

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

С.Д. Блинова

Методические указания к выполнению лабораторных работ по дисциплине
«Техническая документация программного обеспечения»

Киров

2013

Рекомендовано к изданию методическим советом
факультета автоматики и вычислительной техники
ФГБОУ ВПО «ВятГУ»

Рецензент А. П. Новосёлов, к.т.н., доцент кафедры автоматики и
телемеханики ФГБОУ ВПО «ВятГУ»

Блинова С.Д.

Методические указания к выполнению лабораторных работ по дисциплине
«Техническая документация программного обеспечения» для студентов направления
подготовки 230100 «Информатика и вычислительная техника», всех профилей
подготовки всех форм обучения / С.Д. Блинова,. — Киров: ПРИП ФГБОУ ВПО
«ВятГУ», 2013. — с.

ISBN xxx-xxxx-xx-x-x

В издании излагаются методика и последовательность выполнения
лабораторных работ по дисциплине «Техническая документация программного
обеспечения». Даются рекомендации по оформлению отчётов по лабораторным
работам, пояснительных записок курсовых работ для дисциплин, связанных с
разработкой и использованием программного обеспечения. Указания к выполнению
лабораторных работ сопровождаются информативными примерами. Примеры
соответствуют требованиям к оформлению технической документации, приведённым в
Государственных стандартах и стандартах ВятГУ.

Данное учебное пособие рассчитано на бакалавров направления 230100.62
«Информатика и вычислительная техника» при изучении дисциплины «Техническая
документация программного обеспечения».

Пособие может быть использовано для оформления: отчётов по лабораторным
работам, пояснительных записок к курсовым проектам и выпускной квалификационной
работе, выполняемым в процессе учёбы в ВятГУ.

УДК 004.4 (07)

Б 695

ББК xx.xx

© ФГБОУ ВПО «ВятГУ», 2013

ISBN xxx-xxxx-xx-x-x

Оглавление

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1 Цели и общие указания к проведению лабораторных работ..... | 4 |
| 2 Первая лабораторная работа. <i>Постановка задачи, разработка схем программы, данных и ресурсов системы, кодирование с комментариями</i> | 5 |
| 2.1 Постановка задачи..... | 5 |
| 2.2 Алгоритм решения задачи | 5 |
| 2.3 Код программы | 6 |
| 2.4 Результаты работы программы | 6 |
| 2.5 Разработка схемы ресурсов системы..... | 6 |
| 2.6 Разработка схемы данных..... | 7 |
| 3 Вторая лабораторная работа. <i>Постановка задачи, разработка схем работы системы и взаимодействия программ, кодирование с комментариями</i> | 8 |
| 3.1 Постановка задачи..... | 8 |
| 3.2 Разработка схемы взаимодействия программ..... | 8 |
| 3.3 Разработка схемы работы системы..... | 9 |
| 3.4 Программирование пользовательского интерфейса | 9 |
| 4 Третья лабораторная работа. <i>Оформление отчёта по лабораторной работе</i> | 11 |
| 4.1 Цель работы | 11 |
| 4.2 Структура отчёта по лабораторной работе | 11 |
| 4.3 Пример постановки и содержательного анализа задачи | 15 |
| 4.4 Пример выбора методов решения задачи | 15 |
| 4.5 Пример описания выбранных методов | 15 |
| 4.6 Пример разработки и описания структуры программы..... | 16 |
| 4.7 Пример описания используемых алгоритмов..... | 16 |
| 4.8 Пример кодирования и отладки программы..... | 17 |
| 4.9 Пример анализа результатов и выводов о проделанной работе | 17 |
| 4.10 Выводы по л работе..... | 20 |
| 5 Четвёртая лабораторная работа. <i>Оформление пояснительной записки курсовой работы</i> | 21 |
| 5.1 Цель работы | 21 |
| 5.2 Оформление фрагмента пояснительной записки курсовой работы | 21 |
| 5.3 Выводы..... | 21 |

| | |
|-------------------------------------------------------------------------------------------------------------|----|
| 6 Пятая лабораторная работа. <i>Оформление реферата и библиографического списка к курсовой работе</i> | 22 |
| 6.2 Теоретическая часть | 22 |
| 6.3 Постановка задачи к лабораторной работе | 25 |
| 6.4 Выводы по лаработе..... | 25 |
| Приложение А (обязательное). Код программы реализации меню второй лабораторной работы..... | 26 |
| Приложение Б (обязательное). Листингпрограммы третьей лабораторной работы..... | 28 |
| Приложение В (рекомендуемое)..... | |
| Пример оформления отчёта по лабораторной работе дисциплины «Информатика» | 30 |
| Приложение Г (рекомендуемое). Фрагмент пояснительной записки | 39 |
| Приложение Д (рекомендуемое). Пример оформления результата | 51 |
| Приложение Е (рекомендуемое). Пример библиографического списка..... | 53 |
| Библиографический список..... | 54 |

1 Цели и общие указания к проведению лабораторных работ

Проведение лабораторных работ по курсу «Техническая документация программного обеспечения» преследует следующие цели:

- познакомиться с государственными стандартами и стандартами предприятий на оформление технической документации;
- научиться грамотно и в соответствии с требованиями стандартов оформлять техническую документацию на разрабатываемое во время учёбы в Вятском государственном университете программное обеспечение;
- научиться грамотно и в соответствии с требованиями стандартов оформлять отчёты по выполняемым лабораторным работам, пояснительные записки к курсовым проектам и выпускной квалификационной работе;
- научиться грамотно формулировать постановку задачи и анализировать полученные результаты.

Указания к выполнению лабораторных работ выполнены в форме примеров (шаблонов) оформления каждой работы. Название работы выделено курсивом. Оно должно совпадать с соответствующим названием на титульном листе к каждому отчёту по лабораторной работе.

2 Первая лабораторная работа. *Постановка задачи, разработка схем программы, данных и ресурсов системы, кодирование с комментариями*

2.1 Постановка задачи

Написать программу перевода целого числа из десятичной системы счисления в новую (от двоичной до девятеричной). В качестве исходных данных взять: целое число A , основание новой системы счисления CC . Результатом должно быть число CA в новой системе счисления CC . В качестве метода решения взять основное правило перевода делением числа на основание новой системы счисления.

2.2 Алгоритм решения задачи

Алгоритм реализует заданный метод решения задачи. Схема программы перевода целой части числа представлена на рисунке 2.1.

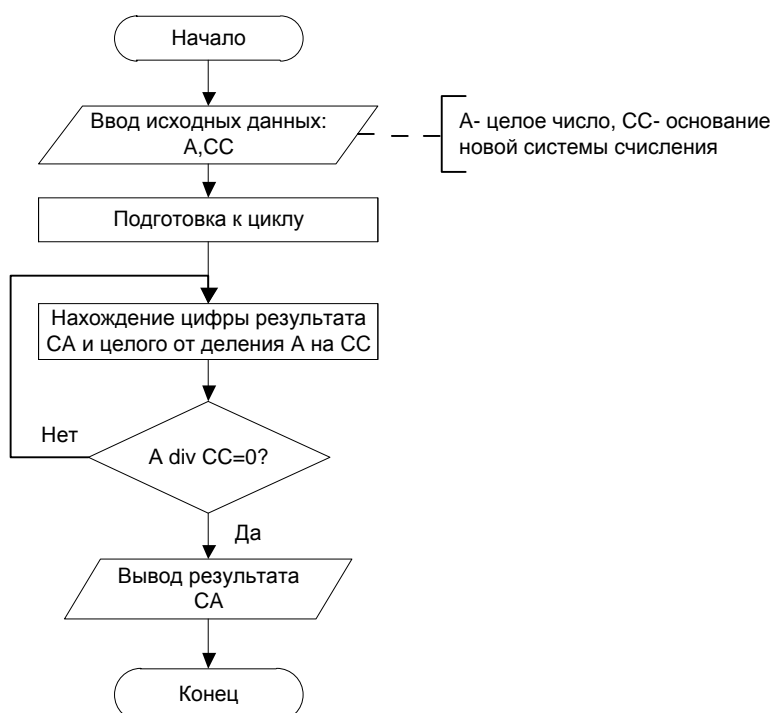


Рисунок 2.1 – Схема программы перевода целой части числа

2.3 Код программы

Листинг программы, реализующей алгоритм перевода целого числа, с пояснениями приведён на рисунке 2.2.

```
Program Perevod;  
Uses crt;  
Var  
A:word;  
cc:byte;  
ca,b:longint;  
Begin  
ClrScr; {Очистка экрана}  
Write ('Введите число и основание новой системы счисления = ');  
Read (a,cc);  
b:=1;  
Repeat {Цикл получения результата}  
ca:=ca+b*(a mod cc); {Результат}  
a:=a div cc; {Целая часть числа}  
b:=b*10; {Для сдвига цифры результата}  
Until (a=0); {Выполнять, пока целая часть не равна нулю}  
write ('Результат = ', ca); {Печать результата}  
ReadKey;  
End.
```

Рисунок 2.2 – Код программы

2.4 Результаты работы программы

Результат работы программы показан на рисунке 2.3.

```
Введите число и основание новой системы счисления = 113 2  
Результат = 1110001
```

Рисунок 2.3 – Результат работы программы

2.5 Разработка схемы ресурсов системы

Схема используемых ресурсов при работе программы показана на рисунке 2.4.

2.6 Разработка схемы данных

Схема данных программы представлена на рисунке 2.5.

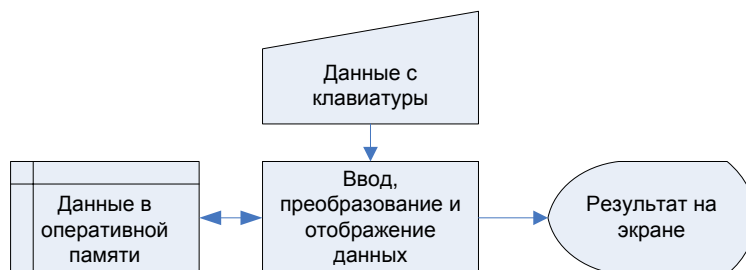


Рисунок 2.4 – Схема ресурсов системы



Рисунок 2.5 – Схема данных

3 Вторая лабораторная работа. *Постановка задачи, разработка схем работы системы и взаимодействия программ, кодирование с комментариями*

3.1 Постановка задачи

Разработать структуру и пользовательский интерфейс программы *Двоичное сложение* - сложения двух двоичных чисел А и В с фиксированной запятой (ФЗ) в обратном коде (ОК), дополнительном коде (ДК), модифицированном дополнительном коде (МДК), а также с плавающей запятой (ПЗ). При разработке предполагается использование соответствующих алгоритмов сложения, каждый из которых будет реализован подпрограммой. При сложении чисел с ПЗ возможно понадобится выполнить нормализацию мантиссы. При использовании модифицированного кода, в случае переполнения разрядной сетки (ПРС), нужно выполнить соответствующую корректировку. Структуру и работу программы показать в виде соответствующих схем. Пользовательский интерфейс запрограммировать с использованием оператора варианта.

