

Кафедра «Горные машины»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению курсовой работы по дисциплине «Информатика»
для студентов специальностей горного профиля
заочной формы обучения



Минск 2005

В методических указаниях приводятся сведения о составе и объеме курсовой работы по дисциплине «Информатика», последовательности её выполнения, содержании пояснительной записки, а также требования к оформлению и представлению материалов курсовой работы на кафедру, порядок её защиты. В приложении представлен пример оформления пояснительной записки.

Составители:

С.М.Петренко, И.В.Недашковская

Рецензенты:

Содержание

Введение.....	4
1. Состав и объем курсовой работы.....	4
2. Последовательность выполнения курсовой работы	5
3. Содержание пояснительной записки.....	5
4. Организация консультирования.....	10
5. Защита курсовой работы.....	10
Рекомендуемая литература.....	12
Приложение 1.....	14
Приложение 2.....	16

Введение

Современные требования к инженерам горного профиля включают в себя владение навыками алгоритмизации научных, инженерных и экологических задач по профилю своей деятельности и умение реализовать разработанные алгоритмы с помощью персонального компьютера.

Курсовая работа является самостоятельной программной разработкой студента на этапе завершения изучения курса информатики. Основная цель выполнения курсовой работы – закрепление и углубление знаний, полученных студентом в процессе изучения дисциплины «Информатика», применение их для решения конкретных прикладных задач с использованием возможностей современного программного обеспечения.

Курсовая работа выполняется на основе знаний и навыков, полученных студентами в ходе предшествующего обучения. Предусматривается самостоятельная работа студентов со специальной литературой, поиск информации в сети Internet.

Данные методические указания составлены для студентов дневной и заочной форм обучения. В методических указаниях излагаются сведения, касающиеся организационных и методических вопросов, содержания и объема курсовой работы, определяется круг решаемых студентом задач и требования к оформлению пояснительной записки к курсовой работе, а также порядок защиты работы.

1. Состав и объем курсовой работы

Курсовая работа по дисциплине «Информатика» состоит из программного продукта, обеспечивающего решение указанной в задании на проектирование прикладной задачи, и пояснительной записки.

Студенты второго курса, выполняющие данную курсовую работу, еще не изучали специальные дисциплины. Поэтому рекомендуемая тематика курсовых работ – решение прикладных математических задач (преимущественно численными методами) с использованием алгоритмических языков Turbo Pascal for Windows или Turbo Pascal 7.0 (Borland Pascal).

Другая возможная тематика курсовых работ – задачи анализа данных и поиска оптимального решения, такие как задачи оптимального использования ресурсов, выбора оптимальных технологий, выбора оптимальной смеси, раскрытия материалов, распределения производственной программы, выбора маршрута и т.п. с использованием возможностей табличного процессора MS Excel.

Такой подход к тематике курсовых работ позволяет подготовить студентов к успешному изучению в последующем таких дисциплин, как «Математические методы и модели», «Моделирование и анализ в горном деле».

Руководитель курсового проектирования по дисциплине может использовать также любую другую тематику курсовых работ, соответствующую дисциплинам учебного плана специальностей горного профиля.

Программа представляется на защиту курсовой работы на дискете или компакт-диске (CD).

Пояснительная записка к курсовой работе выполняется в виде текстового документа, реализованного средствами текстового процессора MS Word и распечатанного на стандартной писчей бумаге формата А4. Для подготовки таблиц, графиков и иллюстраций пояснительной записки возможно использование табличного процессора MS Excel.

Объем пояснительной записки составляет 20-25 страниц.

2. Последовательность выполнения курсовой работы

На первом этапе составляется и выполняется программа для решения конкретной прикладной задачи с помощью персонального компьютера (ПК). Первый этап включает следующие шаги:

- уяснение студентом сути задания на проектирование, постановка задачи;
- математическая формулировка задачи;
- выбор численного метода решения задачи;
- алгоритмизация задачи;
- задание имен (идентификаторов) констант и переменных программы, их соответствия обозначениям констант и переменных в математических формулах (алгоритме);
- построение блок-схемы алгоритма;
- составление программы на алгоритмическом языке;
- отладка программы;
- выполнение программы, вывод результатов;
- анализ полученных результатов.

На втором этапе оформляется пояснительная записка к курсовой работе. Пояснительная записка должна отображать результаты выполнения студентом всех шагов первого этапа (за исключением процесса отладки программы) при решении указанной в задании задачи.

Если метод решения задачи определен в задании, выбор численного метода решения не производится.

3. Содержание пояснительной записки

Пояснительная записка задачи, решаемой средствами алгоритмического языка Pascal, должна содержать в указанной последовательности:

- 1) титульный лист;
- 2) задание на курсовую работу;
- 3) содержание с указанием страниц;
- 4) введение;
- 5) постановку задачи;
- 6) математическую формулировку задачи;
- 7) обоснование выбранного метода решения задачи (если методы не определены в задании);
- 8) перечень идентификаторов программы с указанием их соответствия обозначениям, используемым в математических формулах;
- 9) краткое описание алгоритма решения задачи;
- 10) блок-схему алгоритма головной программы и подпрограмм, если последние используются при составлении программы;
- 11) текст программы;
- 12) результаты выполнения программы;
- 13) анализ результатов;
- 14) инструкцию по работе с программой;
- 15) заключение;
- 16) список использованных источников.

Текст пояснительной записки делят на разделы, и, при необходимости, на подразделы и пункты, используя арабские цифры с последующей точкой (1., 1.1., 1.1.1. и т.д.) в соответствии с требованиями стандартов и правил оформления таких текстовых документов (ГОСТ 2.105-95 и ГОСТ 2.106-96). Каждый раздел следует начинать с новой страницы.

Разделы «Введение», «Заключение» и «Список использованных источников» не нумеруются.

Все формулы пояснительной записки нумеруются арабскими цифрами в круглых скобках (справа на уровне формулы) сквозной нумерацией по всей записке.

Указания по содержанию разделов типовой пояснительной записки приведены ниже.

Введение

Раздел «Введение» должен вводить читателя пояснительной записки в суть изложенного в ней. Во введении кратко формулируется цель работы, показывается, какая задача решается, какие особенности присущи такому классу задач, какие методы решения используются, какие результаты получаются и в каком виде представляются. При необходимости указывается новизна и актуальность способа алгоритмизации или решения задачи.

Следует избегать включения во введение общеизвестных утверждений, например, о преимуществах персональных компьютеров, достоинствах и недостатках используемого алгоритмического языка и т.д.

Раздел 1. Постановка задачи

В этом разделе следует показать, как студент уяснил содержание задания на курсовую работу и какой порядок работы программы им принят.

Необходимо продумать и указать:

- 1) будут ли использоваться и для каких целей подпрограммы-процедуры или подпрограммы-функции при реализации алгоритма решения задачи;
- 2) как будет реализован выбор метода решения задачи пользователем – с использованием меню или путем ввода определенных значений в ответ на запрос программы;
- 3) какие исходные данные будут вводиться для каждого метода, каким образом – в виде перечня констант либо элементов массива, из файла с исходными данными либо с клавиатуры по запросу программы;
- 4) предусматривается ли просмотр промежуточных значений переменных, их корректировка по ходу выполнения программы;
- 5) какие результаты выполнения программы будут выводиться и каким образом – на экран пользователя или в файл результатов, в виде отдельных значений, таблицы или графика, в отдельном окне или совместно с введенными исходными данными;
- 6) будут ли выводиться сообщения пользователю по ходу выполнения программы и в каком виде.

Раздел 2. Математическая формулировка задачи

В разделе «Математическая формулировка задачи» все существующие зависимости между константами и переменными задачи с учетом налагаемых на их значения ограничений должны быть выражены при помощи формул, неравенств и уравнений, которые соответствуют используемому методу нахождения решения и будут описаны средствами алгоритмического языка в программе.

Если метод или методы решения задачи в задании не указаны, то в этом разделе рассматриваются также возможные методы решения и выбирается тот из них, который наиболее эффективно обеспечивает выполнение требования поставленной задачи.

Содержание этого раздела показывает, насколько глубоко студент изучил теоретические основы используемого метода решения задачи.

{Пример математической формулировки задачи решения дифференциального уравнения первого порядка приведен ниже.}

Пусть по заданию требуется найти решение дифференциального уравнения первого порядка $y'=f(x,y)$ на отрезке $[x_0, x_0+H]$ при начальных условиях $y(x_0)=y_0$ методом Рунге-Кутты.

Численное решение таких дифференциальных уравнений заключается [1] в том, что значение искомой функции y вычисляется в каждой из точек x_i , которые разбивают заданный отрезок $[x_0, x_0+H]$ на n частей $(x_0, x_1, \dots, x_i, x_{i+1}, \dots, x_n=x_0+H)$ с шагом

$$h=H/n; \quad (1)$$

При этом переход от координат уже известной точки (x_i, y_i) к координатам точки (x_{i+1}, y_{i+1}) осуществляется по формулам

$$x_{i+1}=x_i+h, \quad (2)$$

$$y_{i+1}=y_i+\Delta y. \quad (3)$$

Различные численные методы решения [1-3] дифференциальных уравнений первого порядка (методы Эйлера, Эйлера-Коши, Рунге-Кутты) отличаются друг от друга способом вычисления величины Δy . Так как по условию задачи метод решения задан, используем для определения Δy соотношения, используемые [1] в методе Рунге-Кутты с постоянным шагом h :

$$\Delta y = \frac{1}{6} (\kappa_1 + 2\kappa_2 + 2\kappa_3 + \kappa_4), \quad (4)$$

где

$$\kappa_1 = h \cdot f(x_i, y_i), \quad (5)$$

$$\kappa_2 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{\kappa_1}{2}\right), \quad (6)$$

$$\kappa_3 = h \cdot f\left(x_i + \frac{h}{2}, y_i + \frac{\kappa_2}{2}\right), \quad (7)$$

$$\kappa_4 = h \cdot f(x_i+h, y_i+\kappa_3). \quad (8)$$

Раздел 3. Алгоритмизация задачи

В данном разделе приводится краткое описание алгоритма решения задачи в виде последовательности арифметических, логических и других действий, которые необходимо проделать для реализации принятого метода решения.

