

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
им. Т.Г. ШЕВЧЕНКО  
Рыбницкий филиал  
*Кафедра прикладной информатики*

**ПРОГРАММИРОВАНИЕ  
И АЛГОРИТМИЗАЦИЯ**

*Методические указания*

Рыбница  
2015

УДК 519.6  
ББК 22.174  
П 78

*Составители:*

**М.А. Скалецкий**, ст. преподаватель

**В.В. Слободянюк**, преподаватель

*Рецензенты:*

**Л.А. Тягульская**, канд. экон. наук, доцент кафедры физики, математики и информатики.

**В.Е. Федоров**, канд. экон. наук, доцент кафедры прикладной информатики.

П78 Программирование и алгоритмизация: Методические указания / Сост.: М.А. Скалецкий, В.В. Слободянюк. – Рыбница, 2015. – 55 с. (в обл.)

*В данной работе приведены методические указания по выполнению курсовой работы по курсу «Программирование и алгоритмизация». В работе подробно рассмотрена структура курсовой работы, даны методические указания по выполнению и оформлению курсовой работы. В пособии также приведен пример выполнения курсовой работы.*

*Методические указания предназначены для студентов, получающих инженерное образование по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств (по отраслям)» профилю подготовки «Автоматизация технологических процессов и производств».*

УДК 519.6  
ББК 22.174

Рекомендовано Научно-методическим советом ПГУ им. Т.Г. Шевченко

© Слободянюк В.В., Скалецкий М.А.,  
составление, 2015

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
I. ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ (ПРОЕКТУ).....	6
1.1. Сущность, цели и задачи курсового проектирования .....	6
1.2. План работы и рекомендации по содержанию курсового проекта .....	7
II. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	13
2.1. Рекомендации по оформлению основного текста работы.....	13
2.2. Оформление графического материала.....	15
III. ПОДГОТОВКА К ЗАЩИТЕ И ЗАЩИТА КУРСОВОЙ РАБОТЫ .....	18
3.1. Защита курсовой работы.....	18
3.2. Оформление презентации для защиты курсовой работы.....	19
IV. ПРИМЕР ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ..	25
ЗАКЛЮЧЕНИЕ .....	49
СПИСОК ЛИТЕРАТУРЫ.....	50
ПРИЛОЖЕНИЯ	

## ВВЕДЕНИЕ

Профессиональная подготовка студентов университета включает в себя перечень исследовательских умений и навыков, которыми следует овладеть будущему специалисту к концу обучения в вузе и которые в дальнейшем будут совершенствоваться самостоятельно.

При выполнении курсовой работы от студента требуется умение самостоятельно решать конкретные задачи из области изучаемой дисциплины, творчески применять полученные знания, умение четко и логично обосновывать свои решения.

Студенты очной формы обучения направления подготовки «Автоматизация технологических процессов и производств (по отраслям)» профиля подготовки 15.03.04 «Автоматизация технологических процессов и производств» выполняют курсовую работу по дисциплине «Программирование и алгоритмизация» во II семестре.

Дисциплина «Программирование и алгоритмизация» входит в состав базовой части учебного цикла Б.3 «Профессиональный цикл» и изучается в первом и во втором семестрах. Знания и навыки, полученные при изучении дисциплины, используются при изучении дисциплин: «Вычислительные машины, системы и сети», «Моделирование систем». Выполнение курсовой работы предполагает предварительное изучение таких дисциплин, как «Информационные технологии», «Математика».

В результате выполнения курсовой работы студенты овладевают следующими общекультурными компетенциями:

– способность использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа

и моделирования, теоретического и экспериментального исследования (ОК-10);

– способность применять основные методы, способы и средства получения, хранения, переработки информации, навыки работы с компьютером как средством управления информацией (ОК-17);

и профессиональными компетенциями:

– способность использовать прикладные программные средства при решении практических задач профессиональной деятельности, методы стандартных испытаний по определению физико-механических свойств и технологических показателей материалов и готовых изделий, стандартные методы их проектирования, прогрессивные методы эксплуатации изделий (ПК-4);

– способность осваивать средства программного обеспечения автоматизации и управления, их сертификации (ПК-26);

– способностью разрабатывать программы, методики, связанные с автоматизацией технологических процессов и производств, инструкции по эксплуатации программного обеспечения (ПК-28);

– способность участвовать в разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления персоналом (ПК-41).

# **I. ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ (ПРОЕКТУ)**

## **1.1. Сущность, цели и задачи курсового проектирования**

Курсовая работа (проект) является важнейшим видом учебной и научной деятельности студента. Написание курсовой работы расширяет и углубляет знания студента по той или иной дисциплине, способствует более глубокому овладению теорией, формирует у него навыки самостоятельной творческой работы.

Курсовая работа является заключительным этапом изучения учебной дисциплины «Программирование и алгоритмизация» и позволяет судить об усвоении учащимися теоретического курса, использования полученных знаний для решения конкретных задач по программированию.

Основной целью курсовой работы является углубление знаний и расширение навыков создания программного обеспечения для решения поставленных задач.

Цели курсовой работы:

- расширение теоретических знаний, полученных при изучении конкретных и смежных дисциплин, а также их закрепление;
- формирование навыков самостоятельной работы при обработке и анализе научной литературы и других источников по теме работы;
- освоение методов научного исследования;
- развитие аналитических способностей, умений структурировать материал и делать обоснованные выводы и рекомендации по рассматриваемой проблеме;
- формирование навыков исследования, которые потребуются при выполнении выпускной квалификационной работы.

Основной задачей курсовой работы является разработка алгоритма и его реализация на языке программирования.

Задачами курсовой работы являются обучение:

- разработке алгоритмов и программ для решения прикладных задач, осуществление постановки и выбор алгоритмов решения задач, в том числе с применением математических методов;

- применению на практике основных алгоритмических структур, графических средств языка;

- использованию алгоритмических языков высокого уровня, средств автоматизации программирования.

В процессе выполнения курсовой работы студент должен показать:

- знания синтаксиса и семантики алгоритмического языка программирования, принципов и методологии построения алгоритмов программных систем;

- знания типов данных и базовых конструкций изучаемого языка программирования;

- знания основных приемов программирования;

- умение проектировать программные алгоритмы и реализовывать их с помощью современных средств программирования;

- владеть навыками проектирования простых блок-схем.

## **1.2. План работы и рекомендации по содержанию курсового проекта**

Структура курсовой работы (проекта) должна отражать логику решения исследуемой проблемы. Традиционно курсовая работа состоит из введения, теоретического и практического разделов, заключения, списка литературы, и, если это необходимо, приложений. Рекомендуемый объём работы (без приложений) – 30-40

страниц, включая список литературы, который, как правило, содержит не менее 15 источников.

Пример структуры курсовой работы «Основные операции над двоичными числами» приведен в приложении 1.

Тема курсовой работы согласовывается с руководителем. Исходя из выбранной темы, формируются конкретные задачи, которые необходимо выполнить в процессе работы над курсовым проектом.

При подготовке к курсовой работе и ее оформлении студенты должны руководствоваться государственными стандартами, использовать научные разработки по исследуемой теме, нормативную документацию, знать фактические достижения и недостатки в соответствующей области знаний.

В числе общих требований к выполняемой работе необходимо указать следующие:

- актуальность выбранной темы;
- высокий теоретический уровень работы;
- комплексность исследования (отражение научно-технических, организационных, социально-психологических и экономических факторов);
- логичность построения, убедительность аргументации, полнота и точность формулировок;
- реальность условий и данных, на базе которых осуществляется исследование;
- тщательность, грамотность оформления текстовой и графической части работы;
- практическая значимость результатов, обоснованность выводов и предложений.

В законченном виде курсовая работа должна содержать 7 основных частей: введение, теоретическое обоснование выбора темы, описание технологической части, экспериментальный раздел, который включает в себя результаты расчетов и анализ проделанной работы, выводы, список литературы, приложения. В некоторых

случаях целесообразно включить справочный материал – таблицы, схемы, графики.

После утверждения выбранной темы студент приступает к ее теоретическому изучению и совместно с руководителем разрабатывается план курсовой работы.

План курсовой работы представляет собой составленный в определенном порядке перечень глав (разделов) и вопросов, которые должны быть освещены в работе. Он определяет ее содержание и последовательность исследования. После утверждения научным руководителем плана курсовой работы студент непосредственно приступает к ее написанию.

### **Содержание основных разделов курсовой работы.**

Структура курсовой работы должна выглядеть следующим образом:

- титульный лист (*приложение 2*);
- содержание (*приложение 3*);
- введение;
- основная часть;
- заключение;
- список литературы (*приложение 4*);
- приложения.

**Введение** представляет собой достаточно ответственную часть текста, в которой отражаются все достоинства работы, элементы новизны, выносимые для руководителя. Все это может окончательно выявиться только на последнем этапе работы, когда автор достиг полной ясности в понимании выбранной темы. Поэтому рекомендуется начинать с основной части текста, над которой придется работать до получения оптимального варианта, а затем только переходить к окончательной формулировке введения. Объем введения составляет обычно 1-2 страницы.