3.2 Разработка схемы взаимодействия программ

Структура программы *Двоичное сложение* в виде схемы взаимодействия программ представлена на рисунке 3.1.

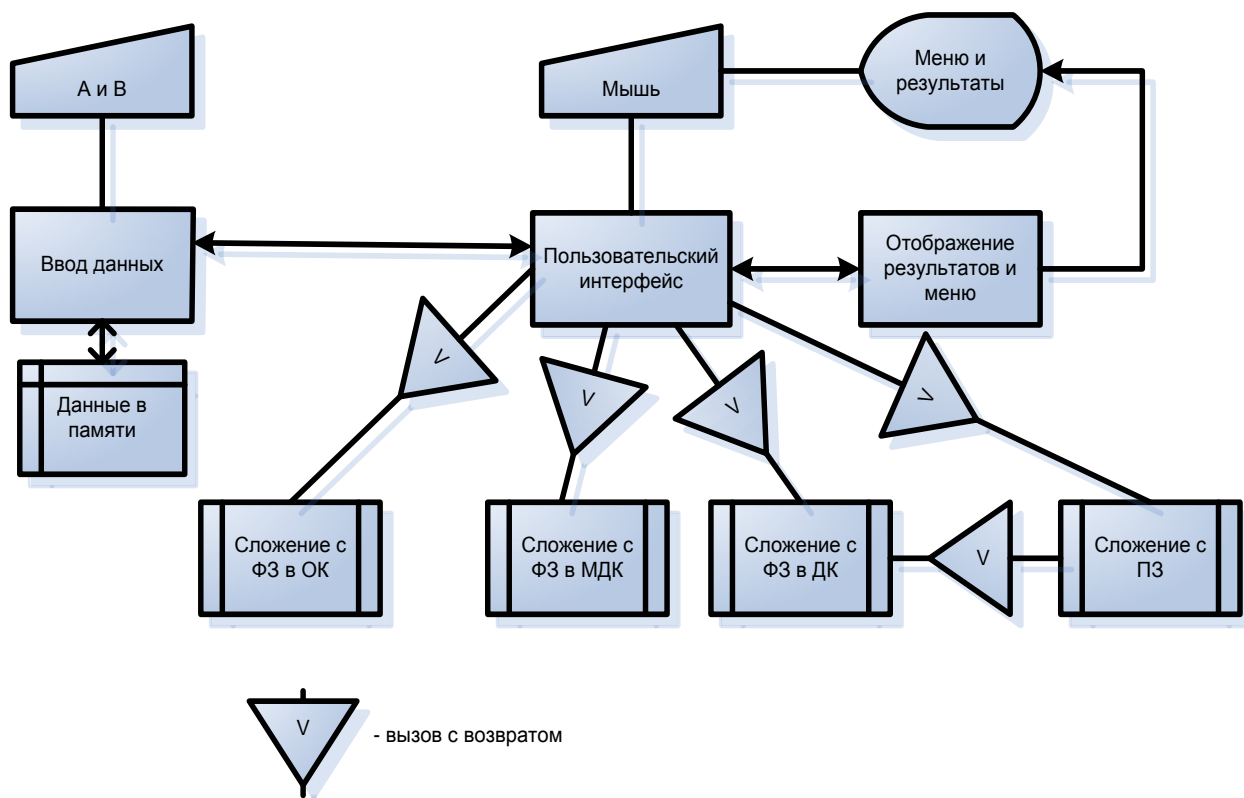


Рисунок 3.1 – Схема взаимодействия программ

3.3 Разработка схемы работы системы

Алгоритм работы программы *Двоичное сложение* в виде схемы работы системы показан на рисунке 3.2.

3.4 Программирование пользовательского интерфейса

Фрагмент кода программы *Двоичное сложение*, реализующий пользовательский интерфейс рисунка 3.1, показан в приложении А.

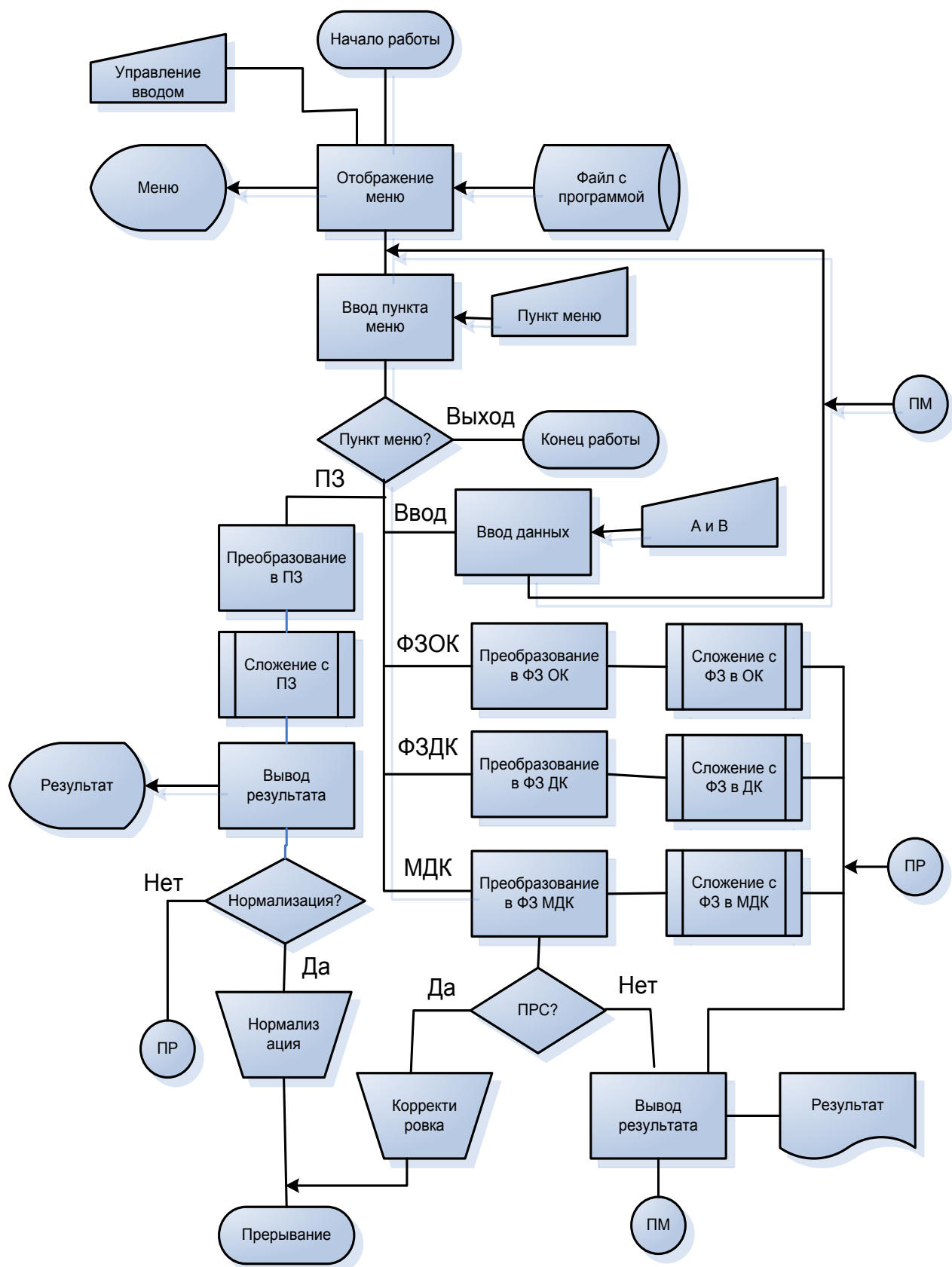


Рисунок 3.2 – Схема работы системы

4 Третья лабораторная работа. *Оформление отчёта по лабораторной работе*

4.1 Цель работы

Цель данной лабораторной работы: научиться оформлять отчёты по лабораторным работам, связанным с разработкой программ, в соответствии с требованиями ЕСКД по оформлению текстовой документации, ЕСПД по оформлению графической части документа, требованиями кафедры и с учётом требований преподавателя – разработчика данной работы. Для достижения цели предлагается разработать программу, по которой затем нужно оформить техническую документацию. Также прилагается пример оформления отчёта по лабораторной работе дисциплины «Информатика», связанной с использованием готовой программы.

4.2 Структура отчёта по лабораторной работе

В общем случае отчёт по лабораторной работе должен иметь следующую структуру:

- титульный лист с точным названием лабораторной работы (по образцу, разработанному на кафедре);
- содержательная часть отчёта.

В содержательную часть отчёта по работе, связанной с разработкой программы, необходимо включить следующие разделы по каждой из требуемых в работе задач:

- цель работы (из методических указаний к лабораторным работам);
- постановка задачи, её точная словесная формулировка;
- содержательный анализ задачи (выбор и формулировка математических моделей постановки задачи, возможно использование схемы данных);
- разработка или выбор методов решения задачи;
- описание методов решения (словесное и/или графическое в виде схемы программы);
- разработка и описание структуры программы для реализации методов (возможно использование схемы взаимодействия программ, схемы ресурсов системы);
- разработка алгоритмов решения и их графическое описание (схемы программы, схемы работы системы);
- реализация алгоритмов (написание, отладка, тестирование программ);
- анализ полученных результатов и выводы по сделанной работе (согласно цели работы).

Постановка задачи или её точная формулировка необходимы для того, чтобы понять задачу. Данный раздел можно объединить с анализом задачи, включив в него анализ исходных данных и ожидаемых результатов. Обычно процесс точной формулировки задачи сводится к постановке правильных вопросов. Например, что дано; что нужно найти; как определить решение; каких данных не хватает, а какие в избытке; какие сделаны допущения при формулировке задачи? Если исходные данные многочисленны и имеют сложную форму, необходимо их структурировать. Здесь же следует установить или перечислить общие

требования к решению задачи. Часто, особенно при решении математических задач, в постановке задачи нужно использовать формулы.