Ссылки на номера формул из раздела 2 «Математическая формулировка задачи» обязательны.

Пример алгоритмизации задачи, решения дифференциального уравнения первого порядка $y'=f(x,y)$, математическая формулировка которой приведена в примере раздела 2:

1. Ввести длину H заданного отрезка и n – количество интервалов разбиения отрезка.
2. Вычислить значение $h=H/n$ по формуле (1).
3. Ввести начальные условия x_0 и $y(x_0)$, обозначив их x и y .
4. Организовать цикл по i от 1 до n .
5. Вычислить κ_1 по формуле (5), κ_2 по формуле (6), κ_3 по формуле (7), κ_4 по формуле (8).

6. Вычислить Δy по формуле (4).
7. Вычислить новое значение $y = y + \Delta y$ по формуле (3).
8. Вычислить новое значение $x = x + h$, по формуле (2).
9. Вывести текущие значения x и y на дисплей.
10. Конец цикла по i .

При использовании данного алгоритма требуется периодически вычислять значение функции $f(x,y)$ для разных значений её аргументов. Для программы решения дифференциального уравнения с конкретной функцией $f(x,y)$ в правой части целесообразно использовать для вычисления её значений подпрограмму-функцию. При изменении вида функции $f(x,y)$ достаточно будет откорректировать выражение для вычисления $f(x,y)$ в подпрограмме-функции.

Раздел 4. Идентификаторы программы

В этом разделе устанавливается соответствие обозначений формальных параметров формул, неравенств и уравнений, используемых при математической формулировке задачи, обозначающим их идентификаторам программы. Следует использовать по возможности идентичные идентификаторы, т.е. « x » в формуле – « x » в программе, хотя разработчик программы имеет право обозначить параметр формулы x идентификатором β или γ в программе.

Соответствие параметров формул и идентификаторов программы следует представлять в виде таблицы (пример для задачи решения дифференциального уравнения первого порядка $y' = f(x,y)$ – в табл.1):

Таблица 1

Обозначение параметров		Смысл параметров
В формулах	В программе	
x_0	X_n	Начальное значение вещественной переменной x (левая граница отрезка интегрирования)
x	X	Текущее значение переменной x
$y(x_0)$	Y	Значение функции $y(x)$ для текущего x
Δy	dy	Изменение функции $y(x)$ при переходе к следующей точке отрезка
x_i	$x [i]$	Элемент массива вещественных чисел $x(i)$
i	i	Целочисленная переменная цикла с параметром

При использовании подпрограмм аналогичные таблицы составляются для локальных констант и переменных каждой подпрограммы в соответствующем подразделе данного раздела.

Раздел 5. Блок-схема алгоритма

Блок-схема – это графическое представление посредством условных графических символов алгоритма решения задачи. Блок-схема алгоритма представляет собой последовательность соединенных друг с другом блоков, наглядно показывающая очередность выполнения соответствующих блокам действий и связи между ними. Внутри каждого блока записывается краткое содержание выполняемых операций, а также может быть записан его номер, используемый для указания связей между блоками, когда блок-схема алгоритма занимает несколько страниц.

Условные графические символы, используемые при построении блок-схем алгоритма, определены ГОСТ 19.701-90.

При использовании в программе подпрограмм в соответствующих подразделах данного раздела приводятся отдельно блок-схемы алгоритма головной программы и подпрограмм.

При описании содержания выполняемых в блоках операций используются идентификаторы констант и переменных, принятые в программе (подпрограмме).

Примеры оформления блок-схем алгоритма головной программы и подпрограммы приведены в приложении.

Раздел 6. Текст программы

Программа решения задачи составляется по разработанной блок-схеме алгоритма на алгоритмическом языке Turbo Pascal for Windows или Turbo Pascal 7.0 (Borland Pascal). В пояснительной записке приводится распечатка исходного текста программы как последовательности операторов (команд, инициирующих выполнение предусмотренных алгоритмом действий).

Примечание. Для распечатки текста исходной программы на алгоритмическом языке Turbo Pascal 7.0 из текстового процессора MS Word, необходимо:

- 1) Выполнить в окне MS Word команду Сервис/Параметры;
- 2) Активизировать щелчком левой кнопки мыши вкладку «Общие»;
- 3) В открывшемся окне «Параметры» установить щелчком левой кнопки мыши флажок режима «Подтверждать преобразование при открытии»;
- 4) Выбрать файл Паскаль-программы и открыть его в среде MS Word;
- 5) В ответ на запрос в окне «Преобразование файла» выбрать «Текст DOS» и нажать ОК.

Раздел 7. Результаты выполнения программы

В данном разделе помещаются распечатки копий экрана с меню, с окнами ввода-вывода результатов, содержимого внешних файлов на диске с результатами счета, таблиц и графиков, отражающих результаты выполнения программы.

Распечатки копий экрана, окон ввода-вывода и графиков представляются как иллюстрации. Все иллюстрации нумеруются арабскими цифрами сквозной нумерацией в пределах всей пояснительной записки (Рисунок 1, Рисунок 2 и т.д.).

До распечатки иллюстрации в тексте данного раздела должна быть соответствующая ссылка на него, например: «Вид окна ввода исходных данных представлен на рисунке 2».

Под иллюстрацией, помимо слова «Рисунок» и номера, может приводиться её наименование, например: «Рисунок 2. Окно ввода исходных данных».

Примеры распечатки результатов работы программы представлены в приложении 2.

Раздел 8. Анализ результатов

Здесь проверяется корректность полученных результатов путем сравнения их с результатами аналитического решения (если оно существует) либо путем подстановки вычисленных значений корней в уравнения либо построением графиков анализируемых функций (в том числе и средствами MS Excel) и т.д. в зависимости от вида решаемой задачи. Оценивается также стабильность работы программы, точность и скорость вычислений при использовании различных методов решения.

В этом разделе приводится краткое описание последовательности действий, которую должен выполнить незнакомый ранее с программой пользователь, чтобы запустить ее, ввести исходные данные и вывести результаты её выполнения.

Заключение

В разделе «Заключение» в краткой форме указывается соответствие созданной программы поставленной в задании задаче и сформулированным в разделе 1 пояснительной записки принципам функционирования программы. Здесь отмечаются также достоинства и недостатки использованных методов решения задачи, возможные варианты использования программы.

Список использованных источников

В этом разделе приводится список литературных и других (веб-сайты, CD и DVD-диски) источников, которые были использованы студентом при выполнении курсовой работы.

Источники в списке располагаются в порядке следования ссылок на них в тексте пояснительной записки.

Правила оформления списка источников определены ГОСТ 7.1-84.

4. Организация консультирования

Вычислительный центр факультета природных ресурсов и экологии выделяет машинное время для учебных групп студентов дневного отделения, выполняющих курсовую работу. В это же время руководители курсовой работы проводят консультации студентов.

Студенты- заочники получают консультации по выполнению курсовой работы на кафедре горных машин в установленные субботние дни.

5. Защита курсовой работы

После завершения курсовой работы, разработанные студентами программы, помещаются в специально выделенную для учебной группы папку, в которой для каждого студента создается персональная папка. Имя персональной папки - фамилия и инициалы студента. В персональной папке хранится текст исходной Паскаль-программы и, желательнее, текстовый файл MS Word, содержащий пояснительную записку. Результат компиляции исходной Паскаль-программы (exe-файл) и, в случае использования программой, файлы с исходными данными или результатами выполнения также помещаются в персональную папку студента.

Пояснительная записка сдается на проверку руководителю студентами дневной формы обучения - не позднее, чем за неделю до установленного срока защиты курсовой работы, студентами заочной формы обучения - не позднее, чем за две недели до начала сессии. Подписанный руководителем титульный лист дает студенту право на защиту курсовой работы.

Защита проводится исполнителем курсовой работы перед комиссией, назначаемой кафедрой. Руководитель защищаемой курсовой работы обязательно входит в состав комиссии по защите.

Процедура защиты курсовой работы включает:


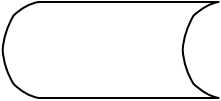
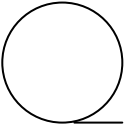
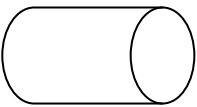





- 1) доклад о проделанной работе;
- 2) демонстрацию работы программы;
- 3) ответы на вопросы членов комиссии по защите;
- 4) обсуждение доклада и результатов работы программы, оценку работы.