Введение должно включать следующие содержательные компоненты:

- актуальность разрабатываемой темы;
- объект исследования;
- предмет исследования;
- цель работы;
- задачи работы.

**Основная часть** должна дать исчерпывающее представление о проведенной работе, начиная с постановки задач и заканчивая детальным описанием и обоснованием принятых решений. В состав работы включают все материалы, которые поясняют результаты и методику выполнения.

Основная часть работы неизбежно делится на главы (разделы), пункты и подпункты в соответствии с логической структурой изложения. В курсовой работе (проекте) разделов может быть два-три. Каждый раздел или глава должны включать не менее двух-трех пунктов. Названия глав и разделов должны быть сформулированы так, чтобы они не оказались по объему содержания шире всей работы.

Основная часть включает следующие разделы:

- теоретический;
- технологический;
- экспериментальный.

Теоретическое обоснование включает анализ литературы, существующей практики, а также историю вопроса. В этой же части излагаются главные посылки, гипотезы, доказательства основных положений.

Описание технологического раздела отражает самостоятельную работу студента и представляет собой описание программного продукта, его алгоритма и структуры.

Программный продукт должен включать в себя пользовательский модуль и основную программу, работающую с этим модулем с помощью меню. При написании работы студент должен продемонстрировать

умения работать с графикой, файлами, записями, пользовательскими типами.

Студент должен провести сравнительный анализ эффективности использования различных типов структур программ и данных при реализации отдельных фрагментов программы. Таким образом, необходимо обосновать выбор способа создания программного продукта.

При запуске основной программы на экран должна выводиться информация о названии курсовой работы, имя пользователя и задание на курсовую работу.

Экспериментальный раздел работы включают систему тестовой проверки, описание результатов эксперимента и рекомендации для пользователя.

В этом же разделе излагаются предложения (рекомендации, разработки) и оценивается их эффективность на основе результатов исследований, должны быть сформулированы обоснованные предложения по устранению выявленных недостатков проделанной работы и даны рекомендации по её использованию. Для повышения компактности и наглядности используются таблицы и иллюстрации. Язык изложения должен обладать характерными чертами делового стиля с использованием соответствующей терминологии, определенностью формулировок, полным отсутствием эмоциональных речевых средств и т. д.

**В заключении** должны быть четко сформулированы основные выводы и результаты проведенного исследования, приведены показатели эффективности, намечены пути и цели дальнейшей работы над темой.

Иногда целесообразно построить заключение как перечень выводов, разбив его по пунктам, в каждом из которых выделив и обосновав один конкретный вывод.

**Список литературы** содержит согласно стандарту всю литературу, которая была использована при написании работы.

Список литературы должен содержать перечень источников, использованных при выполнении работы, как правило, список содержит не менее 15 источников. Источники должны быть расположены в алфавитном порядке.

Существуют некоторые правила оформления списка использованных источников.

*Книга:*

Бычкова С.М. Планирование в аудите [Текст]/ С.М. Бычкова, А.В. Газорян. – М.: Финансы и статистика, 2001. – 263 с.

*Электронный ресурс:*

Исследовано в России [Электронный ресурс]: многопредмет. науч. журн. / Моск. физ.-техн. ин-т. – Электрон. журн. – Долгопрудный: МФТИ, 1998. – Режим доступа к журн.: <http://zhurnul.milt.rissi.ru>

**Приложение** включает в себя листинги программ, инструкции по использованию программ.

Каждое приложение должно начинаться с новой страницы и иметь заголовок с указанием вверху справа страницы слова «ПРИЛОЖЕНИЕ» и его обозначения. Если приложений более одного, то они обозначаются числами, начиная с 1: ПРИЛОЖЕНИЕ 1.

Нумерация страниц, на которых даются приложения, должна быть сквозной и продолжать общую нумерацию страниц основного текста. Нумерация приложения начинается с титульного листа приложения, но номер на титульном листе не проставляется.

## II. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

### 2.1. Рекомендации по оформлению основного текста работы

Курсовую работу выполняют на листах бумаги формата А4 без рамки в текстовом процессоре *Microsoft Word*. Правила оформления текста курсовой работы указаны ниже:

**Заголовок раздела** должен начинаться с новой страницы. Шрифт заголовка – *Times New Roman*, полужирный, 14 pt. Интервал до заголовка – 0 pt, интервал после заголовка – 20 pt. Заголовок раздела должен быть выровнен по центру. Междустрочный интервал в заголовке из нескольких строк должен быть одинарный.

**Заголовок подраздела (параграфа)** тоже должен начинаться с новой страницы. Шрифт заголовка – *Times New Roman*, полужирный, 14 pt. Интервал до заголовка – 12 pt, интервал после заголовка – 8 pt. Заголовок раздела должен быть выровнен по центру.

**Основной текст.** Шрифт основного текста курсовой работы – *Times New Roman*, 14 pt. Весь текст курсовой работы, кроме заголовков, выравнивается по ширине. Абзацный отступ – 1,25 pt. Межстрочное расстояние – 1,5. Перенос слов допускается только в основном тексте, в заголовках раздела и подраздела переносы не ставятся.

**Подписи к рисункам и заголовки таблиц.** Шрифт – *Times New Roman*, полужирный, 12 pt.

**Параметры документа.** Размер листа бумаги для курсовой работы – А4. Верхнее и нижнее поле – 20 мм, левое – 30 мм, правое – 15 мм.

Текст курсовой работы должен быть набран на компьютере по всей ширине страницы с переносом слов.

Все наименования на иностранных языках должны быть выделены курсивом, например, *Internet*.

Все листы работы нумеруются, начиная с титульного листа. Номер страницы на титульном листе не проставляют. Нумерация страниц текста, входящих в состав работы, должна быть сквозная. Номер страницы проставляется в правом нижнем углу.

Каждый абзац должен начинаться с красной строки. Каждый абзац должен содержать законченную мысль и состоять, как правило, из 4-5 предложений. Слишком крупный абзац затрудняет восприятие смысла.

При печати работы необходимо установить запрет "висячих строк", то есть не допускается перенос на новую страницу или оставление на предыдущей странице одной строки абзаца, состоящего из нескольких строк. Следует избегать также оставления на последней строке абзаца части слова или даже одного целого слова. В этом случае лучше изменить формулировку предложения так, чтобы на последней строке абзаца оставалось не менее трех слов.

Каждая глава работы должна начинаться с новой страницы. Подразделы следуют друг за другом без вынесения нового подраздела на новую страницу. Не рекомендуется начинать новый подраздел внизу страницы, если после заголовка подраздела на странице остается одна-две строки основного текста. В этом случае подраздел необходимо начать с новой страницы. Заголовки глав, а также заголовки введения, заключения, содержания и списка литературы должны быть напечатаны прописными (заглавными) буквами и располагаться посередине строки. Если заголовок состоит из нескольких строк, то интервал между ними должен быть одинарным. Заголовки подразделов начинаются с прописной буквы, последующие буквы – строчные. Точка в конце заголовка не ставится.

Например: **2.2. Экспериментальный раздел**

Не допускается использование подчеркивания в заголовках. Не допускается также использование двух и более типов выделения в заголовках (например, курсив и

жирный шрифт).

Размер символов в математических выражениях не должен превышать размер символов основного текста и должен выделяться курсивом.

Разделы, подразделы, пункты и подпункты (кроме введения, заключения, списка литературы и приложений) нумеруют арабскими цифрами. Внутри пунктов или подпунктов могут быть приведены перечисления, которые отделяют друг от друга точкой с запятой.

Слово «глава» не пишется. Заголовки должны четко и кратко отражать содержание раздела.

Содержание курсовой работы оформляется согласно приложению 2: шрифт *Times New Roman*, 14 pt, интервал – одинарный. Переносы в названиях разделов и подразделов не допускаются.

## 2.2. Оформление графического материала

**Рисунки.** Иллюстрации (чертежи, графики, схемы, диаграммы, фотоснимки, рисунки) следует располагать в работе непосредственно после текста, в котором они упоминаются впервые, или на следующей странице, если в указанном месте они не помещаются. На все иллюстрации должны быть даны ссылки в работе. Иллюстрации должны иметь названия, которые помещают под иллюстрацией. Иллюстрации следует нумеровать арабскими цифрами порядковой нумерацией в пределах глав, например, «Рис. 2.1. Блок-схема» (первый рисунок второго раздела).

Ссылки на иллюстрации в тексте следует указывать в круглых скобках, например: (рис 2.1) Иллюстрации каждого приложения обозначают, например: Рис. 1.3. Вид окна ввода.

Если наименование рисунка занимает более одной строки, то междустрочный интервал должен быть одинарным. Точка по окончании наименования рисунка не ставится.

