



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Информатика»

Н. В. Самовендюк, С. А. Чабуркина

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

ПРАКТИКУМ

**по выполнению лабораторных работ
по одноименной дисциплине
для студентов специальности 1-53 01 01
«Автоматизация технологических процессов
и производств (по направлениям)»
дневной формы обучения**

Электронный аналог печатного издания

Гомель 2018

УДК 004.42(075.8)
ББК 32.973.22я73
С17

*Рекомендовано к изданию научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 1 от 04.09.2017 г.)*

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
канд. физ.-мат. наук, доц. *О. А. Кравченко*

Самовендюк, Н. В.

С17

Алгоритмизация и программирование : практикум по выполнению лаборатор. работ по одноим. дисциплине для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)» днев. формы обучения / Н. В. Самовендюк, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2018. – 60 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

ISBN 978-985-535-379-0.

Представлены задания к лабораторным работам, с помощью которых выполняется разработка и практическая реализация основных видов алгоритмов на языке высокого уровня Object Pascal. Рассмотрены сложные структуры данных и алгоритмы их обработки, а также алгоритмы вычислительной математики.

Для студентов специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)» дневной формы обучения.

**УДК 004.42(075.8)
ББК 32.973.22я73**

ISBN 978-985-535-379-0

© Самовендюк Н. В., Чабуркина С. А., 2018
© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2018

ВВЕДЕНИЕ

Дисциплина «Алгоритмизация и программирование» является одной из базовых при подготовке студентов по специальности 1-53 01 01 «Автоматизация технологических процессов и производств (по направлениям)». Целью дисциплины является изучение основ алгоритмизации и закрепление полученных навыков на основе программирования алгоритмов обработки структур данных и алгоритмов вычислительной математики.

Практикум по выполнению лабораторных работ ориентирован на формирование у студентов основных понятий и принципов алгоритмизации и программирования, рациональное использование системы программирования.

В практикуме представлены задания к лабораторным работам, с помощью которых выполняется разработка и практическая реализация основных видов алгоритмов на языке высокого уровня Object Pascal.

Рассматриваются различные структуры данных (двумерные массивы, массивы записей, файлы, динамические структуры) и алгоритмы их обработки, а также алгоритмы вычислительной математики (вычисление интегралов, решение нелинейных уравнений и систем линейных уравнений).

Все задания к лабораторным работам включают разработку приложений в системе программирования, использующих библиотеку визуальных компонентов, и создание приложений, содержащих несколько форм.

Практикум по выполнению лабораторных работ полностью соответствует учебной программе по дисциплине «Алгоритмизация и программирование».

ОБЩИЕ ТРЕБОВАНИЯ К ПОРЯДКУ ВЫПОЛНЕНИЯ РАБОТ И ПО СОДЕРЖАНИЮ ОТЧЕТОВ

Порядок выполнения каждой лабораторной работы предусматривает следующие действия:

1. Алгоритмический анализ задачи.
 - 1.1. Анализ исходных и выходных данных.
 - 1.2. Составление алгоритма решения задачи в соответствии с вариантом задания.
 - 1.3. Подготовка тестовых наборов.
2. Разработка графического интерфейса программы с использованием визуальных компонентов.
3. Реализация разработанного алгоритма на языке Object Pascal.
4. Проверка работоспособности программы и правильности вычислений.
5. Оформление отчета.

Отчет по лабораторной работе должен включать:

1. Титульный лист.
2. Цель работы.
3. Текст задания в соответствии со своим вариантом.
4. Вид окна проекта с указанием визуальных компонентов (ВК).
5. Таблицу используемых ВК.
6. Схему алгоритма решения задания.
7. Таблицу используемых переменных.
8. Тестовые наборы для проверки правильности решения задачи.
9. Программный код алгоритма решения задачи.
10. Результаты работы программы с одним из тестовых наборов.
11. Выводы.

Лабораторная работа № 1
ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ

Цель работы: получить навыки разработки и реализации типовых алгоритмов обработки двумерных массивов на языке Object Pascal.

Краткие теоретические сведения

Массив может иметь несколько измерений, перечисляемых через запятую. Двумерные массивы характеризуются двумя измерениями: первое из них отвечает за индекс (номер) строки, а второе – за индекс столбца. Таким образом, доступ к элементу массива происходит при указании имени массива и в квадратных скобках через запятую указываются номер строки и номер столбца.

$A[1,1]$ – обращение к элементу первой строки первого столбца.

Описание массива

Var <имя массива> : array[<диапазон1, диапазон2>] of <тип элемента>;

Type <имя типа массива>= array[<диапазон1, диапазон2>] of <тип элемента>;

Var <имя массива>:< имя типа массива >;

Диапазон 1 и диапазон 2 определяют нижнюю и верхнюю границы массива по строкам и столбцам и, соответственно, число элементов в нем. При обращении к массиву индекс 1 должен лежать в диапазоне 1, а индекс 2 должен лежать в диапазоне 2. Тип элемента определяет тип каждого элемента массива.

Var Matr:array[1..20,1..20] of real;

Ввод-вывод двумерного массива

Для ввода-вывода элементов двумерного массива используется визуальный компонент StringGrid. Пусть задан двумерный массив, содержащий n -строк и m -столбцов. Тогда программный код ввода-вывода массива:

n:=StrToInt(Edit1.Text);

```

m:=StrToInt(Edit2.Text);
for i:=1 to n do
  for j:=1 to m do
    Matr[i,j]:=StrToFloat(StringGrid1.Cells[j-1,i-1]);
for i:=1 to n do
  for j:=1 to m do
    StringGrid2.Cells[j-1,i-1]:=FloatToStr(Matr[i,j]);

```

Обработка двумерного массива

Обработка двумерного массива заключается в последовательном переборе элементов массива и выполнении необходимых действий. Например, необходимо вычислить сумму отрицательных элементов массива. Тогда программный код этого фрагмента программы:

```

sum:=0;
for i:=1 to n do
  for j:=1 to m do
    if Matr[i,j]<0 then sum:=sum+Matr[i,j];

```

Практическая часть

Задание. Разработать алгоритм и написать программу с графическим интерфейсом в соответствии со своим вариантом (табл. 1.1). Программа должна:

- вводить размерность и элементы матрицы;
- выполнять действия в соответствии с условием задачи;
- выводить исходные данные и результаты вычислений.