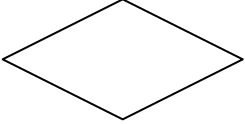
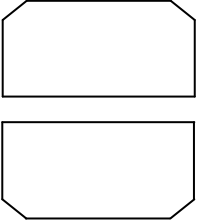
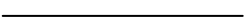
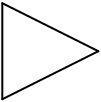
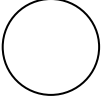

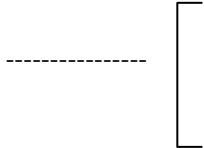
Рекомендуемая литература

1. Вирт Н. Алгоритмы и структуры данных/Пер. с англ.- М.: Мир, 1989.- 360 с.
2. Боглаев Ю.П. Вычислительная математика и программирование. – М.: Высш.шк., 1990.- 544 с.
3. Котов В.М., Мельников О.И. Информатика. Методы алгоритмизации. Минск: Народная асвета, 2000.- 221 с.
4. Пестриков В.М., Маслобоев К.Н. Turbo Pascal 7.0. Изучаем на примерах.- М.: Наука и техника, 2004.- 365 с.
5. Сухарев М. Turbo Pascal 7.0. Теория и практика программирования. М.: Наука и техника, 2004.- 576 с.
6. Культин Н. Turbo Pascal в задачах и примерах. - СПб.: ВHV – Санкт-Петербург, 2003.- 256 с.
7. Климова Л.Н. Pascal 7.0. Издательство КУДИЦ- образ, 2002.- 496 с.
8. Мартынюк Т. Турбо-Паскаль 7.0. Самоучитель. – СПб.: Питер, 2002. – 416 с.
9. Глушаков С.В., Зорянский В.Н., Хоменко С.Н. Turbo Pascal 7.0. –Харьков: Фолио. 2002.- 501 с.
10. Фаронов В.В. Turbo Pascal 7.0. Начальный курс: Учеб. пособие. - М.: Нолидж, 2000.- 575 с.
11. Фаронов В.В. Turbo Pascal 7.0. Начальный курс (7-е издание). - СПб: Питер, 2001.- 576 с.
12. Епанешников А.М., Епанешников В.А. Программирование в среде Turbo Pascal 7.0. – М.: ДИАЛОГ-МИФИ, 1998.- 367 с.
13. Марченко А.И. Программирование в среде Borland Pascal 7.0. – Киев: БЕК:ЮНИОР, 1996.- 476 с.
14. Справочник по процедурам и функциям Borland Pascal With Objects 7.0 – Киев: Диалектика, 1993.- 272 с.
15. Сергиевский М.В., Шалашов А.В. Турбо Паскаль 7.0.: язык, среда программирования. - М.: Машиностроение, 1994.- 254 с.
16. Зуев Е.А. Язык программирования Turbo Pascal.- М.: Унитех, 1992.- 289 с.
17. Бородич Ю.С. Разработка программных систем на языке Паскаль: Справ. пособие. – Мн.: Выш.шк.. 1992.- 143 с.
18. Паскаль для персональных компьютеров: Справ. Пособие/Ю.С. Бородич, А.Н. Вальвачев, А.И. Кузьмич.- Мн.: Выш. шк.: БФ ГИТМП «Ника», 1991.- 365 с.
19. Мирзохи С.В. TURBO PASCAL и объективно-ориентированное программирование. – М.: Финансы и статистика, 1992.- 192 с.
20. Паскаль: [Учебн. пособие] / В.С. Новичков и др. - М.: Высш.шк., 1990.-222 с.
21. Перминов О.Н. Программирование на языке Паскаль.- М.: Радио и связь, 1988.- 224 с.
22. Введение в язык Паскаль: Учебн. пособие/В.А. Абрамов и др. – М.: Наука, 1988.- 318 с.
23. Абрамов С.А. Начала программирования на языке Паскаль.- М.: Наука. 1987.- 110 с.
24. Сборник упражнений по языку Паскаль: Учебное пособие/В.Н. Пильщиков. - М.: Высш. шк., 1990.- 223 с.
25. Ракитин В.И., Первушин Е.В. Практическое руководство по методам вычислений с приложением программ для персонального компьютера: Учебное пособие. - М.: Высш. шк., 1998.- 283 с.

Приложения

Наиболее часто используемые в блок-схемах алгоритмов условные графические символы

Обозначение по ГОСТ 19.701-90	Наименование символа	Действие
Символы данных		
	Данные	Отображает данные вообще, носитель данных не определен (любой).
	Запоминаемые данные	Отображает хранимые данные в виде, пригодном для обработки, носитель данных не определен (любой).
	Запоминающее устройство с последовательной выборкой	Отображает данные, хранящиеся в запоминающем устройстве с последовательным доступом (например, магнитная лента).
	Запоминающее устройство с прямым доступом	Отображает данные, хранящиеся в запоминающем устройстве с прямым доступом (например, магнитный диск).
	Документ	Отображает данные, представленные на носителе в удобочитаемой форме (например, на бумаге).
	Ручной ввод	Отображает данные, вводимые вручную во время обработки с устройств любого типа (например, с клавиатуры).
	Карта	Отображает данные, представленные на носителе в виде карты (например, магнитной).
	Дисплей	Отображает данные, представленные в удобочитаемой форме на носителе в виде отображающего устройства (например, на дисплее).
Символы процесса		
	Процесс	Отображает функцию обработки данных любого вида (выполнение определенной операции или группы операций, приводящее к изменению значения, формы или

		размещения информации).
	Предопределенный процесс	Отображает предопределенный процесс, состоящий из одной или нескольких операций или шагов программы, которые определены в другом месте (например, в подпрограмме или модуле).
	Решение	Отображает решение или функцию переключательного типа, имеющую только один вход и ряд альтернативных выходов, один и только один из которых может быть активизирован после вычисления условий, определенных (указанных) внутри этого символа.
	Граница цикла	Символы отображают начало и конец цикла
Символы линий		
	Линия	Отображает поток данных или управления. Для указания направления потока могут быть добавлены стрелки-указатели.
	Передача управления	Отображает непосредственную передачу управления от одного процесса к другому. Тип передачи управления указывается внутри символа (например, запрос, вызов, событие).
Специальные символы		
	Соединитель	Отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии и продолжения ее в другом месте.
	Терминатор	Отображает выход в начало или конец схемы программы, внешнее использование и источник или пункт назначения данных).
	Комментарий	Используется для добавления описательных комментариев или пояснительных записей к блоку.

Пример оформления пояснительной записки к курсовой работе

Министерство образования Республики Беларусь
Белорусский национальный технический университет

Кафедра «Горные машины»

КУРСОВАЯ РАБОТА
по дисциплине «Информатика»

**Программа вычисления определенного интеграла
методами Симпсона и Боде**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Руководитель

Петренко С.М.

Исполнитель

студент группы 302812
Сидоров И. П.

2004

Задание на курсовое проектирование

Разработать программу вычисления определенного интеграла

$$I = \int_a^b \sqrt{3x+2} dx$$

методом Симпсона или методом Боде. Метод вычисления выбирает пользователь после запуска программы на выполнение. Пределы интегрирования и количество разбиений интервала интегрирования вводятся пользователем по запросу программы в процессе реализации выбранного метода.

Примечание. В реальной пояснительной записке на месте этой страницы размещается утвержденный заведующим кафедрой бланк задания на курсовую работу, подписанный руководителем и исполнителем.

Содержание

Введение.....	4
1. Постановка задачи.....	5
2. Математическая формулировка задачи.....	7
3. Алгоритмизация задачи	10
4. Идентификаторы программы.....	12
5. Блок – схема алгоритма.....	13
6. Текст исходной программы.....	16
7. Результаты выполнения программы.....	22
8. Анализ результатов.....	24
9. Инструкция по работе с программой.....	25
Заключение.....	26
Список использованных источников.....	27

Введение

В курсовой работе в соответствии с заданием на проектирование решается задача разработки программы вычисления определенных интегралов численными методами. Численное интегрирование используется, когда аналитическое нахождение значений первообразной функции сложно или вообще невозможно.

В данной пояснительной записке проводится описание последовательности шагов по составлению программы на алгоритмическом языке Turbo Pascal 7.0 и результаты применения этой программы для вычисления конкретного определенного интеграла численными методами Симпсона и Боде. Рассматриваются вопросы математической формулировки и алгоритмизации задачи, разработки блок-схемы алгоритма её решения, составления исходной Pascal-программы и реализации вычислений по составленной программе.

Выбор метода вычисления, обращение к справке по программе и выход из программы обеспечивается с помощью специального меню. Ввод исходных данных и вывод результатов вычисления интеграла выполняется в отдельном для каждого метода вычислений окне.

В пояснительной записки приводится также анализ точности вычислений определенного интеграла использованными методами.

1. Постановка задачи

Ставится задача составить программу вычисления интегралов вида

$$I = \int_a^b f(x)dx, \quad (1)$$

где a и b – нижний и верхний пределы интегрирования, $f(x)$ – непрерывная подынтегральная функция на отрезке $[a, b]$. Такой определенный интеграл можно трактовать как площадь фигуры (рисунок 1), ограниченной ординатами a и b , осью абсцисс x и графиком подынтегральной функции $f(x)$.

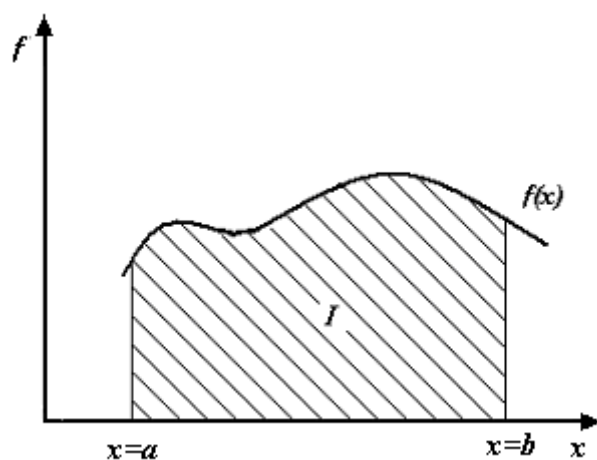


Рисунок 1. К вычислению определенного интеграла

Обыкновенный определенный интеграл, у которого известна его первообразная $F(x)$, вычисляется по формуле Ньютона-Лейбница

$$I = F(b) - F(a). \quad (2)$$

Численное интегрирование применяется, если нахождение $F(x)$ сложно или невозможно. Оно заключается в интерполяции $f(x)$ на отрезке $[a, b]$ подходящим полиномом, для которого известна первообразная $F(x)$. Обычно отрезок $[a, b]$ разбивается на N частей, к каждой из которых применяется соответствующая простая формула.

Таким образом, программа должна обеспечивать возможность:

- быструю смену подынтегральной функции $f(x)$ путем замены аналитического выражения ее правой части в подпрограмме – функции;
- выбора пользователем численного метода вычисления интеграла;
- ввода с клавиатуры значений пределов интегрирования a и b и количества интегралов разбиения n ;
- вывода результатов вычисления на дисплей в удобном для восприятия виде с отображением названия выбранного численного метода, значений пределов интегрирования и принятого числа интегралов разбиения.