**Таблицы.** Цифровой материал рекомендуется помещать в работе в виде таблиц. Таблицу следует располагать в работе непосредственно после текста, в котором она упоминается впервые или на следующей странице, а при необходимости – в приложении. Таблицы располагаются по центру документа. До шапки таблицы и после самой таблицы должна быть пустая строка интервалом 1,0 и размером шрифта 12 *pt*. В шапке таблицы переносы слов запрещены.

На все таблицы должны быть ссылки в тексте. При ссылке следует писать сокращенно слово «таблица» с указанием номера, например, «табл. 2.1». Таблицы следует нумеровать арабскими цифрами порядковой нумерацией в пределах глав.

Слово «таблица» и наименование таблицы начинаются с прописной буквы, точка в конце названия таблицы не ставится.

Каждая таблица должна иметь наименование, которое располагается в центре страницы над таблицей, например:

**Таблица 2.1**

**Таблица ответов**

<b>Числа</b>	<b>Восьмеричная система счисления</b>	<b>Шестнадцатеричная система счисления</b>
20	24	14
4	4	4
2	2	2
142	216	8E

Шрифт в таблицах используется *Times New Roman*, 12 *pt*, одинарный интервал. Таблицу следует размещать так, чтобы читать ее без поворота работы, если такое размещение возможно. При переносе таблицы на другую страницу название столбцов таблицы следует повторить. Если шапка таблицы велика, допускается ее не повторять: в этом случае следует пронумеровать графы и повторить их нумерацию на следующей странице.

**Формулы.** В формулах в качестве символов следует применять обозначения, установленные соответствующими стандартами. Пояснения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Набор формул необходимо осуществлять в соответствующем редакторе. Не допускается вставка формул в виде рисунков.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки одинарного интервала, размером шрифта 12 *pt*. Например,

$$A = (x^3 + y^3) \lg(|x^3 + y^3|) + \frac{x^3}{\cos(x + y)}$$

$$B = \arcsin\left(x + \frac{y^2}{\sqrt{x}}\right) + y^{x-y} + \log_2(|x|)$$

Формулы и уравнения в работе следует нумеровать порядковой нумерацией в пределах главы в круглых скобках в крайне правом положении напротив формулы, например,

$$\begin{cases} -\lambda \cdot P_0 + \mu \cdot P_1 = 0, & n = 0 \\ \lambda \cdot P_{n-1} + \mu \cdot P_{n+1} - (\lambda + \mu) \cdot P_n = 0, & n > 0 \end{cases} \quad (3.1)$$

Если в работе только одна формула или уравнение, то их не нумеруют.

**Ссылки.** При ссылке на учебник или учебное пособие после напоминания о нём в тексте работы проставляют в квадратных скобках номер, под которым оно значится в библиографическом списке, например: [9].

## **III. ПОДГОТОВКА К ЗАЩИТЕ И ЗАЩИТА КУРСОВОЙ РАБОТЫ**

### **3.1. Защита курсовой работы**

Условием получения положительной оценки по курсовой работе является не только подготовка текста, но и устная защита. К защите допускаются готовые работы – окончательный вариант, исправленный на основании замечаний руководителя.

Защита курсовой работы включает в себя доклад студента по теме работы и ответы на вопросы преподавателя. Желательно во время защиты использовать графическое представление результатов с помощью презентации. На доклад студенту отводится до 10 минут.

В докладе необходимо обосновать актуальность темы, цели и задачи исследования, а затем осветить основные моменты и результаты работы, сделать выводы.

В ходе защиты цель студента – продемонстрировать глубокое понимание вопросов темы, хорошее владение дополнительным (практическим) материалом по теме.

Доклад целесообразно делать свободно, не прибегая к чтению текста. Такой прием производит благоприятное впечатление. На основные графики, таблицы, представленные в презентации, необходимо ссылаться в ходе доклада.

Не только содержание доклада, но и характер выступления, уверенность в ответах на поставленные вопросы в значительной мере определяют оценку защиты.

Окончание доклада следует отметить заключительной фразой: "Доклад окончен, благодарю за внимание".

Защита должна иметь характер научной дискуссии с анализом достоверности и обоснованности всех выводов и рекомендаций научного и практического характера, которые содержатся в работе.

На защите студенту могут задаваться вопросы, касающиеся не только непосредственно теоретической и практической частей проекта, но и положений курса по программированию. Студент должен дать краткие, четко аргументированные ответы и доказать, что проект выполнен им самостоятельно. После этого на основании содержания и качества выполненного курсового проекта, уровня теоретической и практической подготовки студента выводится общая оценка проекта.

Защита студентов, не ориентирующихся в разработанных проектах (независимо от их качества), признается неудовлетворительной.

### **3.2. Оформление презентации для защиты курсовой работы**

Презентация для защиты курсовой работы разрабатывается в Microsoft Office PowerPoint и должна содержать от 10 до 15 слайдов.

Презентация должна содержать следующие основные элементы:

1. Название (тему) работы.
2. Ф.И.О. автора работы.
3. Цель, объект и предмет исследования, основные задачи.
4. Ключевые идеи исследования.
5. Результаты.

При создании презентаций необходимо добиваться того, чтобы при максимальной информационной насыщенности обеспечить максимальную простоту и прозрачность организации материала для аудитории. В этом случае слушатели на защите быстрее адаптируются к специфике интерфейса и не отвлекаются на него. Также необходимо создать такую систему изобразительных и звуковых образов, которая комплексно будет воздействовать на ассоциативные образы и идеи, на

зрение, слух и воображение аудитории, создаст нужное настроение, поможет прийти к нужным умозаключениям.

Необходимо учитывать некоторые советы по использованию презентации во время защиты:

1. Перед началом выступления проверьте исправность оборудования и техники, продумайте оформление пространства во время своего выступления. На защите вы должны видеть одновременно и слушателей, и экран.

2. Во время выступления не принято полностью зачитывать информацию со слайда.

3. Отбирая информацию на конкретный слайд, помните о том, что на одном слайде не должно быть слишком много слов. Общепринято, что их должно быть не больше 15-30, включая предлоги, числа, заголовки и т.п. Да и самих слайдов для основного выступления может быть не более 10-15.

4. Проверьте тексты каждого слайда на наличие орфографических, синтаксических и прочих ошибок.

5. Помните о том, что во время ответов на вопросы можно воспользоваться своей презентацией.

6. При необходимости привлечения внимания к себе во время выступления можно воспользоваться приемом «чёрный экран»

Условно можно выделить следующие этапы в разработке презентаций:

– определение цели, плана и места презентации, выбор или создание авторского образа будущей презентации;

– определение общей логики и структуры презентации, последовательности и наполнения слайдов;

– коррекция содержимого слайдов в соответствии с принципами информационного дизайна;

– совмещение собственного выступления и чередования слайдов.

При разработке презентации для защиты курсовой работы необходимо руководствоваться основными требованиями стилевого оформления слайдов.

*Единый стиль презентации.* Вся презентация должна быть выдержана в едином стиле, на базе одного шаблона. Стиль включает в себя:

- общую схему шаблона: способ размещения информационных блоков;
- общую цветовую схему дизайна слайда;
- цвет фона или фоновый рисунок;
- параметры шрифтов (гарнитура, цвет, размер);
- способы оформления иллюстраций, схем, диаграмм, таблиц и др.

Необходимо обеспечить унификацию структуры и формы представления учебного материала. Цветовая схема должна быть одинаковой на всех слайдах. Это создает ощущение связности, преемственности, стильности, комфортности. В стилевом оформлении презентации не рекомендуется использовать более 3 основных цветов и более 3 типов шрифта. Следует избегать излишне пёстрых стилей – оформление слайда не должно отвлекать внимание слушателей от содержательной части доносимой информации. При выборе элементов стиля (цветовых соотношений, размера текста, иллюстраций, таблиц) рекомендуется проводить проверку шаблона презентации на удобство чтения с экрана компьютера.

*Правила использования цвета.* При оформлении презентации следует руководствоваться **следующими рекомендациями по использованию цвета в презентации:**

1. На одном слайде рекомендуется использовать не более трех базовых цветов: один для фона, один для заголовка, один для текста.

2. Составление цветовой схемы презентации начинается с выбора:

- a. трех базовых цветов: фона – текста – заголовка;

в. трех главных функциональных цветов, которые используются для представления обычного текста, гиперссылок и посещенных ссылок.

3. Для фона и текста необходимо использовать контрастные цвета: текст должен хорошо читаться, но не резать глаза.

*Правила использования фона.* Фон является элементом заднего (второго) плана, должен выделять, оттенять, подчеркивать информацию, находящуюся на слайде, но не заслонять ее. Легкие пастельные, холодные тона лучше подходят для фона, чем белый цвет. Вместо того чтобы использовать сплошной цвет, лучше выбрать плавный градиентный переход гармонично сочетающихся цветов, мягкую (неконтрастную) текстуру или нейтральный фон. Любой активный фоновый рисунок повышает утомляемость глаз обучаемого и снижает эффективность восприятия материала.