Если в постановке задачи не указан метод её решения, следует его выбрать. Чаще всего бывает необходимо построить или выбрать математическую модель решения. Её выбор существенно влияет на остальные этапы в процессе решения. Невозможно предложить набор правил, автоматизирующих стадию моделирования. Большинство задач должно рассматриваться индивидуально. Наилучший способ приобретения опыта в моделировании – это изучение удачных моделей. Приступая к разработке модели, следует начать с двух вопросов: какие математические структуры больше всего подходят для задачи; существуют ли решённые аналогичные задачи? Второй вопрос часто даёт ответ на первый. При ответе на первый вопрос нужно описать математически, что известно и что нужно найти. На выбор соответствующей структуры будут оказывать влияние такие факторы, как ограниченность знаний выбирающего; удобство представления; простота вычислений; полезность различных операций, связанных с рассматриваемой структурой. Сделав правильный выбор математической структуры, задачу следует переформулировать (формализовать) в терминах соответствующих математических объектов. Это будет одна из возможных моделей, если по ней можно утвердительно ответить на вопросы: вся важная информация хорошо описана математически; есть математическая величина, ассоциируемая с искомым результатом; выявлены полезные отношения между объектами модели; возможно, удобно работать с полученной моделью?

Алгоритм – однозначно трактуемая процедура решения задачи. Процедура – конечная последовательность точно определённых шагов или операций, для выполнения каждой из которых требуется конечный объём оперативной памяти и конечное время. Разработка алгоритма решения задачи начинается, когда задача чётко поставлена и для неё построена

математическая модель. Алгоритм реализует метод разработки. Метод часто сильно зависит от выбора математической модели и может повлиять на эффективность алгоритма решения. Доказательство правильности алгоритма – это один из наиболее трудных и утомительных этапов создания алгоритма. Проверка осуществляется тестированием алгоритма на всех его шагах со всеми возможными данными.

Как только алгоритм выражен в той или иной форме и проверен на правильность, нужно его реализовать, т.е. написать программу. Для этого следует, например, ответить на такие вопросы: каковы основные переменные; каких они типов; сколько нужно массивов и какой размерности; имеет ли смысл пользоваться связными списками; какие нужны подпрограммы; каким языком пользоваться? Конкретная реализация может существенно влиять на требования к памяти и на быстродействие реализации алгоритма. Причины для анализа алгоритмов могут быть, например, такими, как необходимость:

- получить оценки или границы для объёма памяти или времени работы, которые потребуются для успешной обработки конкретных данных по данному алгоритму;
- иметь количественный критерий для сравнения двух алгоритмов решения одной задачи;
- иметь механизм для выявления наиболее эффективных алгоритмов и сравнительных достоинств нескольких из них.

Важно установить абсолютный критерий оптимальности алгоритма.

После написания кода программы и его отладки, следует выполнить программу с исходными данными, получить решение и сделать выводы о правильности полученных результатов, обратившись к сделанному выше содержательному анализу задачи. В выводах по лабораторной работе следует сделать аналитическое заключение о проделанных шагах и процедурах, использованных для получения решения поставленных задач.

При написании выводов следует руководствоваться целями, поставленными разработчиками соответствующих лабораторных работ и сделать заключения об их достижении или не достижении. В последнем случае следует указать причины, по которым цели не достигнуты.

4.3 Пример постановки и содержательного анализа задачи

Требуется сформировать массив и вывести на экран n простых чисел, если известны три первых простых числа: 1, 2, 3. В программе необходимо реализовать несколько методов решения задачи. Ограничить размерность массива 100 элементами.

Входными данными для решения задачи являются: требуемое количество n простых чисел (целое положительное число, вводится с клавиатуры, но не более 100) и значения первых трёх простых чисел (1, 2, 3), которые присваиваются трём первым элементам выходного массива $P(n)$. Выходные данные: массив P из n простых чисел.

4.4 Пример выбора методов решения задачи

Задачу из подраздела 4.3 можно решить тремя методами: простое решение последовательным делением всех нечётных чисел на найденные простые; эффективное решение, уменьшающее число проверок на делимость и использование решета Эратосфена.

4.5 Пример описания выбранных методов

Число называется простым, если оно делится без остатка только на единицу и на само себя. Исходя из данного определения, самый простой метод нахождения простых чисел состоит в следующем. Поскольку известны три первых простых числа, процесс решения следует начинать с

четвёртого. Все последующие простые числа являются нечётными, поэтому каждое из них будет получаться добавлением числа два к предыдущему. Затем следует проверять полученный результат на «простоту» делением данного числа на все ранее найденные простые числа, начиная с тройки. Если остаток от любого такого деления окажется нулевым, то данное число не является простым. Тогда к нему добавляется число два и процесс деления повторяется. Если ни одно из делений не дало нулевого остатка, данное число – простое, оно добавляется в массив, и следующее проверяемое число получается тем же способом.

4.6 Пример разработки и описания структуры программы

Структуру программы для решения задачи из подраздела 4.3 составляют следующие её элементы: ввод исходных данных с проверкой значения на принадлежность диапазону от 0 до 100, пользовательский интерфейс с вариантами выбора методов решения, три процедуры, реализующие, соответственно, алгоритмы трёх методов решения задачи и вывод результирующего массива простых чисел. Элементы программы и связи между элементами в виде схемы взаимодействия программ показаны на рисунке 4.1. Ввод, вывод и пользовательский интерфейс реализованы в основной программе.

4.7 Пример описания используемых алгоритмов

Алгоритм, реализующий простой метод, описанный в подразделе 4.5, в виде схемы программы, показан на рисунке 4.2.

4.8 Пример кодирования и отладки программы

Алгоритм работы программы для решения задачи из подраздела 4.3, в виде схемы работы системы, показан на рисунке 4.3, листинг данной программы с комментариями приведён в приложении Б.

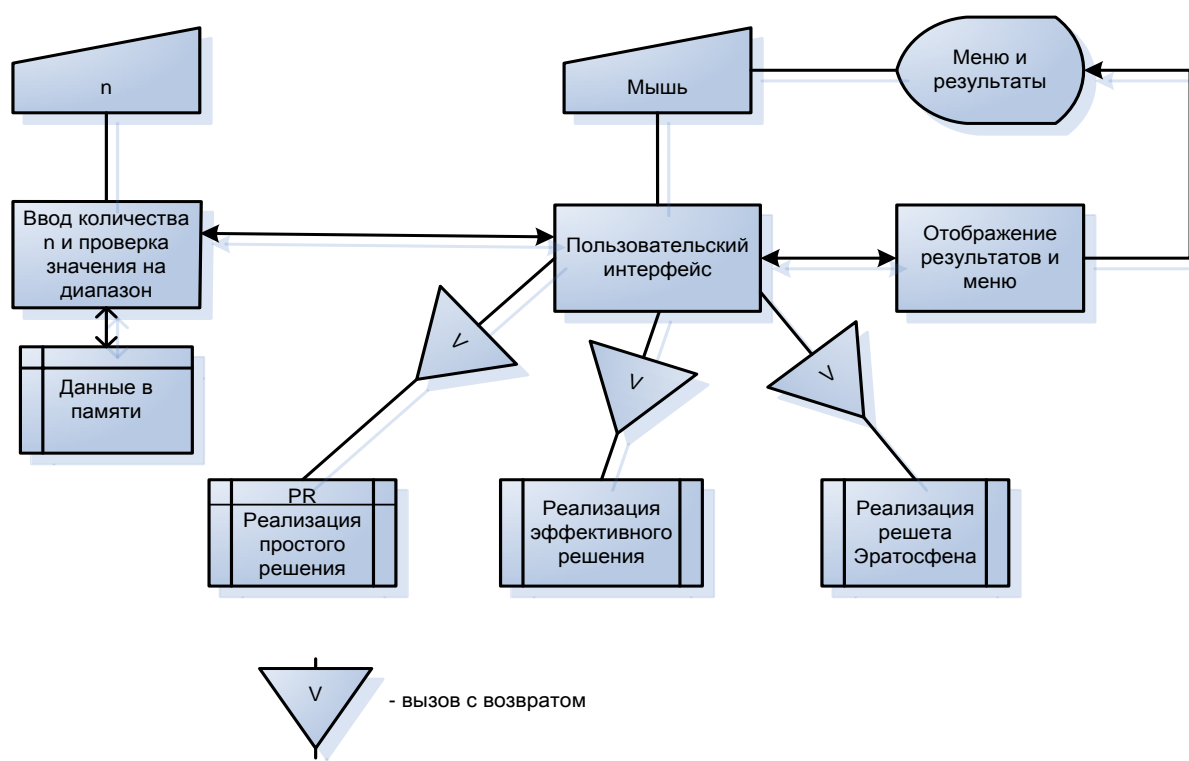


Рисунок 4.1 – Структура программы

4.9 Пример анализа результатов и выводов о проделанной работе

Результатом работы программы из подраздела 4.8 является массив с простыми числами. Количество простых чисел в массиве соответствует требуемому числу, введённому в переменную *n*. Экранные формы результатов работы программы показаны на рисунках 4.4 - 4.5.