Исходные данные для отладки программы выбрать самостоятельно.

Таблица 1.1

Обработка двумерных массивов

Вариант	Задание
1	В каждой нечетной по номеру строке матрицы найти минимальный элемент и вычислить произведение этих элементов
2	В каждом столбце матрицы найти произведение положительных элементов и вычислить сумму этих произведений
3	Определить количество столбцов матрицы, в которых больше трех положительных элементов

Вариант	Задание
4	Определить количество строк матрицы, в которых суммы всех элементов отрицательные
5	В каждой строке матрицы найти самый левый отрицательный элемент и вычислить произведение этих элементов
6	Определить количество строк матрицы, в которых нет положительных элементов
7	Заменить в матрице элементы последней строки на произведение элементов соответствующих столбцов
8	В каждом столбце матрицы найти минимальный элемент и вычислить сумму этих элементов
9	В каждой строке матрицы найти произведение отрицательных элементов и вычислить сумму этих произведений
10	Заменить в матрице элементы предпоследней строки на минимальные элементы соответствующих столбцов
11	Определить количество строк матрицы, в которых произведение положительных элементов больше заданного B
12	В каждой строке матрицы найти самый правый положительный элемент и вычислить сумму этих элементов
13	Определить количество столбцов матрицы, в которых нет положительных элементов
14	Заменить в матрице элементы последнего столбца на суммы элементов соответствующих строк
15	В каждом столбце матрицы найти максимальный элемент и вычислить произведение этих элементов
16	В каждой четной по номеру строке матрицы найти минимальный элемент и вычислить сумму этих элементов.
17	В каждом столбце матрицы найти сумму отрицательных элементов и вычислить произведение этих сумм
18	Определить количество строк матрицы, в которых произведение элементов положительное
19	Заменить в матрице элементы второго столбца на суммы положительных элементов соответствующих строк
20	В каждом столбце матрицы найти первый отрицательный элемент и вычислить сумму этих элементов
21	Определить количество строк матрицы, в которых все элементы отрицательные
22	Заменить в матрице элементы предпоследнего столбца на суммы элементов соответствующих строк

Вариант	Задание
23	В каждом нечетном по номеру столбце матрицы найти минимальный элемент и вычислить произведение этих элементов
24	В каждой строке матрицы найти сумму положительных элементов и вычислить произведение этих сумм
25	Определить количество столбцов матрицы, в которых все элементы положительные
26	Заменить в матрице элементы предпоследней строки на количество положительных элементов соответствующих столбцов
27	В каждом столбце матрицы найти самый нижний положительный элемент и вычислить произведение этих элементов
28	Определить количество строк матрицы, в которых нет положительных элементов
29	Определить количество столбцов матрицы, в которых все элементы нулевые
30	В каждом столбце матрицы найти минимальный элемент и вычислить сумму этих элементов

Контрольные вопросы для защиты работ

1. Как выполнить описание двумерного массива?
2. Как обратиться к элементу массива?
3. Опишите массив вещественных чисел, который может содержать не более 5 строк и 10 столбцов.
4. Какой визуальный компонент используется для ввода элементов массива? Какие свойства необходимо для него установить в Инспекторе Объектов?
5. Как выполнить ввод массива? Нарисуйте схему алгоритма и напишите фрагмент процедуры ввода.
6. Какой визуальный компонент используется для вывода элементов массива? Какие свойства необходимо для него установить в Инспекторе Объектов?
7. Как выполнить вывод массива? Нарисуйте схему алгоритма и напишите фрагмент процедуры вывода.
8. Вычислить сумму элементов заданных строк.
9. Вычислить произведение отрицательных элементов в определенных столбцах.

Лабораторная работа № 2

ОБРАБОТКА СИМВОЛЬНОЙ ИНФОРМАЦИИ

Цель работы: получить навыки разработки и реализации алгоритмов обработки символьной информации на языке Object Pascal.

Краткие теоретические сведения

Строка – это последовательность символов. При программировании строковые значения заключаются в апострофы, например:

```
Edit1.Text('I like music');
```

Описание строковых переменных

Строковая переменная объявляется с помощью зарезервированного слова `string` или с помощью идентификатора типа данных `AnsiString`, например:

```
var  
  FileName: string;  
  EditText: AnsiString;  
  FirstName: string[20];
```

Строку можно считать бесконечной, хотя на самом деле ее длина ограничена 2 ГБ. Поэтому в последней строчке определяется короткая строка с максимально допустимой длиной, равной 20.

Обработка строк

Строка представляет из себя одномерный массив, где в качестве базового элемента используются символы. Следовательно, при обработке строк можно применять все известные алгоритмы обработки одномерных массивов. Однако стандартно подключаемый модуль `System` имеет набор процедур и функций, значительно облегчающих этот процесс.

Concat(S1, S2, ... , Sn): string – возвращает строку, полученную в результате сцепления строк S1, S2, ..., Sn. По своей работе функция `Concat` аналогична операции конкатенации (+).

Copy(S: string, Index, Count: Integer): string – выделяет из строки S подстроку длиной Count символов, начиная с позиции Index.

Delete(var S: string, Index, Count: Integer) – удаляет Count символов из строки S, начиная с позиции Index.