*Правила использования текстовой информации.*

Не рекомендуется:

- перегружать слайд текстовой информацией;
- использовать блоки сплошного текста;
- в нумерованных и маркированных списках использовать уровень вложения глубже двух;
- использовать переносы слов;
- использовать наклонное и вертикальное расположение подписей и текстовых блоков;
- текст слайда не должен повторять текст, который докладчик произносит вслух.

Рекомендуется:

- сжатость и краткость изложения, максимальная информативность текста: короткие тезисы, даты, имена, термины – главные моменты опорного конспекта;
- использование коротких слов и предложений, минимум предлогов, наречий, прилагательных;
- использование нумерованных и маркированных списков вместо сплошного текста;

- использование табличного (матричного) формата предъявления материала;
- основную идею абзаца располагать в самом начале – в первой строке абзаца;
- идеально, если на слайде только заголовок, изображение (фотография, рисунок, диаграмма, схема, таблица и т.п.) и подпись к ней.

*Правила использования шрифтов.* При выборе **шрифтов** для представления информации презентации следует учитывать следующие правила:

1. Не рекомендуется смешивать разные *типы шрифтов* в одной презентации.

2. Учитывая, что *шрифты без засечек* (типа Arial,Tahoma, Verdana и т.п.) легче читать с большого расстояния, чем шрифты с засечками (типа Times New Roman), то:

- для основного текста предпочтительно использовать *шрифты без засечек*;

- для заголовка можно использовать *декоративный шрифт*.

3. Размер шрифта должен быть таким, чтобы его легко можно было прочесть с самого дальнего места.

Рекомендуемые **размеры шрифтов**:

- для заголовков – не менее 32 пунктов и не более 50, оптимально – **36** пунктов;

- для основного текста – не менее 18 пунктов и не более 32, оптимально – **24** пункта.

Не следует злоупотреблять прописными буквами (они читаются хуже строчных), поэтому их допустимо использовать только для смыслового выделения небольших фрагментов текста. Для выделения информации следует использовать цвет, жирный и/или курсивный шрифт.

Выделение подчеркиванием обычно ассоциируется с *гиперссылкой*, поэтому использовать его для иных целей не рекомендуется.

*Правила использования графической информации.* Динамика взаимоотношений визуальных и вербальных элементов и их количество определяются функциональной направленностью материала. Изображение информативнее, нагляднее, оно легче запоминается, чем текст. Поэтому, если можно заменить текст информативной иллюстрацией, то лучше это сделать.

При использовании графики в презентации следует выполнять следующие правила и рекомендации:

1. Графика (рисунки, фотографии, диаграммы, схемы) должна органично дополнять текстовую информацию или передавать её в более наглядном виде.

2. Каждое изображение должно нести смысл: желательно избегать в презентации рисунков, не несущих смысловой нагрузки, если они не являются частью стилевого оформления.

3. Цвет графических изображений не должен резко контрастировать с общим стилевым оформлением слайда.

4. Необходимо использовать изображения **только хорошего качества**. Для этого все изображения, помещаемые в презентацию, должны быть предварительно **подготовлены** в графическом редакторе.

При **подготовке** в графическом редакторе изображения для помещения его на слайд презентации важное значение имеет выбор для него *оптимального размера и разрешения*. Иллюстрации рекомендуется сопровождать пояснительным текстом, пояснительная надпись преимущественно располагается под рисунком. Изображения лучше помещать левее текста: поскольку мы читаем слева направо, то взгляд зрителя вначале обращается на левую сторону слайда. Сложный рисунок или схему следует выводить постепенно. Необходимо четко указать все связи в схемах и диаграммах.

## IV. ПРИМЕР ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

### ВВЕДЕНИЕ

Программирование – это дисциплина, ориентированная не на компьютер, а в конечном итоге, на человека. Поэтому процесс создания программы – это наиболее сложная сфера творческой деятельности человека, требующая больших усилий и специальной технологии разработки.

В основе работы любого компьютера лежит обработка двоичной информации, т.к. именно эта система счисления оказалась наиболее удобной в технической реализации. В то же время более удобной для нас является десятичная система счисления. Переход из одной системы счисления в другую имеет большое значение для обработки информации. Анализ двоичного представления данных необходим во многих случаях: для внутримашинного хранения данных, поиска, сжатия информации, разработки алгоритмов.

С другой стороны подготовка данных для ввода в компьютер выполняется в символьной форме. Преобразование их в машинный код возлагается на специальные программы. В то же время не всегда эти программы способны определить ошибки, допускаемые пользователями при вводе данных с клавиатуры.

*Целью* данной курсовой работы является разработка программного продукта, реализующего систему перевода из десятичной системы счисления в двоичную систему счисления. Также данный программный продукт реализует основные операции над двоичными числами, такие как: сложение и вычитание, умножение и деление. Для достижения поставленной цели были определены следующие *задачи*:

1. Написание и отладка программы на языке программирования.

2. Постановка задачи.

3. Построение модели.

4. Разработка алгоритма.

5. Тестирование программы.

*Объектом исследования* данной курсовой работы является двоичный калькулятор.

*Предметом исследования* курсовой работы является разработка программного продукта, реализующего основные операции над двоичными числами.

# 1. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1. История развития систем счисления

Счисление, нумерация – это совокупность приемов представления натуральных чисел. В любой системе счисления некоторые символы (слова или знаки) служат для обозначения определенных чисел, называемых узловыми, остальные числа (алгоритмические) получаются в результате каких – либо операций из узловых чисел. Системы счисления различаются выбором узловых чисел и способами образования алгоритмических, а с появлением письменных обозначений числовых символов системы счисления стали различаться характером числовых знаков и принципами их записи.

*Система счисления* – совокупность приемов и правил для записи чисел цифровыми знаками.

Наиболее известна десятичная система счисления, в которой для записи чисел используются цифры  $0, 1, \dots, 9$ . Способов записи чисел цифровыми знаками существует бесчисленное множество. Любая предназначенная для практического применения система счисления должна обеспечивать:

- возможность представления любого числа в рассматриваемом диапазоне величин;

- единственность представления (каждой комбинации символов должна соответствовать одна и только одна величина);

- простоту оперирования числами;

В зависимости от способов изображения чисел цифрами, системы счисления делятся на:

- непозиционные,

- позиционные.

*Непозиционной системой* называется такая, в которой количественное значение каждой цифры не зависит от

занимаемой ей позиции в изображении числа (римская система счисления). Позиционной системой счисления называется такая, в которой количественное значение каждой цифры зависит от её позиции в числе (арабская система счисления). Количество знаков или символов, используемых для изображения числа, называется основанием системы счисления.

*Позиционные системы счисления* имеют ряд преимуществ перед непозиционными: удобство выполнения арифметических и логических операций, а также представление больших чисел, поэтому в цифровой технике применяются позиционные системы счисления. Запись чисел может быть представлена в виде:

$$A_{(D)} = D_1 + D_2 + \dots + D_k = \sum_{i=1}^k D_i, \quad (1)$$

где  $A_{(D)}$  – запись числа  $A$  в системе счисления  $D$ ;

$D_i$  – символ системы, образующие базу.

По этому принципу построены непозиционные системы счисления. В общем же случае системы счисления:  $A(B) = a_1 B_1 + a_2 B_2 + \dots + a_n B_n$ . Если положить, что  $B_i = qB_{i-1}$ ,  $B_1 = 1$ , то получим позиционную систему счисления. При  $q = 10$  мы имеем дело с привычной нам десятичной системой счисления. На практике также используют другие системы счисления (табл. 1.1):

**Таблица 1.1**  
**Системы счисления**

<b>q</b>	<b>Название</b>	<b>Цифры</b>
2	двоичная	0,1
3	троичная	0,1,2
8	восьмеричная	0,...,7
16	шестнадцатеричная	0,...,9,A, ...,F

Каждая система счисления имеет свои правила арифметики (таблица умножения, сложения). Поэтому, производя какие-либо операции над числами, надо помнить о системе счисления, в которой они представлены (табл. 1.2).

