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. – Москва : Юрайт, 2020. – 137 с. – ISBN 978-5-534-07834-3.
- 14 **Тузовский, А. Ф.** Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. – Москва : Юрайт, 2020. – 206 с. – ISBN 978-5-534-00849-4.
- 15 **Шилдт, Г.** C++: руководство для начинающих / Г. Шилдт. – 2-е изд. – Москва : Вильямс, 2005. – 672 с.

Дополнительные источники

- 1 C++ в Visual Studio. – Текст : электронный. – URL: <https://docs.microsoft.com/ru-ru/cpp/overview/visual-cpp-in-visual-studio?view=vs-2019>.

2 Базовый курс по C++. – Текст : электронный. – URL: https://www.bestprog.net/ru/sitemap_ru/c/.

3 Руководство по C++. Текст : электронный. – URL: <https://metanit.com/cpp/tutorial/>.

4 Справочник по языку C++. – Текст : электронный. – URL: <https://docs.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=vs-2019>.

5 Справочник стандартной библиотеки C++. – Текст : электронный. – URL: <https://docs.microsoft.com/ru-ru/cpp/standard-library/cpp-standard-library-reference?view=vs-2019>.

Пример оформления титульного листа к пояснительной записке

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

Кафедра *«Вычислительная техника и
автоматизированные системы управления»*

Тема: **«Программирование динамических структур данных»**

Пояснительная записка
к курсовой работе по дисциплине
«Алгоритмизация и программирование»
АИП 01.19 ПЗ

Учебная группа: **АИБ– 2 – 029**

Выполнил студент **Иванов И.И.** _____
(подпись студента)

Руководитель курсовой работы _____ **доц. Игнатьева О.В.**
(подпись руководителя)

Работа допущена к защите _____
(дата)

Работа защищена _____ с оценкой _____
(дата) (оценка) (подпись)

г. Ростов-на-Дону
2021 г.

Пример оформления титульного листа для курсовых работ. Все что выделено желтым цветом нужно изменять.

«**Программирование динамических структур данных**» - это тема общая для всех, которая напечатана в листе задания. У всех она должна звучать одинаково. В исходных данных пишется индивидуальная тема.

Если бы мы брали данный лист задания (Рисунок 1.1), то общая тема в титульном листе звучала бы так: «**Проектирование АРМ для корпоративной информационной системы.**»

" 01 " 11 03 2017

Задание на курсовую работу(проект) № _____

Кафедра: Вычислительная техника и автоматизированные системы управления
Специальность: 09.03.02 Информационные системы и технологии
Форма обучения: ОЧНАЯ(4 ГОДА ОЧНОЕ БАКАЛАВРИАТ)
Дисциплина: Корпоративные информационные системы на транспорте
Вид работы: КУРСОВАЯ РАБОТА
Группа: АИБ-4-031
Студент: Палагута Виктория Сергеевна
Тема курсовой работы(проекта): Проектирование автоматизированного рабочего места для корпоративной информационной системы ←
Предприятие: _____

Исходные данные *1. Спроектировать АРМ для корпоративной системы "Финансирование сделок" с учетом существующих технических средств (при наличии);*
2. Разработать модель проектирования ИИС, проектировать функционал, соответствующий данному типу. ←

Рисунок 1.1 – Пример листа задания

«**Алгоритмизация и программирование**» - это название дисциплины.

Шифр АИП 01.19 ПЗ состоит из следующих частей:

- АИП – это сокращение дисциплины.
- 01 – это номер курсовой работы , общий для всех по решению кафедры,
- 19 – это номер варианта студента по журналу группы,
- ПЗ – пояснительная записка.

Пример оформления листа задания к курсовой работе

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

УТВЕРЖДАЮ
 и.о. зав. кафедрой «ВТ и АСУ»
 О.В. Игнатьева

_____ 20____

Задание на курсовую работу (проект) № _____

Кафедра:	Вычислительная техника и автоматизированные системы управления
Специальность:	09.03.02 «Информационные системы и технологии»
Форма обучения:	ОЧНАЯ (4 ЛЕТ ОЧНОЕ БАКАЛАВРИАТ)
Дисциплина:	Алгоритмизация и программирование
Вид работы:	КУРСОВАЯ РАБОТА
Группа:	АИБ-1-029
Студент:	Иванов Иван Иванович
Тема курсовой работы (проекта)	Программирование динамических структур данных.
Предприятие	
Исходные данные	<i>Разработать программное приложение на C++ для обработки динамического двусвязного списка, содержащего информацию о автобусах в автобусном парке. Выполнить анализ предметной области, сформулировать требования к разработке, выбрать программные инструменты, составить алгоритм работы программы и построить блок-схему, разработать программное приложение на языке C++, разработать систему тестов и выполнить тестирование приложения, написать инструкцию пользователя. (заполняется рукописно!!)</i>

Руководитель курсовой работы (проекта) _____
 доц. Игнатьева О.В.

Дата выдачи задания « _____ » _____ 20____

Задание получил _____
 Иванов А.А.

« _____ » _____ 20____

Лист содержания

Содержание				
Содержание.....				3
Введение.....				4
1 Исследование динамических структур данных. Постановка проблемы.....				6
1.1 Актуальность проблемы и постановка задачи.....				6
1.2 Описание предметной области.....				6
1.3 Техническое задание.....				7
1.4 Выводы по первому разделу.....				9
2 Проектирование программного приложения.....				10
2.1 Выбор программного инструментария.....				10
2.2 Описание алгоритма программы.....				10
2.3 Проектирование блок-схемы.....				10
2.5 Выводы по второму разделу.....				12
3 Разработка программного приложения обработки ... (вписать название).....				14
3.1 Постановка задачи.....				14
3.2 Описание структуры программного приложения.....				14
3.4 Инструкция пользователя программного приложения.....				15
3.4 Тестирование программного приложения.....				16
3.5 Выводы по третьему разделу.....				16
Заключение.....				17
Список используемых источников.....				19
Приложение А.....				21
Приложение Б.....				23
АИП 01.19 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата
Разработал		Фроленко		
Проверил		Игнатъева		
Программирование динамических структур данных.				
Разработка программного приложения на С++ для обработки бинарного дерева, содержащего информацию о расписании поездов				
Лит	Лист	Листов		
У К Р	3	42		
ФГБОУ ВО РГУПС Кафедра ВТ и АСУ Группа АИБ-1-033				

Рисунок В.1 – Лист содержания

Учебное издание

Игнатьева Олеся Владимировна

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Печатается в авторской редакции
Техническое редактирование К. И. Гамзаевой

Подписано в печать 31.03.21. Формат 60×84/16.
Бумага офсетная. Печать офсетная. Усл. печ. л. 4,88.
Тираж экз. Изд. № 5017. Заказ .

Редакционно-издательский центр ФГБОУ ВО РГУПС

Адрес университета: 344038, г. Ростов н/Д, пл. Ростовского Стрелкового
Полка Народного Ополчения, д. 2.