Кроме того, целесообразно предоставить пользователю возможность получить краткую справку по программе, а также давать подсказки по ходу работы с программой.

В результате сформулируем следующую задачу по созданию программы:

- значения подынтегральной функции $f(x)$ должны вычисляться в подпрограмме – функции;
- программа после загрузки должна выводить на дисплей исходное окно-заставку, в которой отображаются общие сведения о статусе программы и её авторах;
- после выполнения указанной в строке подсказки процедуры перехода должно выводиться вертикальное меню с пунктами: «Справка», «Метод Симпсона», «Метод Бодде» и «Выход»;
- при выборе в меню пункта «Справка» должна выводиться краткая справка о назначении программы и порядке работы с ней;
- после выбора в меню варианта численного метода должно открываться отдельное окно, в котором будут выводиться запросы программы о значениях пределов интегрирования, числе интегралов разбиения и в этом же окне после ввода исходных значений будет выводиться результат вычисления интеграла выбранным методом;
- при выборе пункта меню «Выход» программы должна завершать работу.

2. Математическая формулировка задачи

Сущность большинства численных методов вычисления определенных интегралов состоит в замене подынтегральной функции $f(x)$ аппроксимирующей её функцией $\varphi(x)$, для которой можно легко записать первообразную в элементарных функциях, т.е.

$$\int_a^b f(x)dx = \int_a^b \varphi(x)dx + R = S + R, \quad (3)$$

где S – приближенное значение интеграла, R – погрешность вычисления интеграла.

В данной курсовой работе требуется вычислять интеграл численными методами Симпсона и Бode.

В методе Симпсона [1,2] подынтегральную функцию $f(x)$ заменяют интерполяционным полиномом второй степени $P_2(x)$ – параболой, проходящей через узлы x_0, x_1, x_2 (рисунок 2) так, чтобы

$$\int_a^b f(x)dx = \int_a^b P_2(x)dx + R. \quad (4)$$

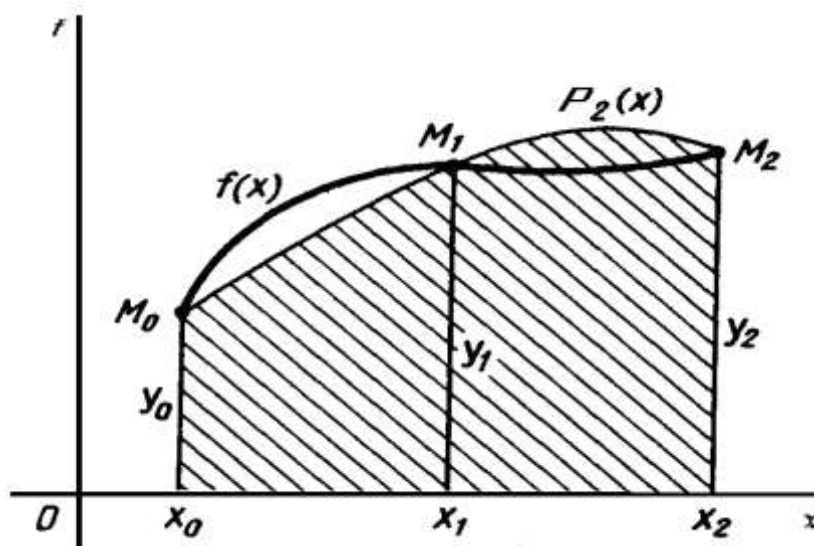


Рисунок 2. К вычислению определенного интеграла методом Симпсона.

Для записи полинома $P_2(x)$ используют интерполяционную формулу Ньютона для трех узлов, так как достаточная для практики точность обеспечивается [2] при использовании многочлена третьей степени:

$$P_2(x) = f_0 + f_{01}(x - x_0) + f_{012}(x - x_0)(x - x_1), \quad (5)$$

где $f_{01} = \frac{f_0 - f_1}{x_0 - x_1} = \frac{f_1 - f_0}{h}$, $f_{012} = \frac{f_{01} - f_{02}}{x_1 - x_2} = \frac{f_0 - 2f_1 + f_2}{2h^2}$ - разделенные разности;

$h = (b-a)/n$ – расстояние между узлами (шаг приращения x);

n – число интервалов интегрирования (разбиения отрезка $[a, b]$).

При вводе новой переменной $z=x-x_0$, $x=z+x_0$ и полином принимает вид

$$P_2(z) = f_0 + (f_{01} - f_{012}h)z + f_{012}z^2, \quad (6)$$

и вычисление интеграла от полинома выражается простыми функциями

$$\int_{x_0}^{x_2} P_2(x) dx = \int_0^{2h} P_2(z) dz = (f_0 + 4f_1 + f_2)h/6, \quad (7)$$

где $f_0 = f(x)$ - значение подынтегральной функции при текущем значении x ; (8)

$$f_1 = 4f(x + h/2); \quad (9)$$

$$f_2 = 2f(x + h). \quad (10)$$

Соотношение (7) называют квадратурной формулой Симпсона или формулой парабол.

При интегрировании сложных функций интервал интегрирования разбивается на несколько равных промежутков n . В результате площадь всей криволинейной трапеции заменяется суммой площадей n маленьких криволинейных трапеций. Вычисление площади каждой маленькой трапеции осуществляется по формуле Симпсона.

В методе Бode [1,2] подынтегральную функцию $f(x)$ заменяют интерполяционным полиномом четвертой степени, проходящим через узлы x_0, x_1, x_2, x_3, x_4 (рисунок 3).

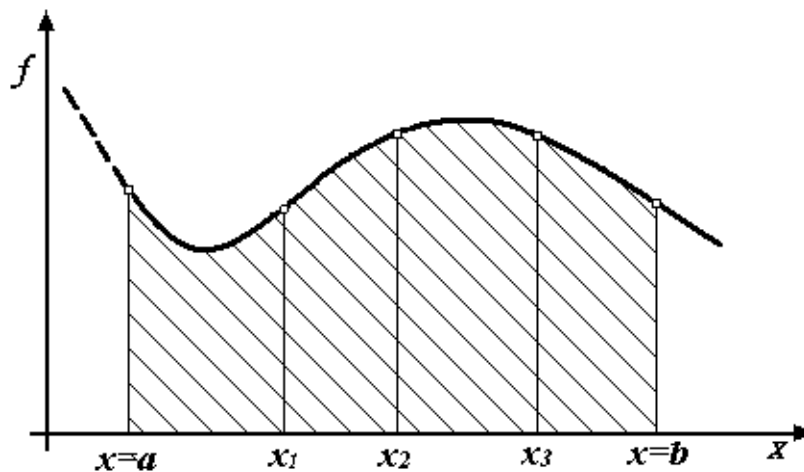


Рисунок 3. К вычислению определенного интеграла методом Боде.

Вычисление интеграла производится по формуле [2]

$$I = \int_{x_0}^{x_4} f(x) dx = \frac{2h}{45} (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4), \quad (11)$$

применяемой к каждому интервалу разбиения отрезка $[a, b]$,

где $f_0 = f(x)$ - значение подынтегральной функции для текущего интервала; (12)

$$f_1 = f(x + h/4); \quad (13)$$

$$f_2 = f(x + h/2); \quad (14)$$

$$f_3 = f(x + 3h/4); \quad (15)$$

$$f_4 = f(x + h). \quad (16)$$

3. Алгоритмизация задачи

В соответствии с постановленной в разделе 2 задачей целесообразно реализовать алгоритм, использующий обращение к соответствующим подпрограммам из головной программы.

Алгоритм работы головной программы следующий:

1. Скрыть курсор с использованием подпрограммы - процедуры скрывания курсора и вывести в специальном окне заставку программы, содержащую сведения о назначении программы, исполнителе и руководителе курсовой работы, а также подсказку для пользователя о последующих действиях, с использованием подпрограммы - процедуры заставки.
2. Запустить подпрограмму-процедуру вертикального меню при нажатии любой клавиши с использованием подпрограмм-процедур построения окна, вывода рамки окна и скрывания курсора.
3. Запустить подпрограмму-процедуру справки и вывести в специальном окне справочные сведения о работе с программой при выборе пункта меню «Справка» с использованием подпрограмм-процедур построения окна, вывода рамки окна и скрывания курсора, а также строки-подсказки о возврате в меню.
4. Запустить подпрограмму-процедуру вычисления интеграла методом Симпсона при выборе пункта меню «Метод Симпсона» с использованием подпрограмм-процедур построения окна, вывода рамки окна и включения курсора, а также строки-подсказки о возврате в меню.
5. Запустить подпрограмму-процедуру вычисления интеграла методом Боде при выборе пункта меню «Метод Боде» с использованием подпрограмм-процедур построения окна, вывода рамки окна и включения курсора, а также строки-подсказки о возврате в меню.
6. Завершить работу программы при выборе пункта меню «Выход».

Алгоритм вычисления интеграла методом Симпсона в подпрограмме-процедуре включает следующие шаги:

1. Создать окно для ввода исходных данных и вывода результатов вычисления по методу Симпсона.
2. Восстановить отображение курсора нормального размера соответствующей подпрограммой - процедурой.
3. Ввести нижний a и верхний b пределы интегрирования как границы отрезка интегрирования и количество n интервалов его разбиения.