**Таблица 1.2**  
**Основание систем счисления**

<b>q=10</b>	<b>q=2</b>	<b>q=16</b>
<b>0</b>	0	0
<b>1</b>	1	1
<b>2</b>	10	2
<b>3</b>	11	3
<b>4</b>	100	4
<b>5</b>	101	5
<b>6</b>	110	6
<b>7</b>	111	7
<b>8</b>	1000	8
<b>9</b>	1001	9
<b>10</b>	1010	A
<b>11</b>	1011	B
<b>12</b>	1100	C
<b>13</b>	1101	D
<b>14</b>	1110	E
<b>15</b>	1111	F

Если основание системы  $q$  превышает 10, то цифры, начиная с 10, при записи обозначают прописными буквами латинского:  $A, B, \dots, Z$ . При этом цифре 10 соответствует знак 'A', цифре 11 - знак 'B' и т.д. В таблице ниже приводятся десятичные числа от 0 до 15 и их эквивалент в различных системах счисления:

В позиционной системе счисления число можно представить через его цифры с помощью следующего многочлена относительно  $q$ :

$$A = a_1 q_0 + a_2 q_1 + \dots + a_n q_n \quad (2)$$

Выражение (2) формулирует правило для вычисления числа по его цифрам в  $q$ -ичной системе счисления. Для уменьшения количества вычислений пользуются так называемой схемой Горнера. Она получается поочередным выносом  $q$  за скобки:

$$A = (\dots((a_n q + a_{n-1})q + a_{n-2})q + \dots)q + a_1 \quad (3)$$

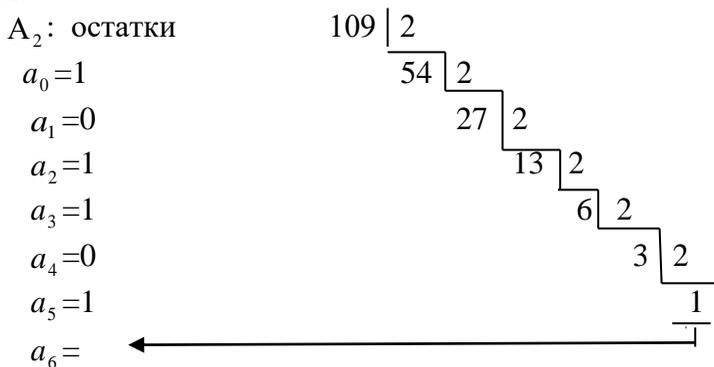
Результат вычисления многочлена будет всегда получен в той системе счисления, в которой будут представлены цифры и основание и по правилам которой будут выполнены операции.

## 1.2. Двоичные системы счисления

В компьютерной технике очень часто используется двоичная система счисления. Двоичная система счисления является позиционной системой. В ней используется две цифры: 0 и 1. В двоичной системе счисления всего две цифры, называемые двоичными (*binary digits*). Сокращение этого наименования привело к появлению термина *бит*, ставшего названием разряда двоичного числа. Веса разрядов в двоичной системе изменяются по степеням двойки. Поскольку вес каждого разряда умножается либо на 0, либо на 1, то в результате значение числа определяется как сумма соответствующих значений степеней двойки. Если какой-либо разряд двоичного числа равен 1, то он называется **значащим разрядом**.

Запись числа в двоичном виде намного длиннее записи в десятичной системе счисления. При выполнении различных операций в современных цифровых системах числа обычно представляются в двоичной системе счисления, основанием которой является число 2. Перевод десятичного числа в двоичный код можно осуществлять путем последовательного деления числа на 2. Остатки (0 или 1), получающиеся на каждом шаге деления, формируют двоичный код преобразуемого числа, начиная

с его младшего разряда. В качестве старшего разряда двоичного кода записывается 1, полученная в результате последнего шага деления. Например, преобразование числа  $A_{10}=109$  в двоичный код выполняется следующим образом:



$$A_{10}=109 = A_2 = a_6 a_5 a_4 a_3 a_2 a_1 a_0 = 1101101$$

Схема 1.1. Преобразование числа в двоичный код

Обратное преобразование выполняется следующим образом:

$$\begin{aligned}
 A_2 &= a_6 a_5 a_4 a_3 a_2 a_1 a_0 = 1101101 = \\
 &= 2^0 + 0^1 + 2^2 + 2^3 + 0^4 + 2^5 + 2^6 = 109 = A_{10} \\
 &1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1
 \end{aligned}$$

Перевод чисел из двоичной системы счисления в восьмеричную. Для записи двоичных чисел используются две цифры, то есть в каждом разряде числа возможны 2 варианта записи. Решаем показательное уравнение:  $2 = 2^i$ . Так как  $2 = 2^1$ , то  $i=1$  бит. Каждый разряд двоичного числа содержит 1 бит информации. Для записи восьмеричных чисел используются восемь цифр, то есть в каждом разряде числа возможны 8 вариантов записи. Решаем показательное уравнение:  $8 = 2^i$ . Так как  $2 = 2^3$ , то  $i=3$  бита. Каждый разряд восьмеричного числа содержит 3 бита информации. Таким образом, для

перевода целого двоичного числа в восьмеричное его нужно разбить на группы по три цифры, справа налево, а затем преобразовать каждую группу в восьмеричную цифру. Если в последней, левой, группе окажется меньше трех цифр, то необходимо ее дополнить слева нулями. Переведем таким способом двоичное число  $101001_2$  в восьмеричное:

$$101001_2 \Rightarrow 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \Rightarrow 51_8$$

Для упрощения перевода можно заранее подготовить таблицу преобразования двоичных триад (групп по 3 цифры) в восьмеричные цифры (табл. 1.3):

**Таблица 1.3**  
**Преобразование двоичных триад**

Двоичные триады	000	001	010	011	100	101	110	111
Восьмеричные цифры	0	1	2	3	4	5	6	7

Для перевода дробного двоичного числа (правильной дроби) в восьмеричное необходимо разбить его на триады слева направо и, если в последней, правой, группе окажется меньше трех цифр, дополнить ее справа нулями. Далее необходимо триады заменить на восьмеричные числа. Например, преобразуем дробное двоичное число  $A_2 = 0,110101_2$  в восьмеричную систему счисления (табл. 1.4):

**Таблица 1.4**  
**Преобразование**

Двоичные триады	110	101
Восьмеричные цифры	6	5

Получаем:  $A_8 = 0,65_8$ . Перевод чисел из двоичной системы счисления в шестнадцатеричную. Для записи шестнадцатеричных чисел используются шестнадцать цифр, то есть в каждом разряде числа возможны 16 вариантов записи. Решаем показательное уравнение:  $16 = 2^i$ . Так как  $16 = 2^4$ , то  $i = 4$  бита. Каждый разряд шестнадцатеричного числа содержит 4 бита информации. Таким образом, для перевода целого двоичного числа в шестнадцатеричное его нужно разбить на группы по четыре цифры (тетрады), начиная справа, и, если в последней левой группе окажется меньше четырех цифр, дополнить ее слева нулями. Для перевода дробного двоичного числа (правильной дроби) в шестнадцатеричное необходимо разбить его на тетрады слева направо и, если в последней правой группе окажется меньше четырех цифр, то необходимо дополнить ее справа нулями. Затем надо преобразовать каждую группу в шестнадцатеричную цифру, воспользовавшись для этого предварительно составленной таблицей соответствия двоичных тетрад и шестнадцатеричных цифр (табл. 1.5). Переведем целое двоичное число  $A_2 = 101001_2$  в шестнадцатеричное:

**Таблица 1.5**

**Таблица соответствий**

Двоичные тетрады	0010	1001
Шестнадцатеричные цифры	2	9

В результате имеем:  $A_{16} = 29_{16}$ . Переведем дробное двоичное число  $A_2 = 0,110101_2$  в шестнадцатеричную систему счисления (табл. 1.6).

Получаем:  $A_{16} = 0, D4_{16}$ . Для того чтобы преобразовать любое двоичное число в восьмеричную или шестнадцатеричную системы счисления, необходимо произвести преобразования по рассмотренным выше алгоритмам отдельно для его целой и дробной частей.

Таблица 1.6

Таблица соответствий

Двоичные триады	1110	0100
Восьмеричные цифры	D	4

### 1.3. Двоичная арифметика

Мы будем рассматривать двоичную систему счисления с цифрами 0,1. Именно эта система счисления получила широкое применение в вычислительных машинах. Начало исследования этой системы относится к XVI веку. Удобство и простоту выполнения арифметических операций в двоичной системе счисления отмечали еще Б. Паскаль, Г. Лейбниц и др. Рассмотрим правила выполнения арифметических операций с двоичными числами.

**Сложение.** Для того чтобы выполнить сложение двух чисел, записанных в двоичной системе счисления, достаточно знать простейшую таблицу сложения:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 10$$

Последняя сумма представляет собой двузначное число. Это следует понимать как перенос одной двоичной единицы в соседний старший разряд. Это можно записать так:  $1 + 1 = 0$  + перенос единицы в соседний старший разряд.

Пример: Сложить двоичные числа:  $x = 110,1011$  и  $x = 10111,10101$ .

Правила арифметики во всех позиционных системах счисления аналогичны. Для выполнения сложения запишем числа столбиком так, чтобы соответствующие разряды чисел оказались друг под другом. Получаем:

$$\begin{array}{r} + 110,1011 \\ \hline 10111,10101 \end{array}$$

10001,00011 – поразрядная сумма без учета переносов.

Сложение нескольких чисел вызывает некоторые трудности, так как в результате поразрядного сложения могут получиться переносы, превышающие единицу. В таких случаях приходится учитывать переносы не только в соседней, но и другие старшие разряды.

**Вычитание.** Таблица вычитания имеет вид:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$10 - 1 = 1.$$

Вычитание в двоичной системе выполняется аналогично вычитанию в десятичной системе счисления. При необходимости, когда в некотором разряде приходится вычитать единицу из нуля, занимается единица из следующего старшего разряда. Если в следующем разряде нуль, то заем делается в ближайшем старшем разряде, в котором стоит единица. При этом следует понимать, что занимаемая единица равна двум единицам данного разряда, т. е. вычитание выполняется по данному правилу.