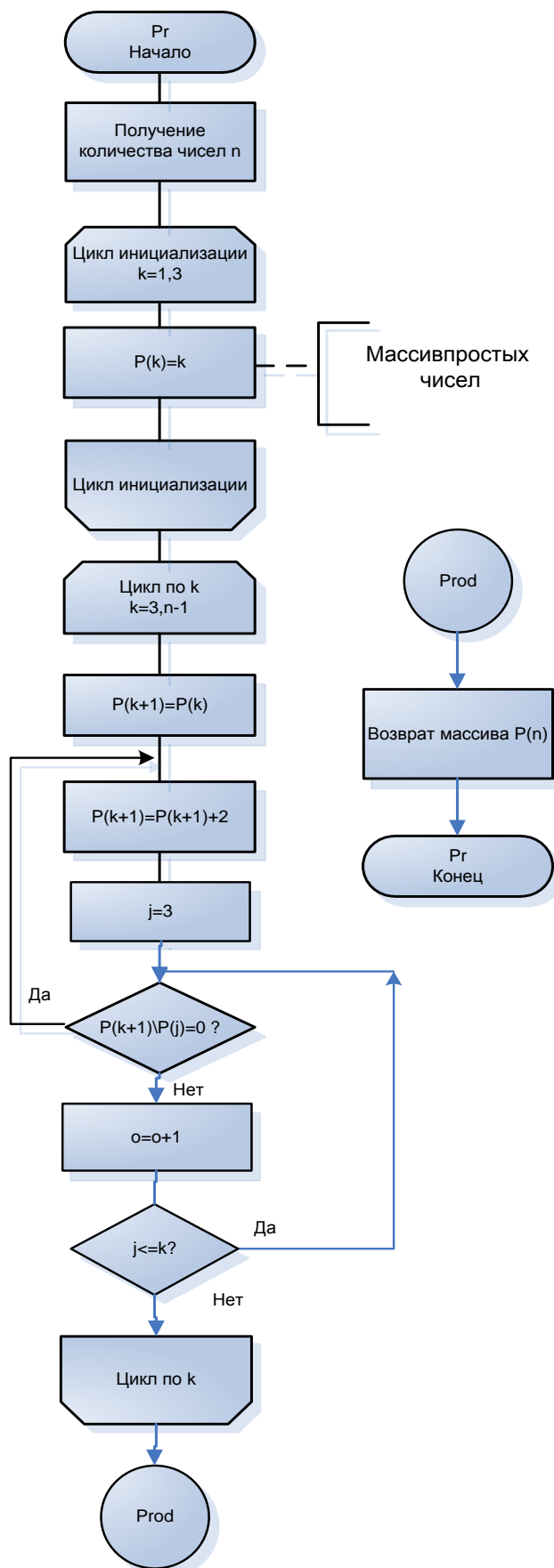


Рисунок 4.2 – Схема алгоритма простого метода

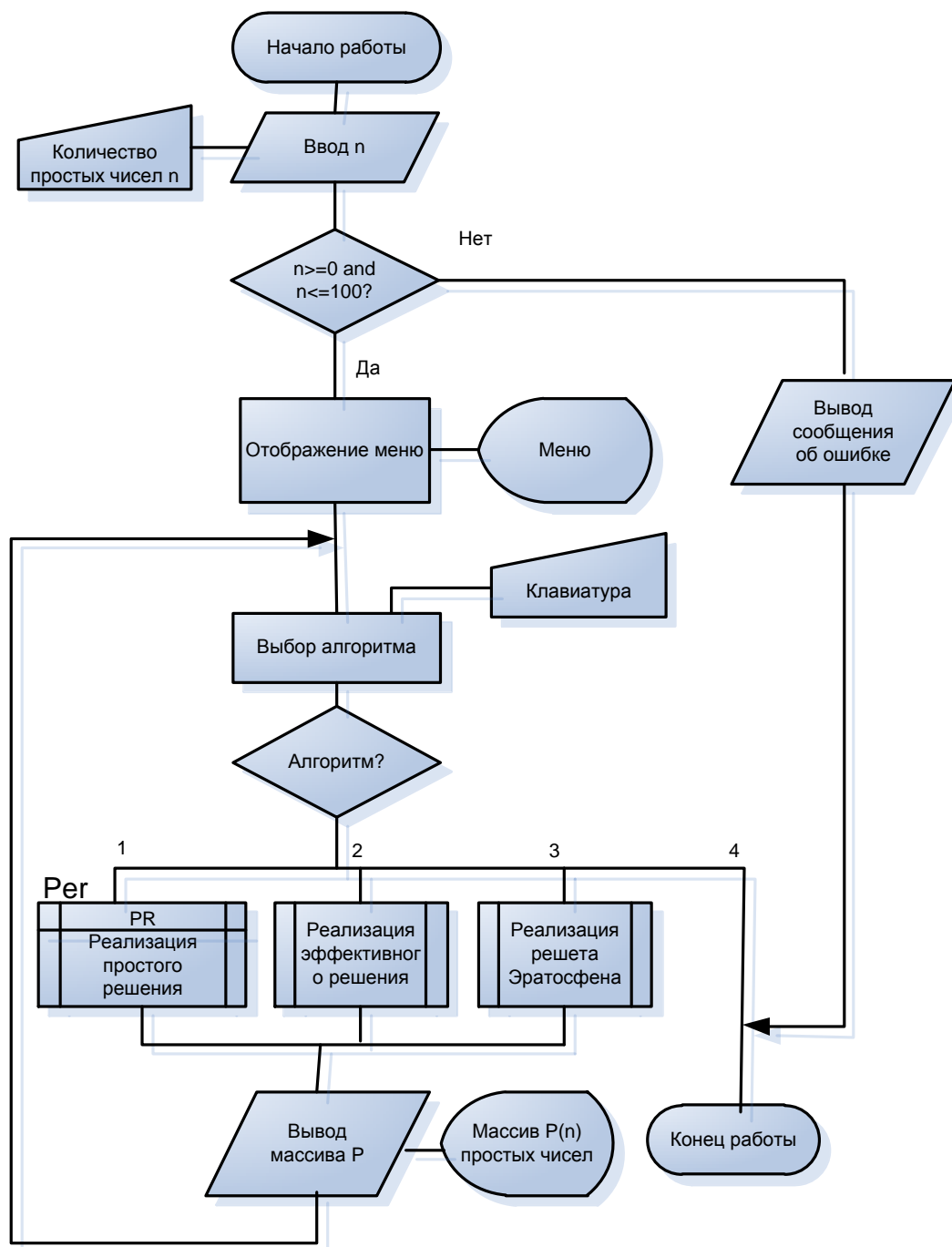
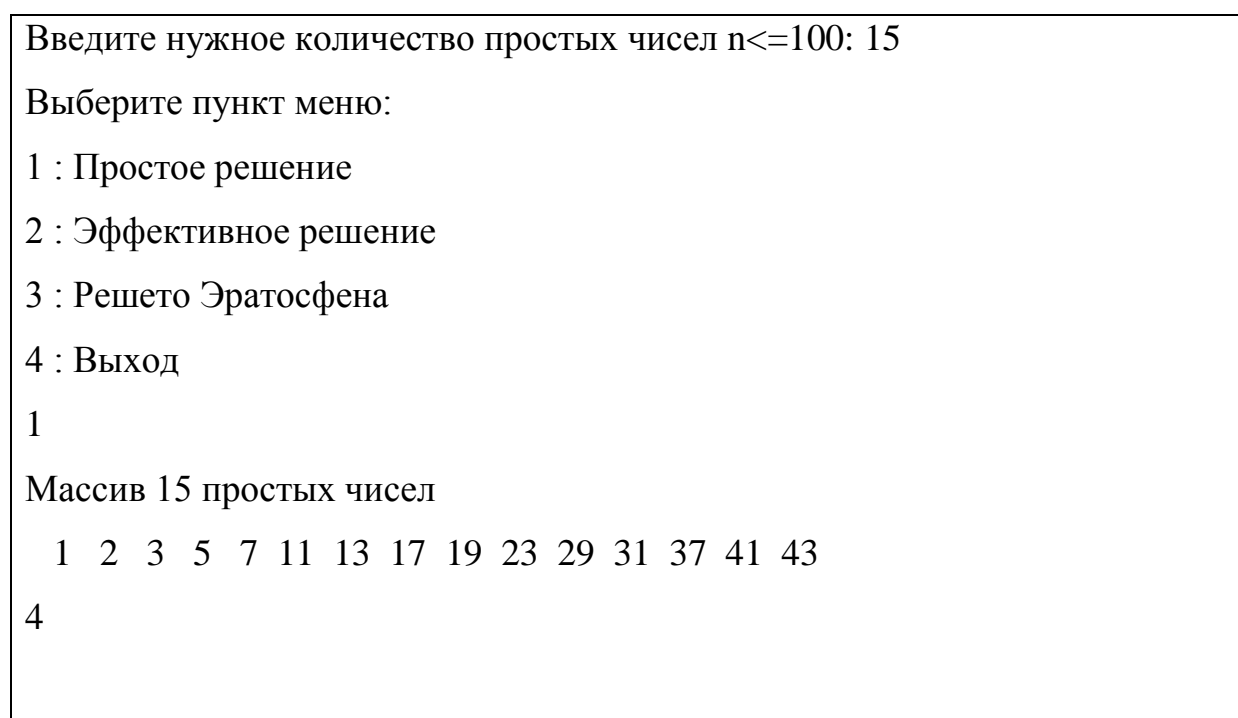


Рисунок 4.3 – Схема алгоритма работы программы

4.10 Выводы по лабораторной работе

В соответствии с целью лабораторной работы был оформлен отчёт о проделанной работе. В нём показан процесс разработки программы и оформления документации на неё.

В приложении В приведён пример оформления отчёта по лабораторной работе, в которой используется программная модель.



Введите нужное количество простых чисел $n \leq 100$: 15

Выберите пункт меню:

- 1 : Простое решение
- 2 : Эффективное решение
- 3 : Решето Эратосфена
- 4 : Выход

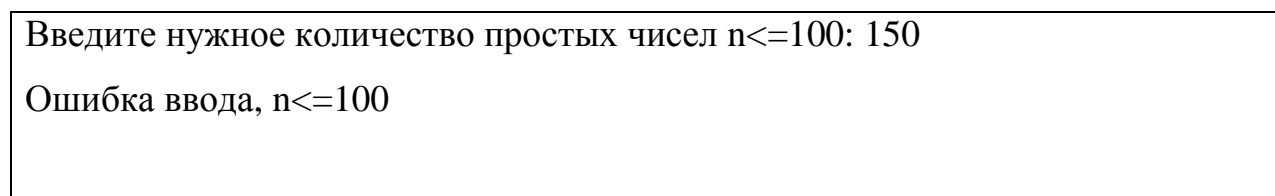
1

Массив 15 простых чисел

1 2 3 5 7 11 13 17 19 23 29 31 37 41 43

4

Рисунок 4.4 – Экранная форма получения пятнадцати простых чисел



Введите нужное количество простых чисел $n \leq 100$: 150

Ошибка ввода, $n \leq 100$

Рисунок 4.5 – Экранная форма превышения диапазона значений количества

5 Четвёртая лабораторная работа. *Оформление пояснительной записки курсовой работы*

5.1 Цель работы

Целью данной лабораторной работы является получение навыков оформления пояснительной записки курсовой работы в соответствии с требованиями нормативных документов. Для достижения цели предлагается выполнить вариант задания из своей курсовой работы и по нему оформить пояснительную записку.

5.2 Оформление фрагмента пояснительной записки курсовой работы

В приложении Г показано оформление фрагмента пояснительной записки курсовой работы по дисциплине «Информатика».