Insert(Source: string; var S: string, Index: Integer) – вставляет строку Source в строку S, начиная с позиции Index.

Length(S: string): Integer – возвращает реальную длину строки S в символах.

Pos(Substr, S: string): Byte – обнаруживает первое появление подстроки Substr в строке S. Возвращает номер той позиции, где находится первый символ подстроки Substr. Если в S подстроки Substr не найдено, результат равен 0.

Практическая часть

Задание 1. Обработка строк. Работа с символами.

В каждой из задач (табл. 2.1) предполагается, что задана строка текста, состоящая не более чем из 255 символов. Программа должна:

- вводить строку текста;
- выполнять вычисления или преобразования в соответствии с условием задачи;
- выводить:
 - а) исходную строку и ее длину;
 - б) результаты вычислений при обработке строки;
 - в) преобразованную строку и ее длину.

Исходные данные для отладки программы выбрать самостоятельно. Для ввода и вывода строки использовать компоненты TEdit или TМемо.

Таблица 2.1

Обработка символов

Вариант	Задание
1	Подсчитать общее количество символов '+' и '-' и заменить каждый символ ';' на ',' и ' '
2	После каждого символа ',' вставить пробел и подсчитать количество букв 'A' и 'B' отдельно
3	Заменить символ '*' на '++' и подсчитать общее количество букв 'F' и 'D'
4	Подсчитать количество букв 'C' и 'D' отдельно и заменить каждую пару символов '**' на символ ' '
5	После каждого символа '!' вставить символ 'Г' и подсчитать общее количество цифр в строке

Вариант	Задание
6	Удалить каждую пару символов 'PQ' и подсчитать общее количество символов '.' и ',' в строке
7	Подсчитать количество пар символов '+ -' и заменить каждый символ '*' на '/-'
8	После каждой цифры вставить такую же цифру и подсчитать количество пар 'AC' в строке
9	Удалить каждый символ 'A', стоящий после ',' и подсчитать количество пар 'BC'
10	Подсчитать количество символов '.', стоящих перед пробелом, и заменить каждую пару символов 'ST' на символ 'P'
11	После каждого символа 'A' вставить пробел и подсчитать количество символов 'B', стоящих между знаками '+' и '-'
12	Удалить каждый символ '?', стоящий после ';', и подсчитать общее количество символов 'o' и 'O'
13	Подсчитать количество символов '+', стоящих между 'A' и 'B', заменить каждый символ '0' на 'OO'
14	В каждую пару символов 'AB' вставить символ '*', подсчитать, сколько раз в строке символ 'I' стоит перед '2'
15	Вставить символ ';' после каждого символа 'A' и после 75 каждого 'B', подсчитать, сколько раз символ 'C' встречается между символами '*' и '/'

Задание 2. Работа со строками, состоящими из последовательности слов.

В каждой из задач (табл. 2.2) предполагается, что задана строка текста, состоящая из последовательности слов. Слова разделяются последовательностью пробелов. Программа должна:

- вводить строку текста;
- выполнять вычисления или преобразования в соответствии с условием задачи;
- выводить:
 - а) исходную строку и ее длину;
 - б) результаты вычислений при обработке строки;
 - в) преобразованную строку и ее длину.

Исходные данные для отладки программы выбрать самостоятельно. Для ввода и вывода строки использовать компоненты TEdit или TМето.

Обработка последовательности слов

Вариант	Задание
1	После каждого слова поставить запятую. Подсчитать количество слов, в которых есть буква 'п'
2	Подсчитать количество букв в третьем слове. Поменять местами первое и последнее слова
3	Во втором слове после каждой буквы вставить пробел. Определить количество слов, которые заканчиваются на 'е'
4	Перед первой буквой каждого слова вставить символ '*'. Определить количество слов, в которых нет ни одной буквы 'г'
5	Для первого слова указать, сколько букв 'и' в нем содержится. Переставить первое слово в конец строки
6	Определить количество слов, начинающихся с буквы 'а'. После каждой буквы предпоследнего слова вставить символ '*'
7	Подсчитать количество букв во втором слове. Каждое слово заключить в кавычки
8	Подсчитать количество слов, длина которых больше 5. Удалить пробелы, стоящие между первым и вторым словом
9	Определить количество слов, в которых буква 'и' встречается хотя бы один раз. Поменять местами первое и второе слова
10	Третье слово строки поставить после первого. Определить количество слов, в которых первая и последняя буквы совпадают
11	Определить количество слов, вторая буква которых 'р'. Удалить последнюю букву в каждом слове
12	Подсчитать количество букв в предпоследнем слове. В каждом слове поменять местами первую и последнюю буквы
13	Перед каждой буквой третьего слова поставить '-'. Определить количество слов, после которых один пробел
14	После последней буквы каждого слова вставить точку. Для пятого слова указать, сколько букв 'и' в нем содержится
15	Удалить все пробелы из строки, кроме тех, которые стоят между первым и вторым словом. Определить количество слов, которые по длине меньше 3

Контрольные вопросы для защиты работ

1. Как представляется символьная информация в памяти компьютера?
2. Какие типы данных используются для хранения символьной информации?

3. Для чего служат кодовые таблицы? Приведите примеры кодовых таблиц.

4. Назовите стандартные функции для работы с символами. Приведите примеры их использования.

5. Назовите стандартные функции для работы со строками. Приведите примеры их использования.

6. Назовите стандартные процедуры для работы со строками. Приведите примеры их использования.

Лабораторная работа № 3

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПОДПРОГРАММ

Цель работы: изучить типы подпрограмм в Object Pascal, их структуру, возможности применения, обращение к подпрограммам; получить навыки использования процедур и функций при разработке приложений.