## II. РЕАЛИЗАЦИЯ ПРОЕКТА

В данном разделе описываются основные возможности разработанного программного продукта. Созданный программный продукт состоит из нескольких различных процедур, которые включают в себя работу с двоичными числами и работу в удобной для пользователя графической среде.

В процедуре *Menu* реализуется выбор одного из пунктов:

1. Теоретический раздел (показывает нам теоретический материал перевода чисел из одной системы счисления в другую и операций над различными двоичными числами).

2. Практический счет:

- перевод из двоичной в десятичную систему;
- перевод из десятичной в двоичную систему;
- сложение двоичных чисел;
- вычитание двоичных чисел;
- операции с двоичными числами.

3. Выход из программы.

В разработанной процедуре *Menu* первоначально используется процедура стандартного в Turbo Pascal модуля *Graph* – *initGraph*, которая инициализирует графическую систему и переводит видеокарту в графический режим. Затем используется еще одна стандартная процедура модуля *Graph* – *SetColor*, которая устанавливает текущий цвет, используя палитру, а *SetColor(5)* делает пятый цвет в палитре цветом текущего рисунка.

Процедура *OutTextXY* посылает строку на устройство вывода. *OutTextXY*, в отличие от *OutText* позволяет вывести текст в любой точке экрана. И процедура *GotoXY* используется для перемещения курсора в заданную позицию экрана.

---

**Код процедуры:**

```
procedure Menu;
begin
  initGraph(a,b,"");
  Setcolor(5);
  outtextxy(100,10,'GLAVNOE MENU');
  outtextxy(100,30,'1-Teoriticheski material');
  outtextxy(100,40,'2-Prakticheskii schet');
  outtextxy(100,50,'3-vihod');
  Gotoxy(25,7);
  outtextxy(100,100,'Vash vibor:');
  read(men);
  Readln;
  closegraph;
end;
```

Процедуры *PERdectodva* и *PERdvatodec* представляет собой диалог с пользователем, в котором осуществляется перевод числа из десятичной системы счисления в двоичную и наоборот, используя при этом процедуры: *DectoDvo* и *Procedure DvatoDec*.

---

**Код процедуры:**

```
procedure PERdectodva;
begin
  initGraph(a,b,"");
  outtextxy(10,5,'Vvedite desiaticnoe chislo ');
  gotoxy(30,1);
  readln(CHIS);
  DvatoDec(CHIS);
  outtextxy(10,20,'dvoichnoe chislo');
  outtextxy(200,20,D);
  Readln;
  closegraph;
end;
```

Процедуры *PERdectodva* подключает стандартный модуль *Graph*. Далее выводит строку «Введите десятичное число» в точке с координатами 10 и 5. Подключает процедуру *DvatoDec*, которая позволяет осуществить перевод числа из десятичной системы счисления в двоичную систему.

В данной процедуре *DvatoDec* представлен цикл с предусловием. Оператор цикла с предусловием выполняет действия заранее неизвестное число раз. Выход из цикла осуществится, если логическое выражение или его результат окажется ложным. Верность логического выражения проверяется в начале: *while CH<>0 do*.

---

#### **Код процедуры:**

```
procedure DvatoDec(CH:integer);
var
  LS,sl,n:integer;
  DD:string;
begin
  DD:="";
  while CH<>0 do
  begin
    ls:=CH mod 2;
    ch:=ch div 2;
    if ls=1 then
      DD:='1'+DD;
    if ls=0 then
      DD:='0'+DD;
  end;
  D:=DD;
end;
```

Процедура *DectoDvo* предназначена для перевода числа из двоичной системы счисления в десятичную систему.

Процедуры *Slojenie* и *Vichit* позволяют складывать и вычитать двоичные числа, использует процедуру инициализации графического режима.

В процедуре *Slojenie* используется стандартный модуль Graph. Программа выводит запрос пользователю для ввода первого двоичного числа, далее подключается процедура *DectoDvo*, которая переводит двоичное число в десятичное и переменной *TMP* присваивается значение числа полученного после перевода из двоичного кода в десятичный. Затем пользователь вводит второе двоичное число, и программа снова переводит его в десятичное число. Далее выполняется сложение  $TMP:=CHIS+TMP$ . Полученная сумма переводится в двоичный код с помощью процедуры *DvatoDec* и ответ сложения двух двоичных чисел выводится на экран.

---

#### **Код процедуры:**

```
procedure Slojenie;
var TMP:integer;
begin
  initGraph(a,b,"");
  outtextxy(10,5,'Vvedite pervoe dvoichnoe chislo');
  gotoxy(35,1);readln(D);
  DectoDvo(D);
  TMP:=CHIS;
  outtextxy(10,20,'Vvedite vtroe dvoichnoe chislo');
  gotoxy(35,2);readln(D);
  DectoDvo(D);
  TMP:=CHIS+TMP;
  DvatoDec(TMP);
  Outtextxy(10,35,'otvet ');
  Outtextxy(100,35,D);
  Readln;
  closegraph;
end;
```

Процедура *Vichit* реализована таким же образом, как и процедура *Slojenie*. Вводятся два двоичных числа, программа переводит их в десятичную систему счисления, а затем вычитает второе число из первого. Затем подключается процедура *DvatoDec*, переводящая число из десятичной системы счисления в двоичную систему и на экране выводится результат вычитания.

---

### **Код процедуры:**

```
procedure Vichit;
var TMP:integer;
begin
  initGraph(a,b,"");
  outtextxy(10,5,'Vvedite pervoe dvoichnoe chislo');
  gotoxy(35,1);readln(D);
  DectoDvo(D);
  TMP:=CHIS;
  outtextxy(10,20,'Vvedite vtroe dvoichnoe chislo');
  gotoxy(35,2);readln(D);
  DectoDvo(D);
  TMP:=TMP-CHIS;
  DvatoDec(TMP);
  Outtextxy(10,35,'otvet ');
  Outtextxy(100,35,D);
  Readln;
  closegraph;
end;
```

Процедура *PERdectoSh* представляет собой диалог с пользователем, в котором осуществляется перевод чисел из десятичной системы счисления в шестнадцатеричную систему, используя при этом процедуру: *DectoSh*.

Процедура *Teoria* знакомит пользователя с теоретическим материалом по бинарной арифметике. В

данной процедуре проверяется не нажата ли какая-нибудь клавиша *keypressED*, если нажата, то считывается информация, то есть открывается выбранный лист. При нажатии клавиши *Enter* осуществляется выход из процедуры.

Процедура *Praktika* позволяет осуществить выбор необходимого пункта меню:

- перевод из десятичной системы счисления в двоичную систему и наоборот;
- перевод из десятичной системы счисления в шестнадцатеричную систему;
- перевод из двоичной системы в восьмеричную систему;
- перевод из десятичной системы в восьмеричную систему счисления;
- перевод из шестнадцатеричной системы в двоичную систему счисления;
- сложение и вычитание двоичных чисел.

---

### **Код процедуры:**

```
procedure Praktika;
begin
  initGraph(a,b,"");
  SPISOK1;
  while RAZDEL<>9 do begin
  case RAZDEL of
    1: PERdvatodec;
    2: PERdectodva;
    3: Slojenie;
    4: Vichit;
  end;
  SPISOK1;
  end;
  closegraph;
  end;
```

В данной программе использованы также некоторые процедуры модуля *Graph*, такие как:

*Rectangle(x1,y2;x2,y2)* - рисует прямоугольник, используя текущий цвет и тип линии. Точка с координатами  $(x1,y1)$  определяет верхний левый угол прямоугольника, а точка с координатами  $(x1,y2)$  определяет нижний правый угол.

*SetColor(color: word)* - используя палитру устанавливает текущий цвет рисунка согласно значению параметра *Color*.

*Circle(x,y:integer;R:word)* - рисует окружность с радиусом R. Точка  $(x,y)$  считается центром окружности.

*Line(x1,y1,x2,y2:integer)* - рисует прямую линию из точки  $(X1,Y1)$  в точку  $(X2,Y2)$ , толщина и тип которой задаются процедурой *SetLineStyle*, а цвет устанавливается процедурой *SetColor*.

*Settextstyle* – стиль заполнения текста.

### III. ЭКСПЕРИМЕНТАЛЬНЫЙ РАЗДЕЛ

Экспериментальный раздел работы включают систему тестовой проверки, описание результатов эксперимента и рекомендации для пользователя.

При запуске программы на экране появляется «Главное меню», состоящее из трех пунктов:

1. Теоретический материал.
2. Практический счет.
3. Выход.

Для выбора интересующего пункта меню необходимо нажать соответствующую цифру и клавишу «Enter».

При нажатии на цифру «1» появляется информация о переводе из десятичной системы счисления в двоичную систему и информация об операциях над двоичными числами.

При нажатии на цифру «2» появляется новое меню:

1. Перевод двоичного числа в десятичное.
2. Перевод десятичного числа в двоичное.
3. Сложение.
4. Вычитание.
5. Выход.