5.3 Выводы по лабораторной работе

В соответствии с целью лабораторной работы был оформлен отчет по проделанной работе. В нем показано оформление фрагмента пояснительной записки курсовой работы по дисциплине «Информатика».

6 Пятая лабораторная работа. Оформление реферата и библиографического списка к курсовой работе

6.1 Цель работы

Целью лабораторной работы является получение навыков в оформлении реферата и библиографического списка к курсовым и выпускной квалификационной работе.

6.2 Теоретическая часть

6.2.1 Реферат – это краткое точное изложение содержания текста первичного документа с основными фактическими сведениями и выводами. Он составляется с учётом требований, сформулированных в [1,2]. Реферат не должен содержать дополнительную интерпретацию или критические замечания автора реферата. Структуру реферата составляют следующие элементы:

- универсальный десятичный код по классификатору универсальной десятичной классификации книг (УДК);
- библиографическая запись;
- ключевые слова и/или сочетания;
- текст реферата.

УДК – иерархическая комбинационная система библиотечно – библиографической классификации, широко используется во всём мире для систематизации науки, литературы и искусства, периодической печати. Код выбранной темы работы ищется в классификаторе (в библиотеке, 4-306) в соответствии с принадлежностью этой темы к той или иной области научно – технической информации. Общий код класса информационных технологий (компьютерных технологий, теории вычислительных машин и систем) – 004, архитектуры вычислительных машин – 004.2.

Библиографическая запись составляется согласно требованиям, изложенным в [1-3].

От пяти до 15 ключевых слов и/или сочетаний должны логически связывать наименование темы работы с текстом реферата и давать максимально сжатую информацию о работе. Они несут существенную смысловую нагрузку с точки зрения информационного поиска.

В тексте реферата содержится информация о содержимом работы. Её необходимо изложить кратко и точно, используя синтаксические конструкции, свойственные языку научно – технических документов. При изложении следует избегать сложных грамматических оборотов. Текст должен отличаться чёткостью, убедительностью формулировок, отсутствием второстепенной информации. Необходимо соблюдать единство терминологии в пределах реферата. В тексте реферата следует применять значимые слова из текста первичного документа для обеспечения автоматизированного поиска.

План текста реферата может быть следующим (пункты плана можно объединять):

- тема, предмет (объект), характер и цель работы, если они не ясны из заглавия документа;
- метод или методология проведения работы (широко известные только называются) или источники данных и характер их обработки (если первичный документ описывает проводимые эксперименты);
- конкретные результаты работы;
- область применения результатов, особенно для патентных документов;
- выводы (рекомендации, оценки, предложения);
- дополнительная информация.

Результаты работы описывают предельно точно и информативно [1]. Приводятся основные теоретические и экспериментальные результаты, фактические данные, обнаруженные взаимосвязи и закономерности. При этом предпочтение отдаётся новым результатам и данным долгосрочного значения, важным открытиям, выводам, опровергающим существующие теории, а также данным, имеющим практическое значение. Следует указать пределы точности и надёжности данных, а также степень их обоснования. Уточняют, являются ли полученные цифровые значения первичными или производными, результатом одного наблюдения или повторных испытаний.

Дополнительная информация может содержать: данные, имеющие значение вне основной темы исследования; название организации, в которой выполнена работа; сведения об авторе работы; ссылки на ранее опубликованные документы по данной теме и т.п.

Рекомендуемый объём реферата – от 850 до 1200 печатных знаков, не более 2000.

Пример оформления реферата приведён в приложении Д.

6.2.2 Библиографический список или список литературы является обязательным для пояснительных записок к курсовым проектам (работам) и выпускной квалификационной работе. Согласно [3] - это последнее приложение текстового документа. Он содержит для каждого источника документа: его номер, область описания, общее обозначение материала согласно требованиям [4]. Пример библиографического списка приведён в приложении Е.

6.3 Постановка задачи к лабораторной работе

Следует написать реферат на курсовую работу по дисциплине «Информатика». При составлении реферата нужно руководствоваться теоретическим материалом из 6.2.1 и приложением А. Далее необходимо составить библиографический список по использованным источникам при работе над курсовой по дисциплине «Информатика». При его составлении следует руководствоваться теоретическим материалом из 6.2.2, приложением Е и библиографическим списком к данному документу.

6.4 Выводы по лабораторной работе

В ходе выполнения лабораторной работы получены навыки в составлении реферата и библиографического списка на примере оформления пояснительной записки курсовой работы по дисциплине «Информатика».

Приложение А

(обязательное)

Код программы реализации меню второй лабораторной работы

```
Program menu;
uses crt;
var
a,b,c,d:integer;
{ Процедуры - заглушки }
procedure FloatDot (k,l:integer;var m:integer);
begin
m:=k+l;
end;
procedure FixOK(k,l:integer;var m:integer);
begin
l:=-l; m:=k+l;
end;
procedure FixDK(k,l:integer;var m:integer);
begin
k:=-k; m:=k+l;
end;
procedure MDK(k,l:integer;var m:integer);
begin
k:=-k; l:=-l; m:=k+l;
end;
{Основная программа}
begin
ClrScr; {Чистка экрана}
{Вывод пунктов меню}
writeln('Выберите пункт меню:');
writeln('1 : Ввод исходных данных');
writeln('2': Сложение с ПЗ);
writeln('3: Сложение с ФЗ в ОК');
writeln('4: Сложение с ФЗ в ДК');
writeln('5: Сложение с ФЗ в МДК');
writeln('6: Выход');
repeat {Цикл опроса}
read (c);
case c of
1: begin
writeln('Введите целые положительные a , b ');
read (a,b);
continue {на конец цикла}
end;
2:FloatDot(a,b,d); (Вызов процедуры сложения с ПЗ)
3:FixOK(a,b,d);{Обращение к процедуре сложения с ФЗ в ОК}
4:FixDk(a,b,d);{Обращение к процедуре сложения с ФЗ в ДК}
```

```
5:MDK(a,b,d); {Вызов процедуры сложения с ФЗ в МДК}  
6:halt;  
else  
end;  
writeln('a+b= ',d);{ Печать результата}  
until false;  
readkey;  
end.
```

Приложение Б
(обязательное)
Листинг программы третьей лабораторной работы

```
program prosto;
uses crt;
type
tp= array [1..100] of integer;
var
n,k,j,c : integer;
p : tp;
procedure per(n:integer;var p:tp);
{Инициализация первых трёх элементов массива простых чисел}
label Metka;
for k:=1 to 3 do p[k]:=k;
{ Формирование последующих элементов массива простых чисел }
for k:=3 to n-1 do
begin
p[k+1]:=p[k];
Metka : p[k+1]:=p[k+1]+2; {Новое число}
{Проверка на делимость}
for j:=3 to k do
if (p[k+1] mod p[j] = 0) then goto Metka {Делится нацело, значит не простое}
end;
procedure Er(n:integer;var p:tp);
begin
end;
procedure Re(n:integer;var p:tp);
begin
end;
{Основная программа}
begin
clrscr;{Чистка экрана}
{Ввод исходного числа элементов массива простых чисел}
write ('Введите нужное количество простых чисел n<=100: ');
readln(n);
if (n >0) and (n<=100) then begin {проверка на диапазон}
{Вывод пунктов меню}
writeln('Выберите пункт меню:');
writeln('1 : Простое решение');
writeln('2 : Эффективное решение');
writeln('3 : Решето Эратосфена');
writeln('4 : Выход');
repeat {Цикл опроса}
read (c);
```

```

case c of
1: per(n,p);
2: Er(n,p);
3: Re(n,p);
4: Halt;
else begin writeln ('Ошибка выбора'); continue end;
end;
{Печать результатов}
writeln('Массив ',n,' простых чисел');
for k:=1 to n do
write(p[k]:4);
writeln;
until false;
end
else
writeln ('Ошибка ввода, n<=100');
readln;
end.

```

Приложение В
(рекомендуемое)
Пример оформления отчёта по лабораторной работе дисциплины
«Информатика»

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Вятский государственный университет»
(ФГБОУ ВПО «ВятГУ»)
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

СПОСОБЫ, АЛГОРИТМЫ УМНОЖЕНИЯ ДВОИЧНЫХ ЧИСЕЛ С ФЗ

Отчет по лабораторной работе дисциплины
«Информатика»

Выполнил студент группы ИВТ-11 _____/Опарин А.Э./
Проверил старший преподаватель _____/Блинова С.Д./

Киров 2013

В.1 Цель работы

Цель данной лабораторной работы: научиться способам и алгоритмам выполнения операций умножения в компьютере на моделях операционных устройств. Любая модель содержит схематические изображения регистров, счетчиков, сумматоров и других элементов операционных устройств с указанием связей между ними и необходимыми наборами управляющих и осведомительных сигналов.

В.2 Общая постановка задачи

Используя программную компьютерную модель, следует перемножить четырьмя способами два числа A и B , представленные в двоичной системе счисления с фиксированной запятой в прямом коде. Исходными данными являются числа: $A = -19_{10} = 1,0010011_2$; $B = 22_{10} = 0,0010110_2$. В результате должно получиться число $C = A * B = -418 = 1,110100010_2$.

В.3 Описания алгоритмов умножения

Алгоритм умножения чисел в прямом коде прост:

- определить знак произведения сложением по модулю два знаковых разрядов сомножителей;
- перемножить модули сомножителей одним из четырех способов умножения и получить $2n$ -разрядный модуль произведения;
- вывести произведение со знаком на шину данных.

Перемножение модулей сомножителей происходит в цикле с использованием трёх элементарных операций:

- анализ цифры очередного разряда множителя;
- суммирование множимого с накапливаемой суммой частичных произведений, если цифра множителя равна единице;
- сдвиги в каждом такте умножения.

Умножение можно выполнять с младших или со старших разрядов n -разрядного множителя, сдвигать можно как сумму частичных произведений, так и множимое, что формирует четыре способа умножения чисел, как показано на рисунке В.1.

| | Сдвиг суммы частичных произведений | Сдвиг множимого |
|------------------------------|------------------------------------------------------------------|------------------------------------------------------------------|
| Сдвиг множителя вправо | I способ | II способ |
| | <p>Множитель Множимое</p> <p>Сумма частичных произведений</p> | <p>Множитель Множимое</p> <p>Сумма частичных произведений</p> |
| Сдвиг множителя влево | III способ | IV способ |
| | <p>Множитель Множимое</p> <p>Сумма частичных произведений</p> | <p>Множитель Множимое</p> <p>Сумма частичных произведений</p> |

Рисунок В.1 - Схемы четырёх способов умножения чисел

Множитель сдвигается во всех способах умножения, так как в каждом такте анализируется очередной разряд: при умножении с младших разрядов сдвиг выполняется вправо - в сторону младших разрядов, при умножении со старших разрядов множитель сдвигается влево. Кроме того, сумма частичных произведений всегда сдвигается в ту же сторону, что и множитель, а множимое сдвигается навстречу множителю, т.е. в противоположную сторону.