Краткие теоретические сведения

Подпрограммой называется именованная логически законченная группа операторов, которую можно вызвать по имени (т. е. выполнить) любое количество раз из различных мест программы. В языке Object Pascal подпрограммы оформляются в виде процедур и функций.

Процедура – это подпрограмма, имя которой не может использоваться в выражениях в качестве операнда. Процедура состоит из заголовка и тела. По структуре ее можно рассматривать как программу в миниатюре. Когда процедура описана, ее можно вызвать по имени из любой точки программы (в том числе из нее самой!). Когда процедура выполнит свою задачу, программа продолжится с оператора, следующего непосредственно за оператором вызова процедуры. Использование имени процедуры в программе называется оператором вызова процедуры.

Функция также является подпрограммой, но в отличие от процедуры ее имя может использоваться в выражениях в качестве операнда, на место которого подставляется результат работы этой функции.

Все процедуры и функции языка Object Pascal подразделяются на две группы: встроенные и пользователя, определенные программистом.

Описание и вызов подпрограмм пользователя

По своей структуре подпрограммы пользователя очень напоминают программу и состоят из заголовка и блока операторов.

Описание процедуры:

```
procedure <имя процедуры> ( <список формальных параметров> );  
const ...;  
type ...;  
var ...;  
begin  
    <операторы>  
end;
```

Описание функции:

```
function <имя функции> ( <список формальных параметров> ):  
<тип результата>;  
const ...;  
type ...;  
var ...;  
begin  
    <операторы>  
end;
```

Заголовок подпрограммы состоит из зарезервированного слова **procedure** или **function**, имени подпрограммы, необязательного, заключенного в круглые скобки списка формальных параметров. При описании подпрограммы-функции дополнительно описывается тип возвращаемого результата. Тело подпрограммы представляет собой локальный блок, по структуре аналогичный программе.

Вызов подпрограммы для выполнения осуществляется по ее имени, за которым в круглых скобках следует список фактических параметров, т. е. передаваемых в подпрограмму данных:

```
<имя процедуры> ( <список фактических параметров> );
```

Параметры процедур и функций

Параметры служат для передачи исходных данных в подпрограммы и для приема результатов работы этих подпрограмм.

Исходные данные передаются в подпрограмму с помощью **входных** параметров, а результаты работы подпрограммы возвращаются через **выходные** параметры. Параметры могут быть входными и выходными одновременно.

Входные параметры объявляются с помощью ключевого слова **const**; их значения не могут быть изменены внутри подпрограммы. Для объявления **выходных параметров** используется ключевое слово **out**. Если параметр является одновременно и **входным**, и **выходным**, то он описывается с ключевым словом **var**. Если при описании параметра не указано ни одно из ключевых слов – **const**, **out**, или **var**, то параметр считается входным.

Результатом выполнения подпрограммы-функции является оператор, присваивающий возвращаемое значение имени функции или неявной локальной переменной **result**.

Для передачи массива в качестве параметра подпрограммы он должен быть предварительно описан как пользовательский тип данных:

```
Type mas=array[1..10] of integer;
```

Подпрограммы ввода и вывода массива могут быть оформлены в следующем виде:

```
Procedure vvod(var a:mas; var b:integer; c:TStringGrid);  
  Var i:integer;  
  Begin  
    for i:=1 to b do  
      a[i]:=StrToInt(c.Cells[i-1,0]);  
  End;
```

```
Procedure vyvod(a:mas; b:integer; c:TStringGrid);  
  Var i:integer;  
  Begin  
    for i:=1 to b do  
      c.Cells[i-1,0]:=inttostr(a[i]);  
  End;
```

Обращение из основной программы для ввода и вывода массива:

```
Vvod(masA,sizeA,StrinGrid1);  
Vyvod(masA,sizeA,StringGrid2);
```

Рассмотрим пример определения количества положительных элементов массива с процедуры:

```
procedure kol(a:mas; b:integer; var k:integer);  
var i:integer;  
begin  
k:=0;  
for i:=1 to b do  
if a[i]>0 then k:=k+1;  
end;
```

Вызов из основной программы:

```
kol(masA,sizeA,countA);
```

То же самое, но с использованием функции:

```
function kol(a:mas;b:integer):integer;  
var i,k:integer;  
begin  
k:=0;  
for i:=1 to b do  
if a[i]>0 then k:=k+1;  
result:=k;  
end;
```

Вызов из основной программы:

```
countA:=kol(masA,sizeA);
```

Практическая часть

Задание. Разработать приложение с графическим интерфейсом для обработки одномерных массивов с использованием подпрограмм. В каждой задаче (табл. 3.1) заданы три одномерных массива разной длины. Программа должна:

- вводить массивы, используя процедуру ввода и главную форму с тремя визуальными компонентами для ввода размерностей массива и тремя ВК для ввода элементов массивов, или модальное окно для ввода каждого массива;

- выводить исходные массивы, используя процедуру вывода и форму с тремя ВК TStringGrid или ВК TMemo;

- выполнять вычисления в соответствии с условием задачи, используя подпрограмму-функцию;

- выводить промежуточные результаты и вычисленное значение функции.

Исходные данные для отладки программы выбрать самостоятельно.