4. Вывести в окне запрос о правильности ввода значений a , b и n .
5. Выполнить анализ кода нажатой в ответ на запрос клавиши: при нажатии “Y” перейти к вычислениям, при нажатии любой другой клавиши перейти к пункту 3 алгоритма.
6. Вычислить значение шага $h=(b-a)/n$ по формуле (6).
7. Присвоить первоначальному значению интеграла s нуль и начальному значению аргумента x_0 значение a .
8. Организовать цикл по i от 1 до n .
9. Вычислить по формулам (8) – (10) значения f_0 , f_1 и f_2 .
10. Вычислить значение $s=s+ f_0 + 4f_1 + f_2$.
11. Вычислить новое значение $x= x+ h$.
12. Закончить цикл по i .
13. Вычислить конечное значение интеграла $s=s/b$ в соответствии с (7).
14. Вывести результаты вычислений в том же окне
15. Вывести в окне запрос о продолжении вычислений с новыми исходными данными.
16. Выполнить анализ кода нажатой в ответ на запрос клавиши: при нажатии “Y” перейти в окно с меню, при нажатии любой другой клавиши перейти в окно с заставкой программы.

Алгоритм вычисления интеграла методом Боде в подпрограмме-процедуре

включает:

1. Создать окно для ввода исходных данных и вывода результатов вычисления по методу Боде.
2. Восстановить отображение курсора нормального размера соответствующей подпрограммой - процедурой.
3. Ввести нижний a и верхний b пределы интегрирования как границы отрезка интегрирования и количество n интервалов его разбиения.
4. Вывести в окне запрос о правильности ввода значений a , b и n .
5. Выполнить анализ кода нажатой в ответ на запрос клавиши: при нажатии “Y” перейти к вычислениям, при нажатии любой другой клавиши перейти к пункту 3 алгоритма.
6. Вычислить значение шага $h=(b-a)/n$ по формуле (6).
7. Присвоить переменной s нуль и начальному значению аргумента x_0 значение a .
8. Организовать цикл по i от 1 до n .
9. Вычислить по формулам (12) – (16) значения f_0 , f_1 , f_2 , f_3 и f_4 .

10. Вычислить значение $s = s + 7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4$.
11. Вычислить новое значение $x = x + h$.
12. Закончить цикл по i .
13. Вычислить конечное значение интеграла $s = s/45$ в соответствии с (11).
14. Вывести результаты вычислений в том же окне.
15. Вывести в окне запрос о продолжении вычислений с новыми исходными данными.
16. Выполнить анализ кода нажатой в ответ на запрос клавиши: при нажатии “Y” перейти в окно с меню, при нажатии любой другой клавиши перейти в окно с заставкой программы.

4. Идентификаторы программы

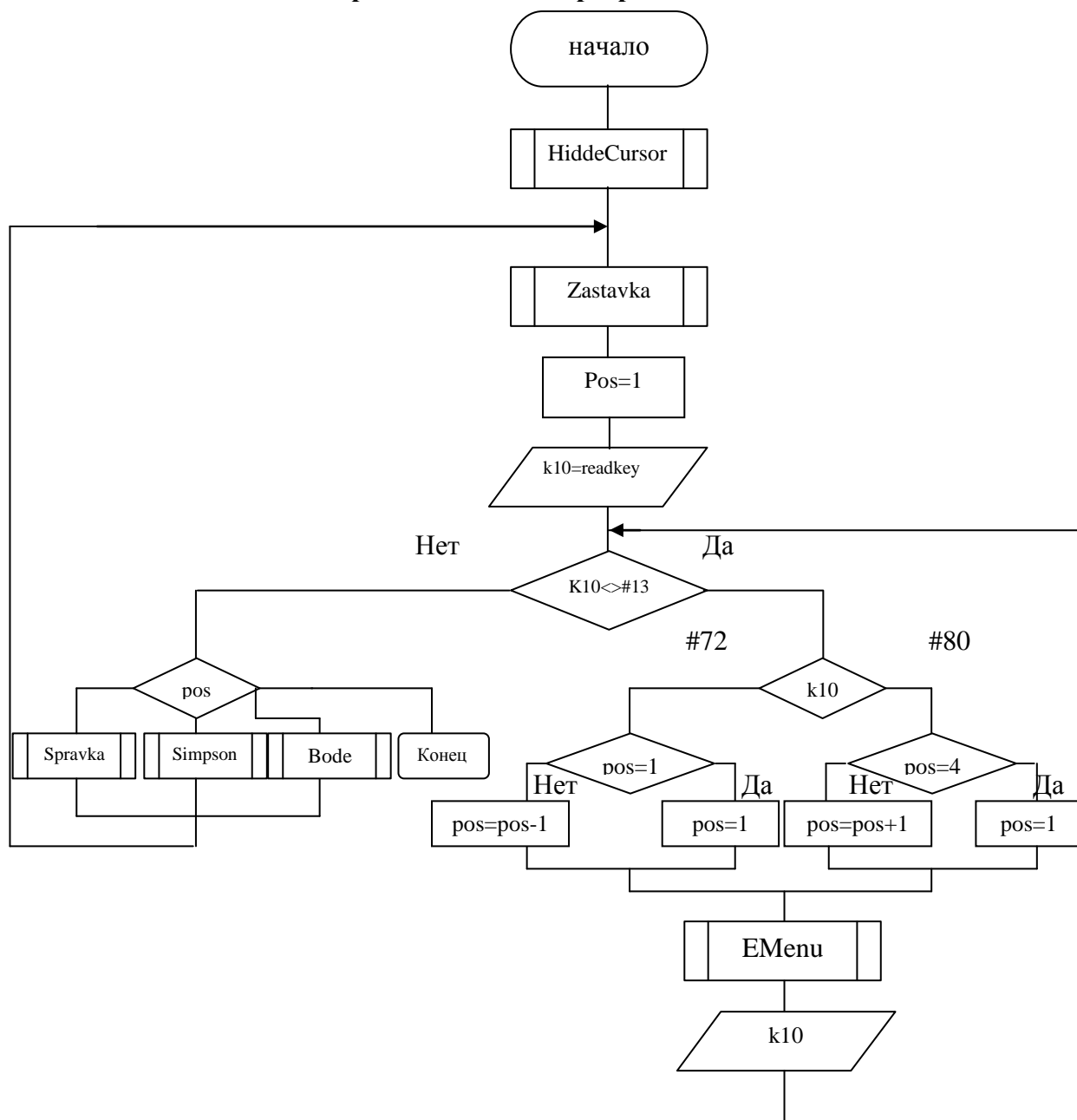
Для указания соответствия обозначений переменных в формулах математической формулировки и их идентификаторов в программе сведем их в таблицу 1:

Таблица 1

Обозначение параметров		Смысл параметра
В формулах	В программе	
A	a	Левая граница отрезка интегрирования (нижний предел интегрирования)
B	b	Правая граница отрезка интегрирования (верхний предел интегрирования)
N	n	Число интервалов разбиения отрезка интегрирования
H	h	Длина интервала разбиения
x	x	Аргумент подынтегральной функции
dx	h ₂ -метод Симпсона h ₄ – метод Боде	Приращение аргумента x
I	S	Значение интеграла