В.3.1 Описание алгоритма умножения первым способом

Первый способ осуществляет умножение с младших разрядов n -разрядного множителя со сдвигом $2n$ -разрядного регистра суммы частичных произведений вправо, как это показано на рисунке В.1. Из

рисунка В.1 видно, что множимое следует прибавлять в старшие n разрядов регистра суммы частичных произведений. Причем, разрядность регистра сумм можно уменьшить вдвое, до n -разрядов, помещая при сдвиге младшие разряды суммы на место освобождающихся разрядов регистра множителя.

Особенность первого способа умножения состоит в том, что возможно временное переполнение разрядной сетки (ПРС) в регистре суммы частичных произведений, которое ликвидируется при очередном сдвиге вправо.

В.3.2 Описание алгоритма умножения вторым способом

Второй способ осуществляет умножение с младших разрядов n -разрядного множителя со сдвигом $2n$ -разрядного множимого влево в соответствии с рисунком В.1. Из рисунка В.1 видно, что для умножения требуется n -разрядный регистр множителя и два $2n$ -разрядных регистра множимого и суммы частичных произведений. Причем, первоначально множимое помещается в младшие разряды регистра, а затем в каждом такте сдвигается на один разряд влево.

В.3.3 Описание алгоритма умножения третьим способом

Третий способ - это умножение со старших разрядов n -разрядного множителя со сдвигом $2n$ -разрядного регистра суммы частичных произведений влево в соответствии с рисунком В.1. Из рисунка В.1 видно, что способ требует двух n -разрядных регистров множителя и множимого и одного $2n$ -разрядного регистра суммы частичных произведений, а суммирование множимого следует выполнять в младших n разрядах регистра суммы частичных произведений.

Особенность третьего способа умножения состоит в том, что в последнем такте не следует выполнять сдвиг в регистре суммы частичных произведений.

В.3.4 Описание алгоритма умножения четвёртым способом

Четвёртый способ - это умножение со старших разрядов n -разрядного множителя со сдвигом $2n$ -разрядного множимого вправо в соответствии с рисунком В.1. Данный способ требует наличие одного n -разрядного регистра множителя и двух $2n$ -разрядных регистров множимого и суммы частичных произведений. Причем, первоначально множимое помещается в старшие разряды регистра, а затем в каждом такте сдвигается на один разряд вправо.

Особенность четвертого способа умножения состоит в том, что перед началом цикла умножения следует множимое сдвинуть на один разряд вправо.

В.4 Описание работы программы

Работа программы по реализации задачи, поставленной в разделе В.1, в виде схемы работы системы, показана на рисунке В.2. Для умножения каждым из четырёх способов были составлены и выполнены микропрограммы (МК1, МК2, МК3, МК4, соответственно), состоящие из набора микрокоманд для каждой. Функциональные схемы для их выполнения и результаты выполнения каждым из четырёх способов умножения показаны, соответственно, на рисунках В.3-В.6.

В.5 Анализ полученных результатов

В каждом из четырёх случаев умножения результатом выполнения соответствующей микропрограммы было число C в двоичном коде, значение которого равно произведению двух введенных чисел A и B из раздела 1. Это число равно ожидаемому результату с учётом незначащих нулей, что показывают рисунки В.3-В.6.

В.6 Выводы по работе

Таким образом, в ходе выполнения лабораторной работы были изучены и освоены четыре алгоритма умножения двоичных чисел в прямом коде. Каждым из алгоритмов были перемножены два числа, получен правильный результат. Умножение чисел выполнялось на программной модели, демонстрирующей и реализующей модели операционных устройств на функциональных схемах операций умножения. Последовательность шагов по выполнению каждого алгоритма умножения реализовывалась написанными микрокомандами, состоящими из одной или нескольких микроопераций – элементарных актов обработки информации на одном операционном устройстве за один такт машинного времени под воздействием нескольких управляющих сигналов. Результаты работы каждой микрокоманды демонстрировались на соответствующих операционных устройствах и их выходах.

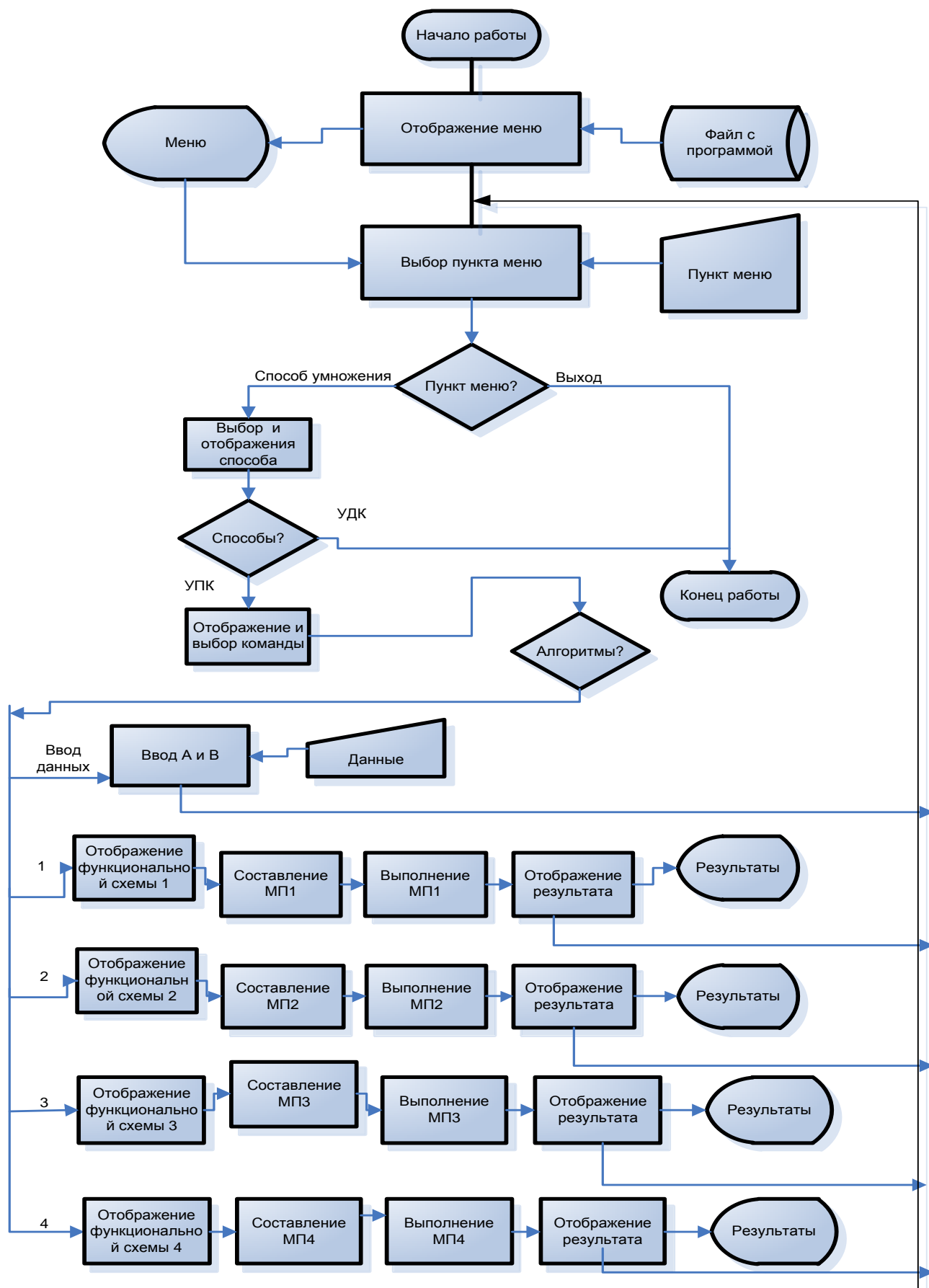


Рисунок В.2 - Схема работы системы

Приложение Г
(рекомендуемое)
Фрагмент пояснительной записки

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Вятский государственный университет»
(ФГБОУ ВПО «ВятГУ»)
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

АРИФМЕТИЧЕСКИЕ ОСНОВЫ ЭВМ

Пояснительная записка курсовой работы по дисциплине
«Информатика»
ТПЖА.230100.62.007 ПЗ

Разработал студент группы ИВТ-12 _____/Альгин М.Н./
Руководитель доцент кафедры
ЭВМ, к.т.н. _____/Фадеева Т.Р./

Проект защищен с оценкой « _____ »
(оценка) (дата)

Члены комиссии _____ / _____/
(подпись) (Ф.И.О)

_____ / _____/

Киров 2013

Содержание

| | |
|------------------------------------------------------------------------------------------------------------------------------------|----|
| 1 Перевод чисел из одной позиционной системы в другую | 3 |
| 1.1 Теоретическая часть | 3 |
| 1.2 Перевод чисел из десятичной системы счисления в двоичную систему счисления..... | 5 |
| 1.3 Изображение чисел в форме с фиксированной запятой | 7 |
| 1.4 Изображение чисел в форме с плавающей запятой | 7 |
| 2 Сложение двоичных чисел | 9 |
| 2.1 Сложение чисел в форме с фиксированной запятой в обратном коде ... | 9 |
| 2.2 Сложение чисел в форме с фиксированной запятой в дополнительном коде | 9 |
| 2.3 Сложение чисел в форме с фиксированной запятой в модифицированном обратном коде или модифицированном дополнительном коде | 10 |
| 2.4 Сложение чисел в форме с плавающей запятой | 11 |
| 3 Умножение двоичных чисел | 13 |
| 3.1 Умножение чисел в форме с фиксированной запятой в прямом коде..... | 15 |
| 3.2 Умножение чисел в форме с фиксированной запятой в дополнительном коде с автоматической коррекцией | 16 |
| 3.3 Умножение чисел в форме с фиксированной запятой в дополнительном коде с простой коррекцией | 18 |
| 3.4 Умножение чисел в форме с плавающей запятой | 20 |
| 4 Деление двоичных чисел | 22 |
| 4.1 Деление с восстановлением остатков | 22 |
| 4.2 Деление без восстановления остатков | 25 |
| 4.3 Деление в дополнительном коде | 27 |
| 4.4 Деление с плавающей запятой | 29 |
| 5 Сложение двоично-десятичных чисел | 31 |
| 5.1 Код с естественными весами (8-4-2-1) | 31 |
| 5.2 Код с избытком три 8-4-2-1 + 3 | 34 |
| 5.3 Код Айкена 2-4-2-1 | 36 |
| 5.4 Пентадный код $3a+2$ | 38 |
| 6 Умножение двоично-десятичных чисел | 40 |
| 6.1 Старорусский метод удвоения – деления пополам | 40 |
| 6.2 Десятично-двоичный метод умножения | 42 |
| Библиографический список | 44 |

Г.4 Деление двоичных чисел

Процесс деления состоит из последовательности операций вычитания и сдвигов, при этом операция вычитания заменяется операцией сложения остатка с делителем, представленным в обратном или дополнительном кодах.