Таблица 3.1

Программирование с использованием подпрограмм

Вариант	Задание
1	Вычислить значение функции $\cos^2 x + 2\cos y + \sin z$, где x, y, z – произведение положительных элементов в массивах А, В и С, соответственно
2	Вычислить значение функции $3x + 3\sin y + \sin^2 z$, где x, y, z – суммы отрицательных элементов, расположенных на местах с четными номерами массивов А, В и С, соответственно
3	Вычислить значение функции $3x^2 + 3y + \cos^2 z$, где x, y, z – количество положительных элементов в массивах А, В и С, соответственно
4	Вычислить значение функции $3x + 3\sin(y + z^2)$, где x, y, z – минимальные элементы массивов А, В и С, соответственно
5	Вычислить значение функции $3\cos x + \sin(y + z)$, где x, y, z – максимальные элементы среди элементов, расположенных на местах с четными номерами массивов А, В и С, соответственно
6	Вычислить значение функции $3\sin x + \cos(y^2 + z^2)$, где x, y, z – количество отрицательных элементов в массивах А, В и С, соответственно
7	Вычислить значение функции $3x + 3\sin(y + z^2)$, где x, y, z – минимальные элементы массивов А, В и С, соответственно
8	Вычислить значение функции $3x - 3\sin(y + z^2) + \sqrt{ x }$, где x, y, z – максимальные элементы массивов А, В и С, соответственно
9	Вычислить значение функции $3\sin x^2 + 3\sqrt{y^2 + 1} + \cos^2 z$, где x, y, z – произведение элементов, расположенных на местах с нечетными номерами в массивах А, В и С, соответственно

Вариант	Задание
10	Вычислить значение функции $3 - 3\sin(y + z^2) + \sqrt{ x }$, где x, y, z – максимальные значения среди элементов, расположенных на местах с четными номерами, массивов А, В и С, соответственно
11	Вычислить значение функции $x + 3\sin(y + z - 4)$, где x, y, z – суммы элементов, расположенных на местах с нечетными номерами, в массивах А, В и С, соответственно
12	Вычислить значение функции $3(x + \sin y) + \cos^2 z$, где x, y, z – произведение элементов, расположенных на местах с четными номерами, в массивах А, В и С, соответственно
13	Вычислить значение функции $3x + \sqrt{(y^2 + z^2)}$, где x, y, z – количество элементов, не принадлежащих интервалу $[0; 5]$ в массивах А, В и С, соответственно
14	Вычислить значение функции $3x + 3\sin(y + z^2)$, где x, y, z – суммы квадратов элементов, расположенных на местах с четными номерами, в массивах А, В и С, соответственно
15	Вычислить значение функции $3\cos x + 3(\sin y + \sin 2z)$, где x, y, z – произведение элементов, расположенных на местах с номерами, кратными 3, в массивах А, В и С, соответственно
16	Вычислить значение функции $\cos^2 x + 2\cos y + \sin z$, где x, y, z – произведение всех элементов в массивах А, В и С, соответственно
17	Вычислить значение функции $3x + 3\sin y + \sin^2 z$, где x, y, z – количество положительных элементов массивов А, В и С, соответственно
18	Вычислить значение функции $3x^2 + 3y + \cos^2 z$, где x, y, z – среднее арифметическое отрицательных элементов массивов А, В и С, соответственно
19	Вычислить значение функции $3x + 3\sin(y + z^2)$, где x, y, z – минимальные среди элементов на местах с нечетными номерами массивов А, В и С, соответственно
20	Вычислить значение функции $3\cos x + \sin(y + z)$, где x, y, z – произведение элементов, расположенных на местах с четными номерами в массивах А, В и С, соответственно
21	Вычислить значение функции $3\sin x + \cos(y^2 + z^2)$, где x, y, z – количество отрицательных элементов в массивах А, В и С, соответственно
22	Вычислить значение функции $3x + 3\sin(y + z^2)$, где x, y, z – минимальные элементы массивов А, В и С, соответственно
23	Вычислить значение функции $3x - 3\sin(y + z^2) + \sqrt{ x }$, где x, y, z – максимальные элементы массивов А, В и С, соответственно

Вариант	Задание
24	Вычислить значение функции $3\sin x^2 + 3\sqrt{y^2 + 1} + \cos^2 z$, где x, y, z – суммы квадратов элементов в массивах А, В и С, соответственно
25	Вычислить значение функции $3 - 3\sin(y + z^2) + \sqrt{ x }$, где x, y, z – среднее арифметическое элементов, расположенных на местах с четными номерами, в массивах А, В и С, соответственно
26	Вычислить значение функции $x + 3\sin(y + z - 4)$, где x, y, z – суммы элементов, расположенных на местах с нечетными номерами, в массивах А, В и С, соответственно
27	Вычислить значение функции $3(x + \sin y) + \cos^2 z$, где x, y, z – произведение положительных элементов, расположенных на местах с четными номерами, в массивах А, В и С, соответственно
28	Вычислить значение функции $3x + \sqrt{(y^2 + z^2)}$, где x, y, z – суммы элементов, принадлежащих интервалу $[-10; 10]$, в массивах А, В и С, соответственно
29	Вычислить значение функции $3x + 3\cos(y + \sin z)$, где x, y, z – произведение ненулевых элементов в массивах А, В и С, соответственно
30	Вычислить значение функции $3\cos x + 3(\sin y + \sin 2z)$, где x, y, z – суммы элементов, расположенных на местах с номерами, кратными 3, в массивах А, В и С, соответственно

Контрольные вопросы для защиты работ

1. Для всех подпрограмм в выполненной работе указать их описание и обращение к ним, формальные и фактические параметры этих подпрограмм.

2. Как формируется список фактических параметров подпрограммы? В чем заключается его соответствие списку формальных параметров?

3. Какие формальные параметры называются входными, какие – выходными? В чем отличие в записи этих параметров в списке формальных параметров?

4. В каком случае подпрограмма может быть реализована в виде функции?

5. Как передает результат работы в вызывающую процедуру функция и как – процедура? Можно ли те действия, которые выполняет функция, реализовать в виде процедуры? Если нет, то почему? Если да, то написать описание такой процедуры и изменить обращение к функции на обращение к процедуре.

6. Можно ли те действия, которые выполняет процедура, реализовать в виде функции? Если нет, то почему? Если да, то написать описание такой функции и изменить обращение к процедуре на обращение к функции.