Для выбора интересующего пункта меню необходимо также нажать соответствующую цифру и клавишу «Enter». При нажатии цифры «5» мы возвращаемся в «Главное меню». Для окончательного выхода из программы необходимо нажать цифру «3» в «Главном меню».

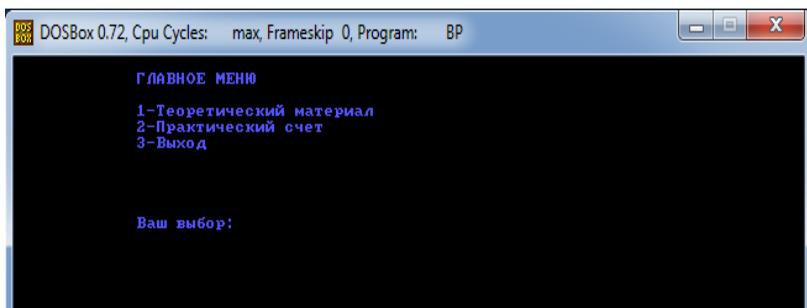


Рис 3.1. Главное меню

При выборе пункта меню «Теоретический раздел» пользователю предоставляется возможность изучить теоретический материал по теме «Двоичные числа» (рис. 3.2).

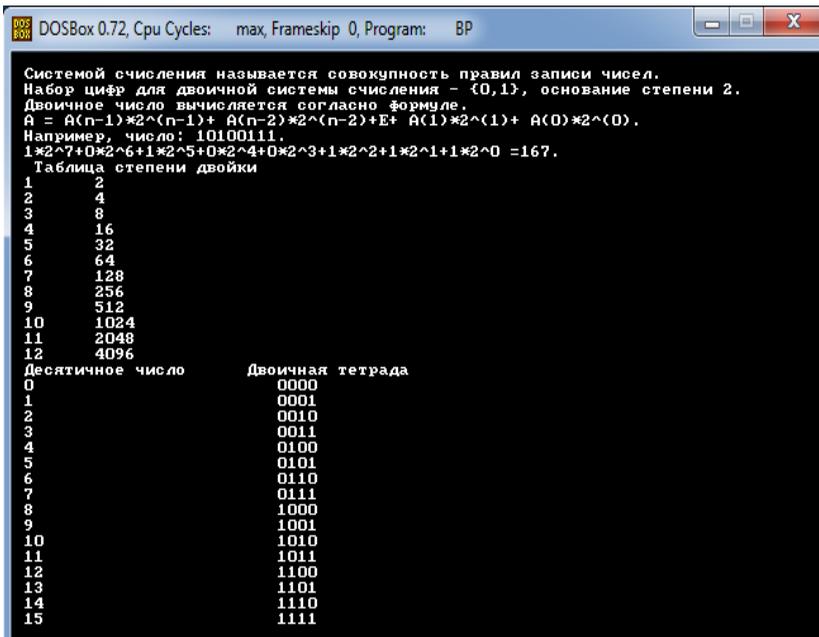


Рис. 3.2. Теоретический материал

Пункт меню «Практический счет» реализует возможности двоичного калькулятора (рис. 3.3).

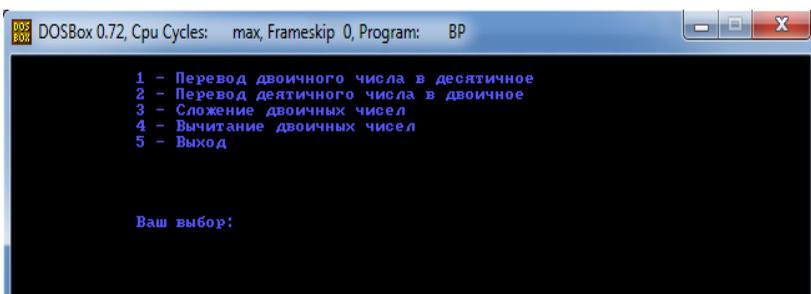


Рис. 3.2. Практический счет

Также для выхода из программы необходимо выбрать пункт меню «Выход».

Для тестовой проверки программы были проведены три эксперимента, которые описаны ниже.

*Эксперимент 1.* Для реализации перевода из двоичной системы счисления в десятичную систему было введено двоичное число: 101000. Результат перевода: 40 (рис. 3.4).

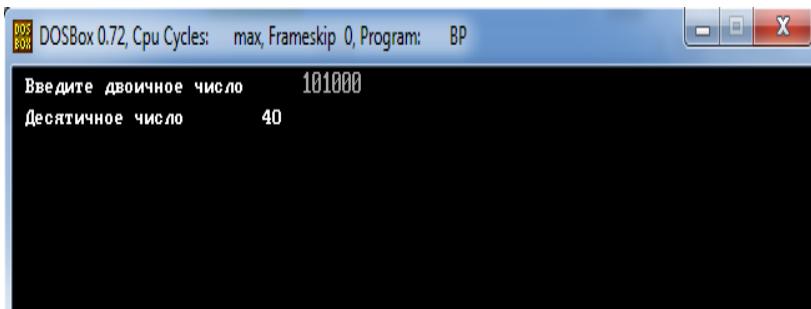


Рис. 3.4. Перевод из двоичной системы счисления в десятичную систему

Для реализации перевода из десятичной системы счисления в двоичную систему было введено двоичное число: / 45. Результат перевода: 101101 (рис. 3.5).

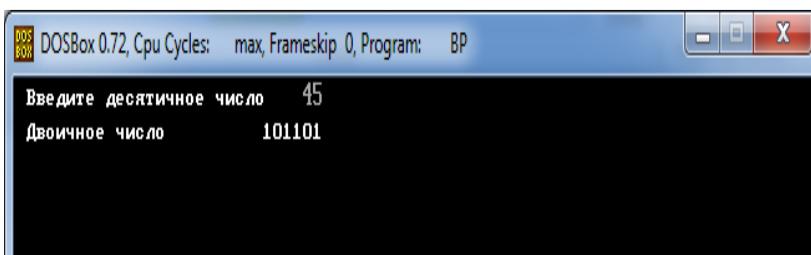


Рис. 3.5. Перевод из десятичной системы счисления в двоичную систему

Для сложения двух двоичных чисел были введены числа: 0011100,1010011. Результат сложения: 1101111 (рис. 3.6).

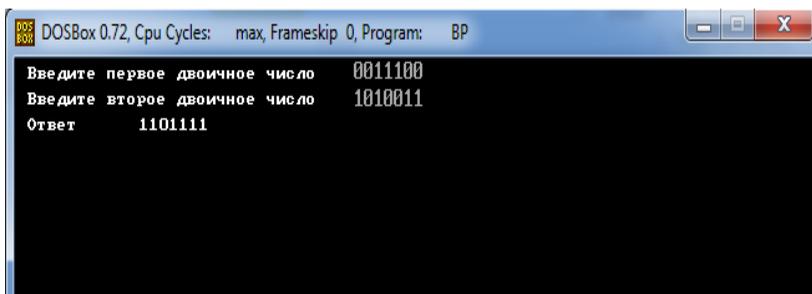


Рис. 3.6. Сложение двоичных чисел

Для вычитания двух двоичных чисел были введены числа: 1101111, 1010011. Результат вычитания: 11100 (рис 3.7).

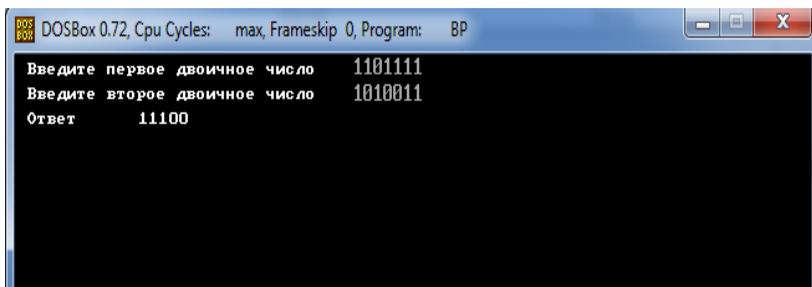


Рис. 3.7. Вычитание двоичных чисел

*Эксперимент 2.* Для реализации перевода из двоичной системы счисления в десятичную систему было введено двоичное число: 010010. Результат перевода: 18 (рис 3.8).

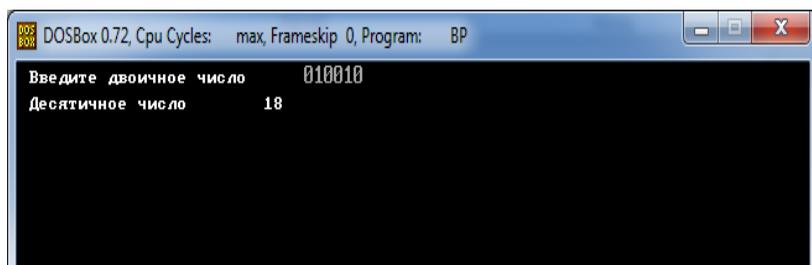


Рис. 3.8. Перевод из двоичной системы счисления в десятичную систему

Для реализации перевода из десятичной системы счисления в двоичную систему было введено двоичное число: 159. Результат перевода: 10011111 (рис. 3.9).