При делении чисел в прямом коде знак частного определяется сложением по модулю два знаковых разрядов делимого и делителя, и далее в процессе деления участвуют модули операндов.

Так как операция деления противоположна умножению и начинается всегда со старших разрядов, существуют два способа деления – обращенный третий и четвертый способы умножения, что видно из рисунка Г.1. Причём, нередко для реализации умножения и деления целесообразно использовать одно и то же оборудование: регистр множимого как регистр делителя, регистр множителя как регистр частного, а регистр частных сумм как регистр делимого, в который затем заносят остатки от деления.

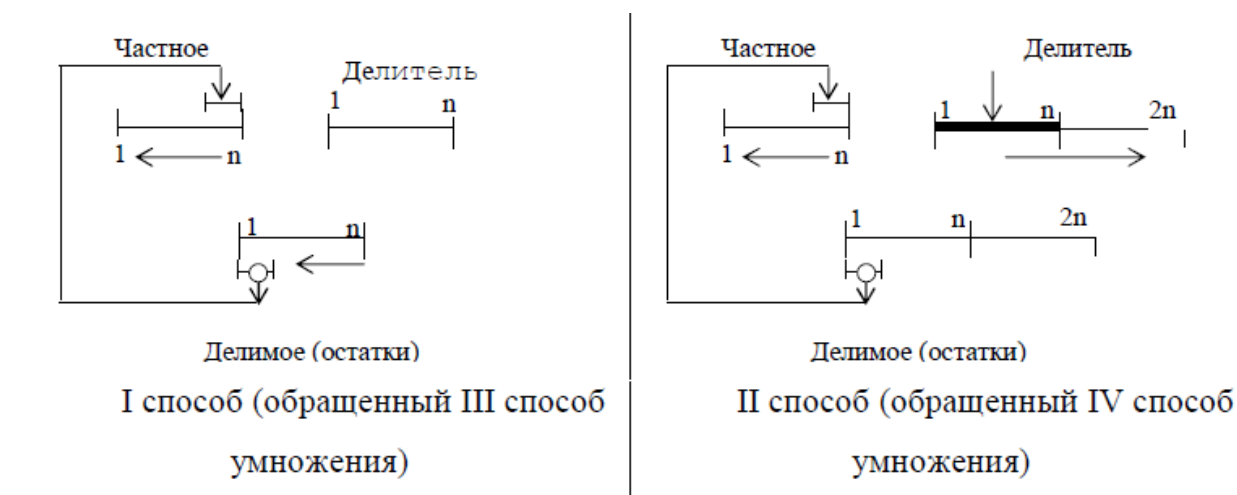


Рисунок Г.1 – Способы деления двоичных чисел

Приведенные два способа деления можно выполнять, используя два алгоритма:

- с восстановлением остатков;
- без восстановления остатков.

Г.4.1 Деление с восстановлением остатков

Г.4.1.1 Алгоритм деления с восстановлением остатков:

- а) определить знак частного сложением по модулю два знаковых разрядов делимого и делителя, далее использовать модули операндов;
- б) вычесть из делимого делитель, путем сложения в обратном коде (ОК) или дополнительном коде (ДК);
- в) проанализировать знак остатка после первого вычитания:
 - 1) если остаток положительный, произошло переполнение разрядной сетки (ПРС), операцию следует прекратить для смены масштабов операндов;
 - 2) если остаток отрицательный, в частное заносится нуль (в последующем, он будет в знаковом разряде), нужно восстановить остаток, прибавив к нему делитель;
- г) выполнить сдвиги;
- д) в цикле формирования цифр частного вычитать из остатка делитель, прибавляя его в ОК или ДК;
- е) проанализировать знак полученного остатка:
 - 1) если больше нуля, то в частное заносится единица;
 - 2) если меньше нуля, то в частное заносится нуль;
- ж) восстановить отрицательный остаток, сложив его с делителем;
- з) выполнить сдвиги;
- и) сформировать $(n+1)$ -ый остаток для округления частного;
- к) выполнить округление результата и присвоить частному знак из пункта а).

Г.4.1.2 Дано: знаки операндов C и D противоположны, $C > 0$, $D < 0$, причём, C - делимое.

Требуется представить числа в форме с фиксированной запятой (ФЗ) в прямом коде (ПК), выполнить деление первым способом, применив алгоритм деления с восстановлением остатков с использованием ОК при вычитании. Проверить результат операции, оценить погрешность округления.

Выполнение задания показано на рисунке Г.2.

$$C = 48_{10} = 110000_2$$

$$D = -54_{10} = -110110_2$$

Масштаб:

Знак произведения: $0 \oplus 1 = 1$;

Деление модулей – первый способ (с восстановлением остатков и с использованием ОК при вычитании):

| Частное ← | Делимое (остатки) ← | Пояснения | Частное ← | Делимое (остатки) ← | Пояснения |
|--------------|-----------------------------------------------------------|---------------------|--------------|-----------------------------------------------|---------------------|
| 0,000000 | 0,110000 1,001001 | Вычитание | 0,001110 | 1,010101 0,110110 | Восстановление |
| 0,000000 | 1,111001 0,110110 | Восстановление | | 10,001011 0,001100 0,011000 1,001001 | Сдвиги Вычитание |
| | 10,101111 0,110000 1,100000 1,001001 | Сдвиги Вычитание | 0,011100 | 1,100001 0,110110 | Восстановление |
| 0,000001 | 10,101001 0,101010 1,010100 1,001001 | Сдвиги Вычитание | | 10,010111 0,011000 0,110000 1,001001 | Сдвиги Вычитание |
| 0,000011 | 10,011101 0,011110 0,111100 1,001001 | Сдвиги Вычитание | 0,111000 | 1,111001 0,110110 | Восстановление |
| 0,000111 | 10,000101 0,000110 0,001100 1,001001 | Сдвиги Вычитание | | 10,101111 0,110000 1,100000 1,001001 | Сдвиги Вычитание |
| | | | 0,111000(1) | 10,101001 0,101010 | Результат |

$$\frac{C}{D} = -0,111001_2 = -0,890625_{10}$$

Проверка: $\frac{48}{-54} \approx -0,888888$

Относительная погрешность: $\delta = \frac{(-0,890625 - (-0,888888))}{-0,890625} * 100\% = 0,2\%$

Рисунок Г.2

Г.4.2 Деление без восстановления остатков

Г.4.2.1 Алгоритм деления без восстановления остатков следующий:

- а) определить знак частного путем сложения по модулю два знаковых разрядов делителя и делимого, далее использовать модули операндов;
- б) вычесть из делимого делитель путем сложения в ДК или ОК;
- в) проанализировать знак остатка после первого вычитания:
 - 1) если положителен, то произошло ПРС, операцию следует прекратить для смены масштаба операндов;
 - 2) если остаток отрицателен, то в частное занести 0 и продолжить операцию деления;
- г) выполнить сдвиги;
- д) если до сдвига остаток был положителен, то вычесть из остатка делитель, если был отрицателен – прибавить к остатку делитель;
- е) если вновь полученный остаток положителен, то в очередной разряд частного занести единицу, в противном случае - нуль;
- ж) выполнить пункты г) – е) алгоритма (n+1) раз, причем последний сдвиг частного не выполнять, так как (n+1)-ый разряд формируется для округления;
- з) выполнить округление результата и присвоить частному знак из пункта а) алгоритма.

Г.4.2.2 Дано: операнды С и D отрицательны; С – делимое.

Требуется выполнить вторым способом деление чисел в форме с ФЗ в ПК, применив алгоритм деления без восстановления остатков с использованием ДК при вычитании.

Реализация данного задания показана на рисунке 3.

$$C = -48_{10} = -110000_2$$

$$D = -54_{10} = -110110_2$$

Масштаб:

Знак произведения: $1 \oplus 1 = 0$;

Деление модулей – второй способ (без восстановления остатков и с использованием ДК при вычитании)

| Частное ← | Делитель → | Делимое (остатки) | Пояснения |
|--------------|----------------|----------------------------------|---------------------|
| 0, | 0,110110000000 | 0,110110000000 1,001010000000 | Вычитание |
| 0,1 | 0,011011000000 | 1,111010000000 0,011011000000 | Сдвиги Сложение |
| 0,11 | 0,001101100000 | 0,010101000000 1,110010100000 | Сдвиги Вычитание |
| 0,111 | 0,000110110000 | 0,000111100000 1,111001010000 | Сдвиги Вычитание |
| 0,1110 | 0,000011011000 | 0,000000110000 1,111100101000 | Сдвиги Вычитание |
| 0,11100 | 0,000001101100 | 1,111101011000 0,000001101100 | Сдвиги Сложение |
| 0,111000 | 0,000000110110 | 1,111111000100 0,000000110110 | Сдвиги Сложение |
| 0,111000(1) | 0,000000011011 | 1,11111111010 0,000000011011 | Сдвиги Сложение |
| | | 0,000000010101 | Результат |

$$\frac{C}{D} = 0,111001_2 = 0,890625_{10}$$

| | |
|----------------------------|-------------------------------------------------------------------|
| Проверка: | $\frac{48}{54} \approx 0,888888$ |
| Относительная погрешность: | $\delta = \frac{(0,890625 - 0,888888)}{0,890625} * 100\% = 0,2\%$ |

Рисунок Г.3

Г.4.3 Деление в дополнительном коде

Г.4.3.1 Алгоритм деления в ДК следующий:

- а) если знаки делимого и делителя совпадают, то в частное заносится 0, в противоположном случае заносится 1, этот разряд знаковый;
- б) если знаки операндов совпадают, то делитель вычитается из делимого, в противном случае делитель прибавляется к делимому;
- в) если знак первого остатка совпадает со знаком делимого, то произошло ПРС, операцию деления прекратить;
- г) выполнить сдвиги;
- д) все последующие остатки формируются по правилу: если знаки делителя и остатка до сдвига совпадают, то делитель вычесть из остатка, в противном случае – делитель прибавить к остатку;
- е) если знаки нового остатка и делителя совпадают, то в очередной разряд частного занести 1, в противном случае заносится 0;
- ж) выполнить пункты г) – е) (n+1) раз, причем последний сдвиг частного не выполнять, выполнить округление результата.