7. Написать описание процедуры и функции и обращение к ним для вычисления суммы, произведения, количества, среднего арифметического, определения максимума и минимума, формирования нового массива при обработке одномерных и двумерных массивов.

Лабораторная работа № 4 **АЛГОРИТМЫ СОРТИРОВКИ**

Цель работы: изучить типовые алгоритмы быстрой и медленной сортировок.

Краткие теоретические сведения

Задача сортировки заключается в упорядочении заданного списка каких-либо элементов в возрастающем порядке. На практике обычно требуется отсортировать по возрастанию список чисел, расположить символы и строки в алфавитном порядке или, что наиболее важно, упорядочить набор записей, содержащих различную информацию: сведения о студентах вуза, читательских формулярах библиотеки, сотрудниках организации и т. д. В случае набора записей необходимо выбрать фрагмент записи, содержащий данные, по которым будет осуществляться сортировка. Например, набор записей о студентах можно отсортировать по фамилии, номеру студенческого билета или среднему баллу. Специально отобранный фрагмент данных, по которому осуществляется сортировка, называется **ключом**.

На данный момент специалистами по вычислительной технике разработаны десятки алгоритмов сортировки. Тем не менее, пока еще не придуман универсальный алгоритм сортировки, который бы наилучшим образом подходил для всех случаев.

Классификация методов сортировки

Все методы сортировки можно разделить на пять групп:

- 1) методы извлечения;
- 2) методы включения;
- 3) методы обменов;

- 4) методы слияния;
- 5) методы распределения.

Общая концепция **методов извлечения** заключается в следующем: из исходного массива извлекается минимальный элемент и меняется местами с первым элементом массива, затем извлекается минимальный элемент из части массива, начиная со второго элемента, и меняется местами со вторым элементом и т. д. Последний раз минимальный элемент выбирается из двух последних элементов массива. В результате получится массив, упорядоченный по неубыванию. Различные методы извлечения отличаются объектом извлечения (минимальный или максимальный элемент) и, соответственно, объектами перестановки (первый или последний элемент), а также условием окончания процесса сортировки.

Идея **методов включения** состоит в том, что сначала первый элемент массива рассматривается как упорядоченный массив и в этот массив включается следующий элемент исходного массива так, чтобы получился упорядоченный по неубыванию массив из двух элементов. Затем в полученный упорядоченный массив включается третий элемент массива так, чтобы опять-таки получился упорядоченный массив. Процесс продолжается до тех пор, пока не будет включен последний элемент. Различные алгоритмы включения отличаются способами выбора элемента для включения, способами определения места включения и методами самого включения.

Идея **методов обменов** состоит в следующем: в исходном массиве выбирается пара элементов, и они сравниваются между собой. Если их положение не удовлетворяет требованию упорядоченности, то элементы переставляются. Затем выбирается следующая пара элементов и так – до тех пор, пока не получим упорядоченный массив. Различные алгоритмы обменов отличаются способами выбора пары элементов для сравнения и перестановки, а также условиями окончания процесса сортировки.

Практическая часть

Задание. Разработать алгоритмы и программы сортировки одномерных массивов для трех задач в соответствии со своим вариантом (табл. 4.1). Задачи выбираются из табл. 4.2. Каждая задача должна быть оформлена как подпрограмма.

Исходные данные для отладки программы выбрать самостоятельно.

Таблица 4.1

Варианты заданий

Вариант	Номер задачи из табл. 4.2
1	1, 7, 14
2	2, 8, 15
3	3, 9, 16
4	4, 10, 17
5	5, 11, 18
6	6, 12, 19
7	7, 13, 20
8	8, 14, 21
9	9, 15, 22
10	23, 17, 10
11	24, 18, 11
12	25, 19, 12
13	26, 20, 13
14	27, 21, 14
15	28, 22, 15
16	29, 23, 16
17	29, 1, 8
18	28, 2, 9
19	27, 3, 10
20	26, 4, 11
21	25, 5, 12
22	24, 6, 13
23	23, 7, 14
24	12, 18, 26
25	13, 19, 27
26	14, 20, 28
27	15, 21, 29

Таблица 4.2

Задачи

Номер задачи	Описание задачи
1	Упорядочить массив по неубыванию методом обменов рядом стоящих элементов с фиксированным числом просмотров, направленных слева направо

Номер задачи	Описание задачи
2	Упорядочить массив по неубыванию методом извлечения минимального элемента, просмотр элементов при извлечении минимального элемента проводить справа налево
3	Упорядочить массив по невозрастанию методом включения с просмотром элементов при выборе включаемого элемента слева направо
4	Упорядочить массив по неубыванию методом включения с просмотром элементов при выборе включаемого элемента слева направо
5	Упорядочить массив по невозрастанию методом включения с просмотром элементов при выборе включаемого элемента справа налево
6	Упорядочить массив по невозрастанию методом обмена рядом стоящих элементов с просмотром элементов справа налево
7	Упорядочить массив по невозрастанию методом извлечения максимального элемента, просмотр элементов при поиске максимального элемента проводить слева направо
8	Упорядочить массив по неубыванию методом включения, просмотр элементов при выборе включаемого элемента – справа налево
9	Получить упорядоченный по неубыванию массив методом слияния двух упорядоченных по невозрастанию массивов
10	Упорядочить массив по неубыванию методом обменов рядом стоящих элементов с просмотром элементов справа налево
11	Упорядочить массив по неубыванию методом извлечения максимального элемента, просмотр элементов при поиске максимального элемента проводить справа налево
12	Упорядочить массив по невозрастанию методом включения с просмотром элементов при поиске включаемого элемента справа налево
13	Получить упорядоченный по невозрастанию массив методом слияния двух упорядоченных по неубыванию массивов
14	Упорядочить массив по невозрастанию методом обменов рядом стоящих элементов с фиксированным числом просмотров, направленных слева направо
15	Упорядочить массив по невозрастанию методом извлечения минимального элемента, просмотр элементов при поиске минимального элемента проводить слева направо
16	Упорядочить массив по неубыванию методом включения с просмотром элементов при поиске включаемого элемента слева направо
17	Получить упорядоченный по неубыванию массив методом слияния двух массивов, один из которых упорядочен по неубыванию, а другой – по невозрастанию