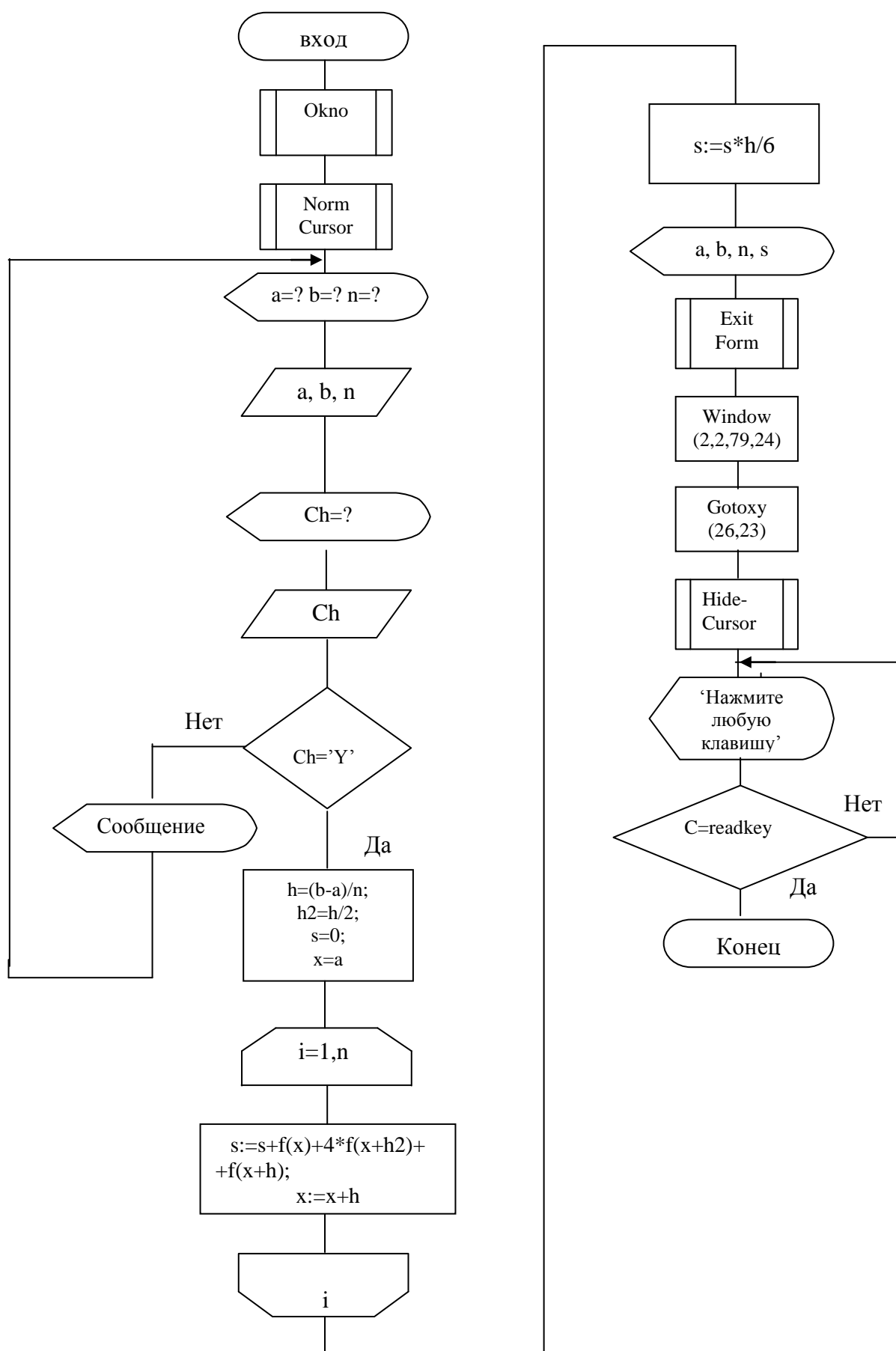
5. Блок-схема алгоритма

5.1. Блок-схема алгоритма головной программы

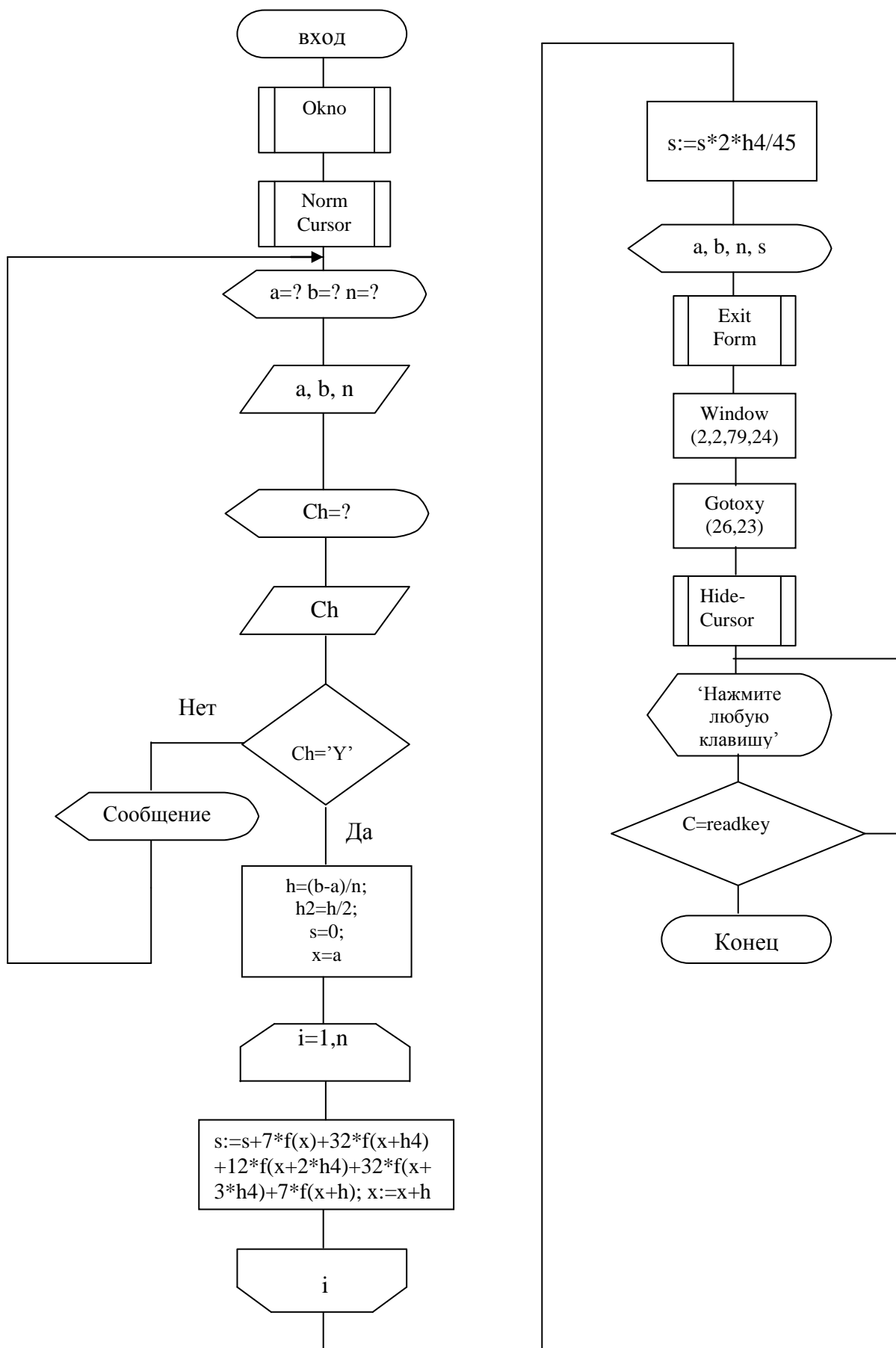


Примечание. Блок-схемы алгоритмов процедур CursorSize, HiddeCursor, NormCursor, Zastavka, Okno, Ramka, Spravka, Emenu, Exitform и подпрограммы-функции f являются очень простыми и в данном методическом пособии не приводятся. В реальной пояснительной записке следует приводить блок-схемы алгоритмов всех подпрограмм, используемых в программе.

5.2 Блок-схема алгоритма подпрограммы–процедуры Simpson



5.3 Блок-схема алгоритма подпрограммы–процедуры Bode



6. Текст исходной программы

```

program Kursovoi_integral;
Uses crt,dos;
Label 11,12,m1;
Type
  mas=Array[1..4] of String[50];
Const
  menu:mas=('Справка          ',
            'Метод Симпсона ',
            'Метод Боде       ',
            'Выход            ');

var
  i,s,pos: Integer;
  c,kod,k10:Char;
  pass: String;
  ch: char;

function f(t:real): real;
begin
  f:=sqrt(3*t+2)
end;
{_____}

{-----Cursorcursor init-----}
Procedure CursorSize(Size: word);
var Regs: Registers;
Begin
  With Regs do
    Begin
      AH:=$01;
      CH:=Hi(size);
      Cl:=Lo(Size);
      intr($10,Regs);
    end;
end;
{-----Cursor init-----}

{-----Cursor OFF-----}

procedure HiddeCursor;
begin
  CursorSize($2000);
end;
{-----Cursor OFF-----}

{-----Cursor ON-----}
Procedure NormCursor;
begin
  CursorSize($0607);
end;
end;

```



```
{-----Cursor ON-----}
```

```
{-----Create of Frame-----}
```

```
procedure ramka(x1,y1,x2,y2: byte);
  const
    a=#201;
    b=#205;
    c=#187;
    d=#186;

    e=#188;
    f=#200;
  Begin
    Textcolor(15);
    Textbackground(0);
    Clrscr;
    gotoxy(x1,y1);
    write(a);
    for i:=x1+1 to x2-1 do write(b);
    write(c);
    for i:=y1+1 to y2-1 do
      Begin
        gotoxy(x1,i);
        write(d);
        gotoxy(x2,i);
        write(d);
      End;
    Gotoxy(x1,y2);
    write(f);
    for i:=x1+1 to x2-1 do write(b);
    write(e);
  End;
```

```
Procedure Okno(x1,y1,x2,y2,Fcolor,Tcolor:byte);
```

```
  Begin
    Clrscr;
    TextMode(3);
    HiddeCursor;
    Ramka(x1,y1,x2,y2);
    Window(x1+1,y1+1,x2-1,y2-1);
    textbackground(Fcolor);
    textcolor(Tcolor);
    Clrscr;
    Gotoxy(1,1);
  End;
```

```
Procedure Zastavka;
```

```
  Begin
```

```
Okno(1,1,79,23,3,15);
  gotoxy(3,1);
```

```

        writeln(' ');

        writeln('
Республики Беларусь');
        writeln('
технический университет');
        gotoxy(1,6);
        writeln('
ния интеграла ');
        writeln;
        writeln;
        writeln;
        writeln;
        writeln('
та ');
        writeln('
форматика" ');
        gotoxy(1,15);
        writeln('
полнитель: Сидоров И.П. ');
        writeln('
гр. 302812');

        writeln('
тель: Петренко С.М. ');
        writeln;
        writeln;
        writeln('
');

        window(0,0,80,25);
        gotoxy(40,23);
        Textcolor(15);
        gotoxy(1,79);
        writeln('
любую клавишу... ');
        repeat until keypressed;
        while keypressed do c:=readkey;
        clrscr;
        End;

        Procedure Spravka;
        var
        ch: char;
        Begin
        Okno(1,1,79,23,3,15);
        Writeln('
К А ');
        Writeln('
вычислять интеграл ');

        Writeln('
I=SQRT(3*x+2)dx методом Симпсона или ме-
тодом Боде для заданных');

```

Министерство образования

Белорусский национальный

Программа вычисле-

Курсовая рабо-

по дисциплине "Ин-

Ис-

Руководи-

Минск 2004

Для продолжения нажмите

С П Р А В

Данная программа позволяет

```

Writeln('          значений пределов интегрирования
А(нижний) и В(верхний).');
Writeln('          После перехода в меню выберите
клавшиами управления курсора');
writeln('          необходимый пункт. ');
Writeln('          При выборе метода Симпсона
или метода Боде введите по запросу ');
Writeln('          программы целочисленные значения
пределов интегрирования и число');
Writeln('          интервалов интегрирования. ');
Writeln('          Для интеграла с другой по-
дынтегральной функцией следует');
Writeln('          записать ее в правой части выра-
жения для вычисления значения функции');
Writeln('          в подпрограмме-функции f в разде-
ле объявлений головной программы. ');

```

```

Window(2,2,79,24);
Textcolor(15);
gotoxy(5,20);
write('          Для возврата в меню нажмите лю-
бую клавишу... ');
ch:=readkey;
End;

```

```

Procedure Emenu;
Begin
  clrscr;
  Okno(15,9,55,17,3,15);
  For i:=1 to 4 do
    begin
      Gotoxy(15,i+1);

      if i=pos then Textbackground(4) else Textback-
ground(3);

      write(Menu[i]);
    end;
  End;
  procedure Exitfrom;
  Begin
    writeln;
    gotoxy(50,26);
    Writeln('          Для продолжения вычисления
нажмите Y, для выхода - N');
    ch:=readkey;
    if ch =#121 then Emenu else Zastavka;
  End;