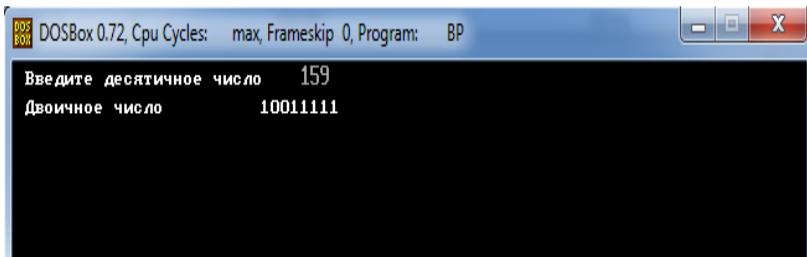


Рис. 3.9. Перевод из десятичной системы счисления в двоичную систему

Для сложения двух двоичных чисел были введены числа: 1110011, 0101001. Результат сложения: 10011100 (рис. 3.10).

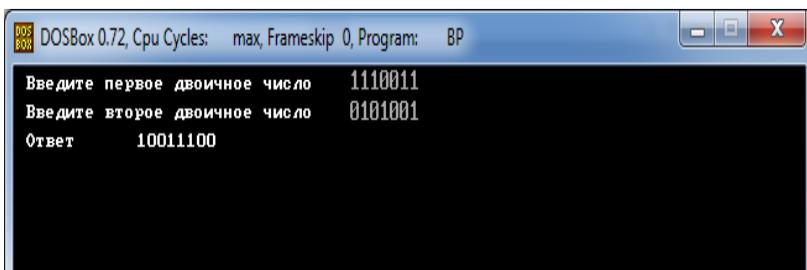


Рис. 3.10. Сложение двоичных чисел

Для вычитания двух двоичных чисел были введены числа: 1110001, 000101. Результат вычитания: 1101100 (рис. 3.11).

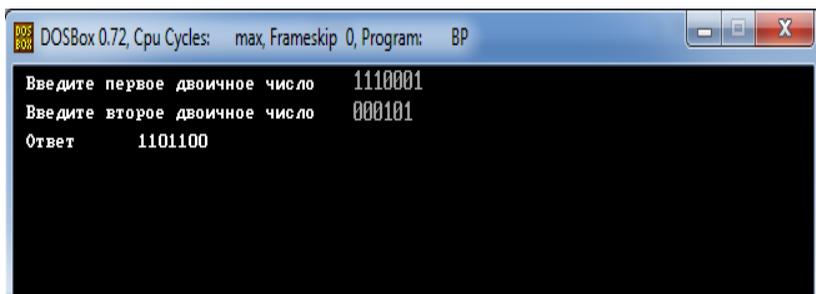


Рис. 3.11. Вычитание двоичных чисел

*Эксперимент 3.* Для реализации перевода из двоичной системы счисления в десятичную систему было введено двоичное число: 110100. Результат перевода: 52

Для реализации перевода из десятичной системы счисления в двоичную систему было введено двоичное число: 78. Результат перевода: 1001110

Для сложения двух двоичных чисел были введены числа: 100110,0010110. Результат сложения: 111100

Для вычитания двух двоичных чисел были введены числа: 1010101,00011. Результат вычитания: 1010010.

## ЗАКЛЮЧЕНИЕ

Современный человек в повседневной жизни постоянно сталкивается с числами: запоминает номера автобусов и телефонов, в магазине подсчитывает стоимость покупок, ведёт свой семейный бюджет в рублях и копейках и т.д. Числа, цифры, они везде.

Понятие числа – фундаментальное понятие, как математики, так и информатики. Сегодня, для записи чисел человечество использует в основном десятичную систему счисления, а компьютерная техника основывается на двоичной системе счисления.

Основной задачей данной работы была разработка программного продукта – двоичный калькулятор на языке программирования *Turbo Pascal*.

В результате выполнения курсовой работы был изучен теоретический материал по системам счисления, а также разработан программный продукт, позволяющий переводить числа из одной системы счисления в другую и выполнять простейшие арифметические операции над ними, такие как сложение и вычитание. Были закреплены знания и умения в написании программ с использованием различных процедур и функций, а также были приобретены навыки работы в графической среде.

Курсовой проект позволяет продемонстрировать знания, полученные при изучении курса программирования. Выполнение курсового проекта требует творческого подхода и применения дополнительных знаний.

## СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 2.105-95. ЕСКД. Общие требования к текстовым документам [Текст]. – Москва: Госстандарт России, 1995.

2. Воронцов Г.А. Письменные работы в вузе [Текст]: Учеб. пособие для студентов./Г.А.Воронцов. – Ростов-на-Дону: Изд.центр. «МарТ», 2002.

3. Кузнецов И.Н. Научное исследование. Методика проведения и оформление [Текст] / И.Н. Кузнецов. – 2-е изд., перераб. и доп. – М.: Издательско-торговая корпорация «Дашков и К<sup>0</sup>», 2006.

4. Научные работы [Текст]: Методика подготовки и оформления / Сост. И.Н.Кузнецов. – 2-е изд., перераб. и доп. – Мн: Амалфея, 2000.

5. Сукиасян Э.Р. Список литературы к курсовой и дипломной работе. Рекомендации по составлению [Текст] / Э.Р. Сукиасян. – Москва, 2001.

6. Ворона В.А. Научные исследования. Концептуальные, теоретические и практические аспекты [Текст]. Учебное пособие для вузов / В.А. Ворона, В.А. Тихонов. – М.: Горячая линия. – Телеком, 2009.

## **ПРИЛОЖЕНИЯ**

# ПРИЛОЖЕНИЕ 1

## Образец оформления титульного листа

ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ им. Т.Г. ШЕВЧЕНКО  
Рыбницкий филиал  
*кафедра прикладной информатики*

КУРСОВАЯ РАБОТА  
по дисциплине «**Программирование и  
алгоритмизация**»  
на тему:  
**«ОСНОВНЫЕ ОПЕРАЦИИ НАД ДВОИЧНЫМИ  
ЧИСЛАМИ»**

студента I курса  
направления «Автоматизация  
технологических процессов и  
производств (по отраслям)»  
группы РФ14ДР62АП  
*Иванова И.П.*

Руководитель:  
преподаватель  
*Слободянюк В.В.*

Рыбница, 2015

## ПРИЛОЖЕНИЕ 2

### Образец оформления содержания

#### СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	25
I. ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ.....	27
1.1. История развития систем счисления.....	27
1.2. Двоичные системы счисления.....	27
1.3. Двоичная арифметика.....	34
II. РЕАЛИЗАЦИЯ ПРОЕКТА.....	36
III. ЭКСПЕРИМЕНТАЛЬНЫЙ РАЗДЕЛ.....	43
ЗАКЛЮЧЕНИЕ.....	49
СПИСОК ЛИТЕРАТУРЫ.....	54

## ПРИЛОЖЕНИЕ 3

### Образец оформления списка литературы

#### СПИСОК ЛИТЕРАТУРЫ

1. Абрамов С.А. и др. Задачи по программированию. – М.: Наука, 1988.
2. Бабушкина И.А. и др. Практикум по Турбо Паскалю. – М.: АБФ, 1998.
3. Быстрицкий В. Редактор блок-схем. Версия 3.0.0.3.
4. Зубов В.С. Программирование на языке Turbo Pascal. – М.: Филинь, 1997.
5. Кассера В. Ф. Turbo Pascal 7.0. – М.: Диасофт, 2003.
6. Кемени Дж., Снелл Дж., Томпсон Дж.. Введение в конечную математику. –М.: Иностранная литература, 1963.
7. Кук Д., Бейз Д. Компьютерная математика. – М.: Наука, 1990.
8. Моргун А. Н. Справочник по Turbo Pascal для студентов. – М.: Диалектика, 2006.
9. Немнюгин С.А. Turbo Pascal. – СПб.: Питер, 2000.
10. Новичков В.С. и др. Паскаль. – М.: В.Ш., 1990.
11. Новиков Ф.А. Дискретная математика для программистов. – СПб.: Питер, 2001.
12. Стюарт Ян. Концепции современной математики. – Минск: Высшая школа, 1980.
13. Фаронов В.В. Turbo Pascal 7.0. – Нолидж, 1999.
14. Фаронов В. В. Turbo Pascal. Наиболее полное руководство. СПб.: ВHV-Санкт-Петербург, 2007.
15. Шиханович Ю.А. Введение в современную математику. – М.: Наука, 1965.

## **ДЛЯ ЗАМЕТОК**

Учебное издание

**Программирование и алгоритмизация**

*Методические указания*

Формат 60 × 90/16. Усл. печ. л. 3,4. Тираж 10 экз.