Номер задачи	Описание задачи
18	Упорядочить массив по неубыванию методом обменов рядом стоящих элементов с просмотром элементов слева направо
19	Упорядочить массив по неубыванию методом извлечения максимального элемента, просмотр элементов при поиске максимального элемента проводить слева направо
20	Получить упорядоченный по невозрастанию массив методом слияния двух упорядоченных по невозрастанию массивов
21	Упорядочить массив по невозрастанию методом обменов рядом стоящих элементов за один просмотр (с возвратами) слева направо
22	Упорядочить массив по невозрастанию методом извлечения минимального элемента, просмотр элементов при поиске минимального элемента проводить справа налево
23	Получить упорядоченный по неубыванию массив методом слияния двух упорядоченных по неубыванию массивов
24	Упорядочить массив по неубыванию методом обменов рядом стоящих элементов с фиксированным числом просмотров, направленных справа налево
25	Упорядочить массив по неубыванию методом извлечения минимального элемента, просмотр элементов при поиске минимального элемента проводить слева направо
26	Упорядочить массив по невозрастанию методом обменов рядом стоящих элементов с просмотром элементов слева направо
27	Упорядочить массив по невозрастанию методом извлечения максимального элемента, просмотр элементов при поиске минимального элемента проводить справа налево
28	Упорядочить массив по невозрастанию методом обменов рядом стоящих элементов с просмотром элементов справа налево
29	Упорядочить массив по неубыванию методом обменов рядом стоящих элементов с просмотром элементов справа налево

Контрольные вопросы для защиты работ

1. Что называется ключом сортировки?
2. Какие свойства алгоритмов сортировки Вы знаете?
3. Перечислите классификацию алгоритмов сортировки.
4. В чем суть алгоритмов сортировки, использующих метод извлечения?
5. В чем суть алгоритмов сортировки, использующих метод включения?

6. В чем суть алгоритмов сортировки, использующих метод обмена?
7. В чем суть алгоритмов сортировки, использующих метод слияния?
8. В чем суть алгоритма быстрой сортировки?

Лабораторная работа № 5 **ОБРАБОТКА ЗАПИСЕЙ**

Цель работы: получить навыки работы с записями и массивами записей, а также научиться разрабатывать программы с использованием многооконного интерфейса.

Краткие теоретические сведения

Запись – это составной тип данных, состоящий из фиксированного числа элементов одного или нескольких типов. Описание типа записи начинается словом **record** и заканчивается словом **end**. Между ними заключается список элементов, называемых полями, с указанием идентификаторов полей и типа каждого поля:

```
type
    TPerson = record
        FirstName: string[20]; // имя
        LastName: string[20]; // фамилия
        BirthYear: Integer; // год рождения
    end;
```

Идентификаторы полей должны быть уникальными только в пределах записи. Допускается вложение записей друг в друга, т. е. поле записи может быть, в свою очередь, тоже записью.

Чтобы получить в программе реальную запись, нужно создать переменную соответствующего типа:

```
var
    Friend: TPerson;
```

Доступ к содержимому записи осуществляется посредством идентификаторов переменной и поля, разделенных точкой. Такая

комбинация называется составным именем. Например, чтобы получить доступ к полям записи Friend, нужно записать:

```
Friend.FirstName := 'Alexander';  
Friend.LastName := 'Ivanov';  
Friend.BirthYear := 1999;
```

Записи могут быть элементами массива. Такие массивы обрабатываются стандартными алгоритмами.

Практическая часть

Задание. Телефонный справочник жителей города оформлен в виде массива записей (табл. 5.1). Номера телефонов – шестизначные числа.

Составить программу, которая должна вводить данные справочника в модальном диалоге, выводить массив записей в виде таблицы и выполнить задание, указанное в табл. 5.2, с выводом информации в диалоговом окне.

Таблица 5.1

Вид телефонного справочника

Номер	Фамилия	Имя	Отчество	Улица	Дом	Квартира	Телефон
1	Киселев	Павел	Лукич	Победы	10	149	573590
...							

Таблица 5.2

Варианты заданий

Вариант	Задание
1	Определить, сколько жителей с фамилией «Киселев» или «Киселева» имеют телефоны, номера которых заканчиваются на '22'
2	Определить, сколько жителей с именем «Иван» и отчеством «Петрович» имеют телефоны, номера которых начинаются с цифры '5'
3	Определить, сколько жителей, живущих по улице «Советская» в доме «100», имеют телефоны
4	Определить, сколько жителей по улице «Цветочная» имеют номер телефона, заканчивающийся на 00
5	Определить, сколько жителей с фамилией "Петров", имеют номер, начинающийся на "48"