```

```

Procedure Simpson;
label m1;
Var
  i,n : byte;

```

```

a,b,h,h2,s,x : real;
ch: char;
Begin
  Okno(1,1,79,23,3,15);
  NormCursor;
  Writeln('          Вычисление интеграла мето-
дом Симпсона');
  Writeln;
  Writeln;
  m1:
  Writeln('          Введите пределы интегриро-
вания A и B');
  Write('          Нижний предел A=');
  read(a);
  Write('          Верхний предел B=');
  read(b);
  Write('          Введите число интервалов ин-
тегрирования N=');
  read(n);
  Begin
  writeln;
  Writeln('          Значения введены правильно
(Y/N)?');
  ch:= readkey;
  If ch=#110 then
  begin
  writeln;
  Writeln('          Данные введены невер-
но. Повторяем ввод');
  writeln;
  writeln;
  goto m1;
  end;
  h:=(b-a)/n;
  h2:=h/2;
  s:=0;
  x:=a;
  for i:=1 to n do
  begin
  s:=s+f(x)+4*f(x+h2)+2*f(a+2*h2);
  x:=x+h;
  end;
  s:=s*h/6;
  writeln;
  writeln;

  writeln('          При введенных пределах ин-
тегрирования A=',a:5:2,' B=',b:5:2);

  writeln('          Значение интеграла I=', s:9:5);
  Exitfrom;
  Window(2,2,79,24);
  Gotoxy(26,23);

```

```

    Textcolor(4);{меняю на }
    Hiddecursor;
    Write('Нажмите любую клавишу...');
    ch:=readkey;
End;
end;

Procedure Bode;
label m1;
Var
    i,n : byte;
    a,b,h,h4,s,x : real;
    ch: char;
Begin
    clrscr;
    Okno(1,1,79,23,3,15);
    NormCursor;
    Writeln('                                Вычисление интеграла мето-
дом Бode');
    Writeln;
    Writeln;
m1:
    Writeln('                                Введите пределы интегриро-
вания A и B');
    Write('                                Нижний предел A=');
    read(a);
    Write('                                Верхний предел B=');
    read(b);
    Write('                                Введите число интервалов ин-
тегрирования N=');
    read(n);
    Begin
        writeln;
        Writeln('                                Значения введены правильно
(Y/N)?');
        writeln;
        ch:= readkey;
        If ch=#110 then
            begin
                Writeln('                                Данные введены невер-
но. Повторяем ввод');
                writeln;
                goto m1;
            end;
        h:=(b-a)/n;
        h4:=h/4;
        s:=0;
        x:=a;

        for i:=1 to n do
            begin
                s:=s+7*f(x)+32*f(x+h4)+12*f(x+2*h4)+32*f(x+3*h4)+7*(x+h);
            end;
        end;
    end;
end;

```

```

        x:=x+h;
        end;
        s:=2*s*h4/45;
        writeln('
тегрирования A=',a:5:2,' B=',b:5:2);
        writeln('
I=',s:9:5);
        Gotoxy(26,23);

        Exitfrom;
        Window(2,2,79,24);
        Gotoxy(26,23);
        Textcolor(4);
        Hiddecursor;
        Write('Нажмите любую клавишу...');
        ch:=readkey;
        End;
end;

```

При введенных пределах ин-
Значение интеграла

BEGIN

```

        Clrscr;
        hiddecursor;
        Zastavka;

11:
        pos:=1;
        Emenu;
        k10:=readkey;
        while k10<>#13 do
            begin
                Case k10 of
                    #72: if pos=1 then pos:=4 else pos:=pos-1;
                    #80: if pos=4 then pos:=1 else pos:=pos+1;
                end;
            Emenu;
            k10:=readkey;
        end;
        Case pos of
            1: Spravka;
            2: simpson;
            3: bode;
            4: exit;
        end;
        goto 11;

```

END.

7. Результаты работы программы

После запуска программы в соответствии с поставленной задачей на проектирование выводится окно заставки программы, приведенное на рисунке 4.

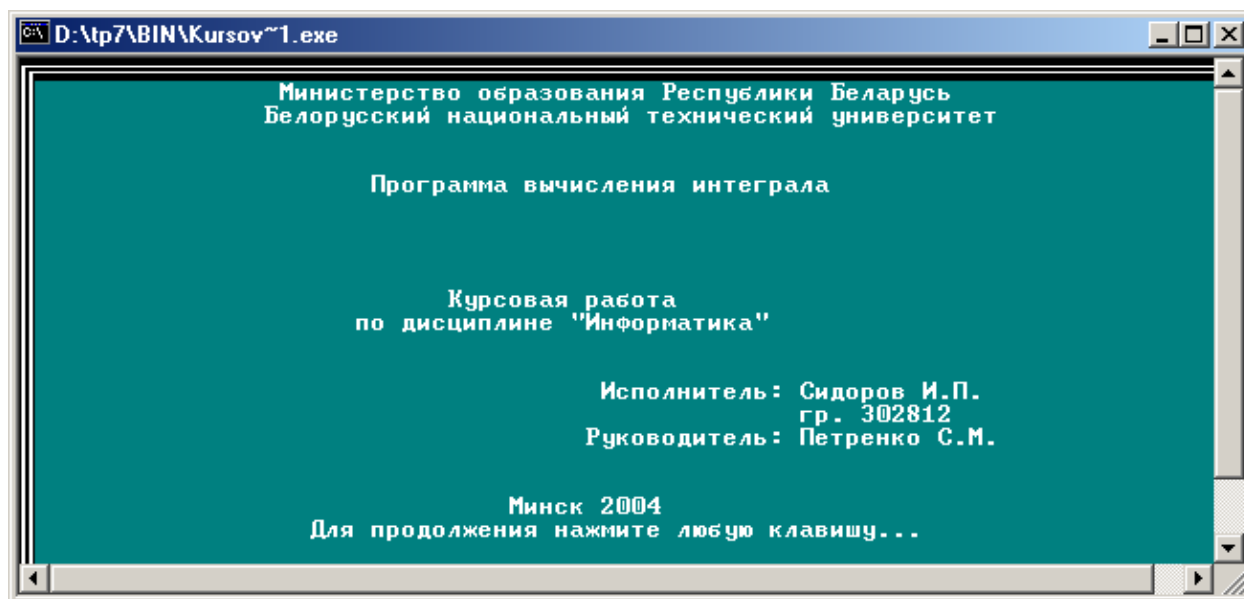


Рисунок 4. Окно заставки.

После нажатия любой клавиши выводится окно с меню, представленное на рисунке 5.

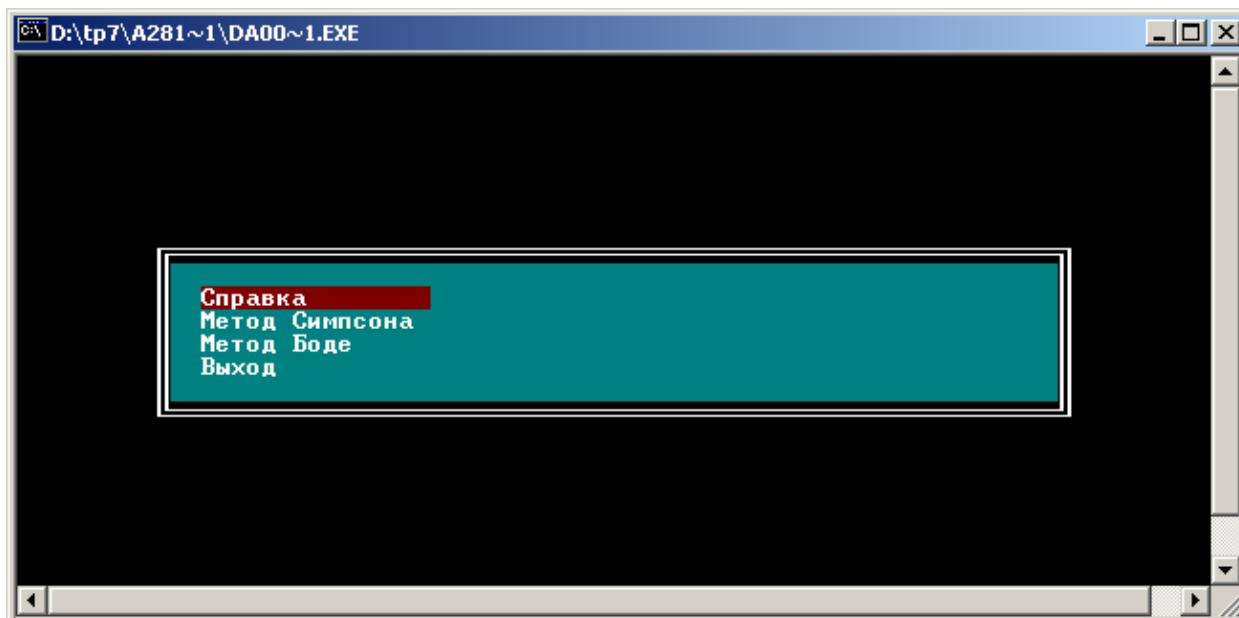


Рисунок 5. Окно с меню.

При выборе пункта меню «Справка» открывается соответствующее окно в котором находится справочная информация о программе. Пример такого окна представлен на рисунке 6.