Г.4.3.2 Дано: знаки операндов С и D противоположны, причём $C < 0$, $D > 0$, а D - делимое.

Требуется представить числа в форме с ФЗ в ДК, выполнить деление вторым способом в соответствии с алгоритмом деления в ДК (с автоматической коррекцией) Проверить результат операции, оценить погрешность округления.

Последовательность действий по выполнению задания показана на рисунке 4.

$$C = -48_{10} = -110000_2$$

$$D = 54_{10} = 110110_2$$

Масштаб:

Знак произведения: ;

Деление в ДК – второй способ (с автоматической коррекцией).

| Частное ← | Делитель → | Делимое (остатки) | Пояснения |
|--------------|----------------|----------------------------------|---------------------|
| | 1,010000000000 | 0,110110000000 1,010000000000 | Сложение |
| | | 0,000110000000 | ПРС! |
| 1, | 1,010000000000 | 0,011011000000 1,010000000000 | Сложение |
| 1,0 | 1,101000000000 | 1,101011000000 0,011000000000 | Сдвиги Вычитание |
| 1,01 | 1,110100000000 | 0,000011000000 1,110100000000 | Сдвиги Сложение |
| 1,011 | 1,111010000000 | 1,110111000000 0,000110000000 | Сдвиги Вычитание |
| 1,0110 | 1,111101000000 | 1,111101000000 0,000011000000 | Сдвиги Вычитание |
| 1,01101 | 1,111110100000 | 0,000000000000 1,111110100000 | Сдвиги Сложение |
| 1,011011 | 1,111111010000 | 1,111110100000 0,000000110000 | Сдвиги Вычитание |
| 1,0110111 | 1,111111101000 | 1,111111010000 0,000000011000 | Сдвиги Вычитание |
| 1,0110111(1) | 1,111111110100 | 1,111111101000 0,000000001100 | Сдвиги Вычитание |

| | | | |
|------------------------------------------------------------------------------------------------------------------|--|----------------|-----------|
| | | 1,111111110100 | Результат |
| $\left(\frac{D}{C}\right) = 1,0111000_2$ | | | |
| $\left(\frac{D}{C}\right)^{\text{ПК}} = -0,1001000_2 * \left(\frac{2^7}{2^6}\right) = -1,001000_2 = -1,125_{10}$ | | | |
| <p>ПК</p> | | | |
| <p>Проверка: $\frac{54}{-48} = 1,125$</p> | | | |
| <p>Относительная погрешность: $\delta = \frac{(-1,125 - (-1,125))}{-1,125} * 100\% = 0\%$</p> | | | |

Рисунок Г.4

Г.4.4 Деление с плавающей запятой

Г.4.4.1 Алгоритм деления с плавающей запятой:

- а) определить знак частного путем сложения по модулю два знаковых разрядов операндов;
- б) разделить модуль мантиссы делимого на модуль мантиссы делителя по правилам деления дробных чисел с ФЗ;
- в) определить порядок частного вычитанием порядка делителя из порядка делимого, используя ОК или ДК;
- г) нормализовать мантиссу результата и присвоить знак из пункта а) алгоритма.

В отличие от деления чисел с ФЗ при выполнении пункта б) данного алгоритма, получение положительного остатка при первом вычитании не означает ПРС. При обработке чисел с ПЗ такая ситуация требует денормализации мантиссы делимого сдвигом её на один разряд вправо с одновременным увеличением порядка делимого на единицу.

Г.4.4.2 Дано: оба операнда С и D положительны, причём D – делимое.

Требуется представить числа в форме с ПЗ в разрядной сетке условной машины, разделить числа, используя первый способ деления, алгоритм выбрать самостоятельно. Далее полученное частное необходимо изобразить в разрядной сетке условной машины и проверить результат операции.

Процесс выполнения данных действий показан на рисунке 5.

$$C = 48_{10} = 110000_2$$

$$D = 54_{10} = 110110_2$$

| | Знак числа | Знак порядка | Порядок | Мантисса |
|----|------------|--------------|---------|----------|
| C: | 0 | 0 | 0110 | 110000 |

| | Знак числа | Знак порядка | Порядок | Мантисса |
|----|------------|--------------|---------|----------|
| D: | 0 | 0 | 0110 | 110110 |

Знак произведения: $0 \oplus 0 = 0$;

Деление с ПЗ – первый способ (без восстановления остатков с использованием ДК при вычитании).

| Частное ← | Делимое (остатки) ← | Пояснения | Частное ← | Делимое (остатки) ← | Пояснения |
|--------------|----------------------------------|---------------------|--------------|----------------------------------|--------------------|
| 0,000000 | 0,110110 1,010000 | Вычитание | 0,010010 | 1,010000 0,100000 0,110000 | Сдвиги Сложение |
| 0,000001 | 0,000110 0,001100 1,010000 | Сдвиги Вычитание | 0,010010 | 1,010000 0,100000 0,110000 | Сдвиги Сложение |
| 0,000010 | 1,011100 0,111000 0,110000 | Сдвиги Сложение | 0,100100 | 1,010000 0,100000 0,110000 | Сдвиги Сложение |
| 0,000100 | 1,101000 1,010000 0,110000 | Сдвиги Сложение | 1,001000 | 1,010000 0,100000 0,110000 | Сдвиги Сложение |
| 0,001001 | 0,000000 0,000000 1,010000 | Сдвиги Вычитание | 1,001000(0) | 1,010000 | Результат |
| | | | | | |

Вычисление порядка частного:

$$0.0110$$

$$\underline{1.1010}$$

$$0.0000 = 0_{10}$$

Мантисса нормализована. Результат в разрядной сетке:

| $\frac{D}{C}$: | Знак числа | Знак порядка | Порядок | Мантисса |
|-----------------|------------|--------------|---------|----------|
| | 0 | 0 | 0000 | 1001000 |

$$\left(\frac{D}{C}\right) = 1,001000_2 = 1,001000_2 = 1,125_{10}$$

ПК

Проверка: $\frac{54}{48} = 1,125$. Относительная погрешность: $\delta = \frac{(1,125 - 1,125)}{1,125} * 100\% = 0\%$

Рисунок Г.5

Приложение Д
(рекомендуемое)
Пример оформления реферата

УДК 004.4

Реферат

Рыков А.Н. Автоматизированная система документооборота для ЗАО “Западные электросети”: ТПЖА.230101.115 ПЗ: Дипл. проект / ВятГУ, каф. ЭВМ; рук. О.В.Серов. - Киров, 2006. – Гр. ч. 8 л. ф.А1; ПЗ 180 с., 56 рис., 27 табл., 18 источников, 3 прил.; технол. докум. 7 л., специф. 5 л., програм. докум. 7 л.

ДОКУМЕНТООБОРОТ, ДЕЛОПРОИЗВОДСТВО, ПРОГРАММА, БАЗА ДАННЫХ, ИЕРАРХИЧЕСКАЯ СТРУКТУРА, КОДИРОВАНИЕ ИНФОРМАЦИИ, СЖАТИЕ ИНФОРМАЦИИ, МНЕМОСХЕМА, ШКАФ, ПАПКА, ДОКУМЕНТ

Объект исследования и разработки - программа для ведения документооборота на предприятиях, подобных электросетям.

Цель дипломного проекта - обеспечить пользователей возможностью объединить различные документы в единой базе данных с помощью данного программного продукта.

Разработанная программа позволяет пользователям работать с документами различных типов, создавать междокументные ссылки из документа в документ неограниченной вложенности в базе данных программы, обеспечивает поиск и отбор документов. Она разграничивает доступ к документам, используя систему паролей, обеспечивает резервное копирование и восстановление БД после сбоев.

Разработана программа, реализующая некоторые функции документооборота и делопроизводства, специфичные для предприятий данного типа.

Приложение Е
(рекомендуемое)
Пример библиографического списка

1. Дарахвелидзе, П.Г. Программирование в Delphi 5 [Текст] / Дарахвелидзе П.Г., Марков Е.П., Котенок О.А. – СПб.:БХВ – Санкт-Петербург, 2000. – 784 с.
2. Оформление курсовых и дипломных проектов [Текст]: для студентов специальности 220100 – «Вычислительные машины, комплексы, системы и сети» / Ростовцев В.С., Блинова С.Д. – Киров, 2001 – 33 с.
3. Carter, Alan Программистский камень [Электронный ресурс]: [Электронная книга] Электрон. текстовые, граф. дан. (800 КБ) / Carter Alan, Sanger Colston – Режим доступа: <http://progstone.nm.ru>. – Систем. требования: WINDOWS 95; Pentium 90 Mhz; 16 Mb RAM; CD-ROM drive; VIDEO 2 Mb; mouse, подключение к Internet.
4. Сайт [Электронный ресурс]. - Электрон. текстовые, граф. дан. - Режим доступа: <http://www.icn.ru>. – Систем. требования: ПК 486 или выше; WINDOWS 95 и выше; 8 Мб ОЗУ; SVGA 32768 и более цв.; 640x480; мышь; подключение к Internet.

Библиографический список

1. ГОСТ 7.9-95 (ИСО 214-76) Система стандартов по информации, библиотечному и издательскому делу. Реферат и аннотация. Общие требования [Текст]. – Введ. 01.07.97. Минск: ИПК Изд-во стандартов, 1996. – 7с. – (Межгосударственный стандарт).
2. СТП ВятГУ 103 – 2004 Общие требования к структуре, представлению и оформлению дипломных проектов и работ [Текст]. – Киров, 2004.
3. ГОСТ 2.105-95 Единая система конструкторской документации. Общие требования к текстовым документам [Текст]. – Введ. с 01.07.96. – Минск: ИПК Изд-во стандартов, 1996. – 36с. – (Межгосударственный стандарт).
4. ГОСТ 7.1-2003 Система стандартов по информации, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления [Текст]. – Введ. с 01.07.2004. – М.:ИПК Издательство стандартов, 2004.