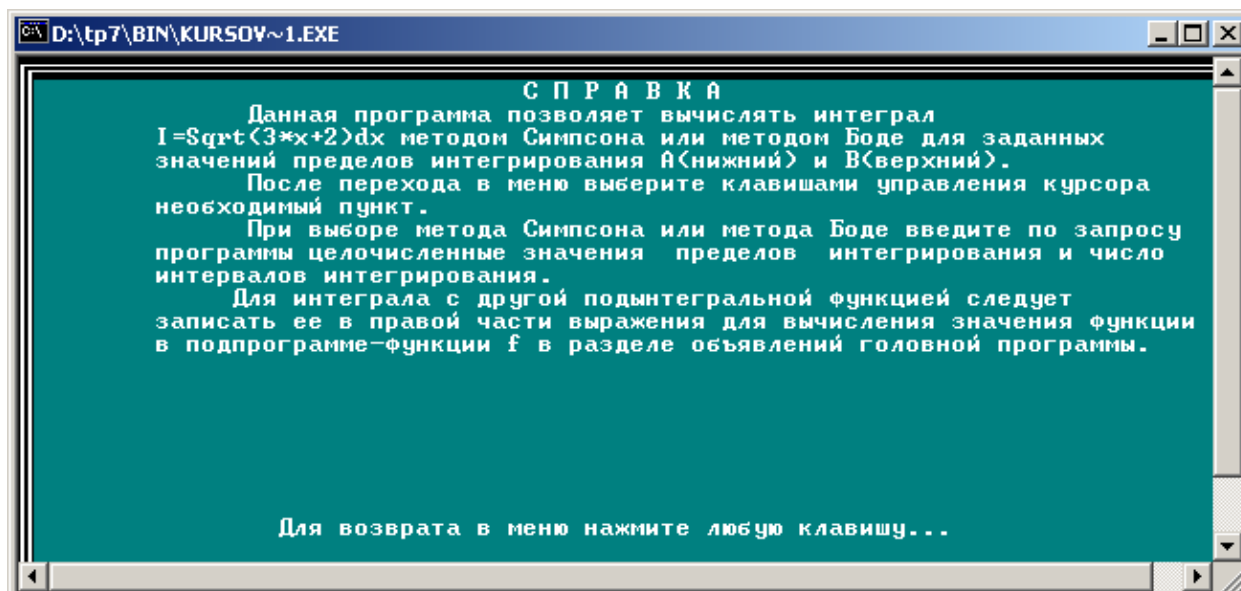


Рисунок 6. Окно справки.

При выборе пункта меню «метод Симпсона» или «метод Бode» открывается соответствующее окно, в которых вводятся исходные данные и, после подтверждения правильности введенных данных, выводятся результаты вычисления. Примеры таких окон представлены на рисунках 7 (метод Симпсона) и 8 (метод Бode).

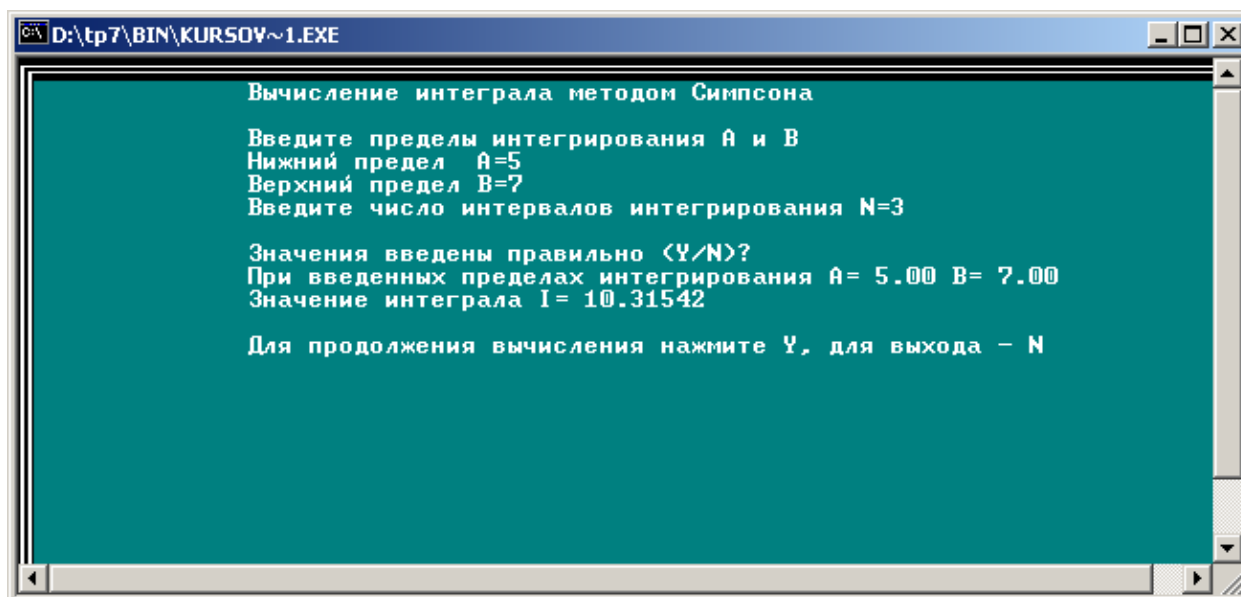


Рисунок 7. Вычисление интеграла методом Симпсона.


```
D:\tp7\BIN\KURSOV~1.EXE

Вычисление интеграла методом Боде

Введите пределы интегрирования А и В
Нижний предел А=5
Верхний предел В=9
Введите число интервалов интегрирования N=3

Значения введены правильно <Y/N>?
Данные введены неверно. Повторяем ввод

Введите пределы интегрирования А и В
Нижний предел А=5
Верхний предел В=8
Введите число интервалов интегрирования N=10

Значения введены правильно <Y/N>?
При введенных пределах интегрирования А= 5.00 В= 8.00
Значение интеграла I= 14.34522

Для продолжения вычисления нажмите Y, для выхода - N
```

Рисунок 8. Вычисление интеграла методом Боде.

8. Анализ результатов

Для анализа результатов сведем данные нескольких расчетов представленного интервала при различных пределах интегрирования и при различном количестве интервалов интегрирования в таблицу.

Таблица 2 – Результаты расчетов

Пределы интегрирования	Число интервалов интегрирования	Значение интервала при использовании различных методов		Точное значение интеграла $I = \frac{2}{9} (3x + 2)^{\frac{3}{2}} \Big _a^b$
		Метод Симпсона	Метод Боде	
$I = \int_0^1 \sqrt{3x + 2} dx$	N=2	1,85595	1,85598	1,855981
	N=5	1,85598	1,85598	
	N=10	1,85598	1,85598	
$I = \int_0^4 \sqrt{3x + 2} dx$	N=2	11,00783	11,01200	11,012172
	N=5	11,01198	11,01217	
	N=10	11,01216	11,01217	
$I = \int_0^{10} \sqrt{3x + 2} dx$	N=2	39,53758	39,59271	39,597979
	N=5	39,59358	39,59781	
	N=10	39,59754	39,59797	

При анализе точности расчетов различными методами, очевидно, что метод Боде более точен, чем метод Симпсона.

Из таблицы видно, что при маленьком отрезке интегрирования (от 0 до 1, от 0 до 4) достаточно большая точность вычисления достигается обоими методами даже при небольшом (до 5) количестве интервалов интегрирования. При увеличении отрезка интегрирования (от 0 до 10) при небольшом количестве интервалов интегрирования метод Симпсона менее точен, чем метод Боде. Однако даже в этом случае погрешность обоих методов составляет незначительную величину (менее 0,2%).

Можно рекомендовать при нахождении значения интеграла с достаточно большим отрезком интегрирования для получения хорошей точности результата использовать не менее 10-ти интервалов интегрирования.

9. Инструкция по работе с программой

Файл `integral.pas` с исходным текстом Паскаль-программы находится по адресу `D:\kurs812\Sidorov`. Результат компиляции исходной программы `integral.exe` находится в той же папке `Sidorov`.

Необходимо запустить на выполнение `integral.exe` либо загрузить Turbo Pascal 7.0, сделать текущим каталог `Sidorov`, открыть файл с исходной Паскаль-программой `integral.pas` и запустить ее на выполнение командой `Run`.

После вывода заставки программы нажать любую клавишу для перехода в меню и открыть окно справки. После ознакомления со справкой нажатием любой клавиши возвратится в окно меню, с помощью клавиш управления курсором выбрать метод вычисления и ввести по запросу программы целочисленные значения нижнего и верхнего предела интегрирования, а также число интервалов интегрирования. Затем проверить правильность введенных данных и сообщить результат программе (да-`Y`, нет-`N`). При необходимости повторить процедуру ввода исходных данных.

Результаты вычисления интеграла выводятся в том же окне, где вводились исходные данные.

Для продолжения вычислений следует ответить на запрос программы нажатием клавиши `'Y'`, что обеспечит переход к меню программы. Нажатие клавиши `'N'` приводит к переходу в окно заставки программы.

Завершение работы с программой реализуется выбором пункта меню "Выход".

Заключение

В данной курсовой работе разработана блок-схема и реализована средствами языка программирования Турбо-Паскаль программа, позволяющая вычислить интеграл

$$I = \int_a^b \sqrt{3x + 2} dx \text{ методом Симпсона или методом Боде.}$$

Программа обладает дружеским интерфейсом, по запросу пользователя сообщает общие сведения по работе с программой и производит вычисления, выбор метода вычисления производится посредством меню, ввод пределов интегрирования и количества интегралов интегрирования осуществляется с клавиатуры по запросу программы.

Из анализа результатов вычисления можно сделать вывод о большей точности вычисления по методу Боде по сравнению с методом Симпсона.

Список использованных источников

1. Начала программирования на языке Паскаль/С.А.Абрамов – М., 1987
2. Программирование в среде Турбо-Паскаль/Д.Б.Поляков – М., 1992
3. Численные методы/Н.Н.Калиткин –М.: «Наука», 1978
4. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ/В.П.Дьяконов – М.: Наука, 1987
5. Вычислительные основы линейной алгебры/В.В.Воеводин – М.: Наука, 1977
6. Численные методы анализа/Б.П.Демидович – М.: Государственное издательство физико-математической литературы, 1962