Вариант	Задание
6	Определить, сколько жителей имеют номер телефона, содержащий 3 нуля
7	Заменить номера, начинающиеся на "55", на номера, начинающиеся на "58" (например, "551440" заменить на "581440")
8	Заменить название улицы "1-я линия" на "Цветочная"
9	Заменить адрес: улица "Попова", дом "12д" на улица "Лебедева", дом "1"
10	Заменить фамилию, имя, отчество владельца телефона с номером "451440" на "Жигулев" "Семен" "Иванович"
11	Заменить фамилию, имя, отчество владельца телефона, живущего по адресу: улица "Правды", дом "6", квартира "40" на "Петров" "Степан" "Антонович"
12	Заменить номер телефона владельца, живущего по адресу: улица "Круглова", дом "17", квартира "94" на "592020"
13	Заменить адрес владельца телефона "651480" на улица "Пирогова", дом "4", квартира "17"
14	Вывести исходную таблицу в сокращенном виде: столбцы "имя" и "отчество" исключить, столбец "фамилия" дополнить инициалами с точкой
15	Вывести исходную таблицу в сокращенном виде: исключить столбцы "дом", "квартира", а столбец "улица" дополнить через запятую номером дома и квартиры
16	Вывести исходную таблицу в измененном виде: столбец с номером телефона поставить перед фамилией владельца.
17	В номерах телефонов, начинающихся на "48", заменить первые две цифры на "57"
18	Определить количество жителей 1961-го года рождения
19	Определить количество жителей с фамилией "Киселев"
20	Определить год рождения участника с фамилией, именем, отчеством "Петров" "Игорь" "Иванович"

Контрольные вопросы для защиты работ

1. Что такое поле записи?
2. Какие типы данных используются для описания полей записи?
3. Как обратиться к полю записи?
4. Для чего используется оператор with?
5. Сформировать запись определенной структуры.

6. Как отобразить информацию из массива записей, удовлетворяющих определенному условию?
7. Как добавить запись в определенное место массива записей?
8. Как обменивать местами записи в массиве записей?

Лабораторная работа № 6

ОБРАБОТКА ФАЙЛОВ

Цель работы: получить навыки разработки и реализации алгоритмов работы с файлами на языке Object Pascal.

Краткие теоретические сведения

Понятие файла

С точки зрения пользователя **файл** – это именованная область данных на диске или любом другом внешнем носителе. В программе файл представляется как последовательность элементов некоторого типа. Так как размер одного файла может превышать объем всей оперативной памяти компьютера, доступ к его элементам выполняется последовательно с помощью процедур чтения и записи.

Для файла существует понятие текущей позиции. Она показывает номер элемента, который будет прочитан или записан при очередном обращении к файлу. Чтение-запись каждого элемента продвигает текущую позицию на единицу вперед. Для большинства файлов можно менять текущую позицию чтения-записи, выполняя прямой доступ к его элементам.

В зависимости от типа элементов различают три вида файла:

- файл из элементов фиксированного размера; элементами такого файла чаще всего являются записи;
- файл из элементов переменного размера (нетипизированный файл); такой файл рассматривается просто как последовательность байтов;
- текстовый файл; элементами такого файла являются текстовые строки.

Для работы с файлом в программе объявляется файловая переменная. В файловой переменной запоминаются имя файла, режим доступа (например, только чтение), другие атрибуты.

Для работы с файлом, состоящим из типовых элементов, переменная объявляется с помощью словосочетания `file of`, после которого записывается тип элемента:

```
Var  
    F: file of TPerson;
```

К моменту такого объявления тип `TPerson` должен быть уже описан.

Стандартные подпрограммы управления файлами

Для обработки файлов в языке Delphi имеется специальный набор процедур и функций.

AssignFile(var F; FileName: string) – связывает файловую переменную `F` и файл, имя которого указано в `FileName`.

Reset(var F [: File; RecSize: Word]) – открывает существующий файл. Если открывается нетипизированный файл, то `RecSize` задает размер элемента файла.

Rewrite(var F [: File; RecSize: Word]) – создает и открывает новый файл.

Append(var F: TextFile) – открывает текстовый файл для добавления текста.

Read(F, V1 [, V2, ..., Vn]) – начиная с текущей позиции, читает из типизированного файла подряд расположенные элементы в переменные `V1, V2, ..., Vn`.

Read(var F: TextFile; V1 [, V2, ..., Vn]) – начиная с текущей позиции, читает из текстового файла символы или строки в переменные `V1, V2, ..., Vn`.

Write(F, V1 [, V2, ..., Vn]) – начиная с текущей позиции, записывает в типизированный файл значения `V1, V2, ..., Vn`.

Write(var F: TextFile; V1 [, V2, ..., Vn]) – начиная с текущей позиции указателя чтения-записи, записывает в текстовый файл значения `V1, V2, ..., Vn`.

CloseFile(var F) – закрывает ранее открытый файл.

Seek(var F; NumRec: Longint) – устанавливает позицию чтения-записи на элемент с номером `NumRec`; `F` – типизированный или нетипизированный файл.

Eof(var F): Boolean – возвращает булевское значение True, если текущая позиция чтения-записи находится сразу за последним элементом, и False – в противном случае.

Последовательность действий для работы с файлами

Приступая к работе с файлом, нужно первым делом вызвать процедуру AssignFile, чтобы файловой переменной поставить в соответствие имя файла на диске:

```
AssignFile(F, 'MyBase.dat');
```

В результате этого действия поля файловой переменной F инициализируются начальными значениями. При этом в поле имени файла заносится строка 'MyBase.dat'.

Если файл открывается для записи, используется процедура Rewrite(F), если для чтения – Reset(F).

Далее обычно организовывается цикл для записи или чтения данных. Для записи данных используется процедура Write, а для чтения – Read. При работе с файлами первый параметр этих процедур показывает, куда происходит вывод данных или откуда происходит ввод данных.

После работы файл должен быть закрыт процедурой CloseFile(F).

Практическая часть

Задание. Разработать информационную систему (ИС) по определенной теме с возможностью сохранения данных в файле.

В каждом варианте задания (табл. 6.1) необходимо обеспечить выполнение следующих действий (операций):

- ввод информации в файл;
- просмотр на экране содержимого файла;
- поиск необходимой информации;
- корректировка содержимого файла.

Отладку программ производить с самостоятельно выбираемым набором данных. Число записей файла – произвольное, но не менее 15.

При организации поиска использовать следующие способы:

- для варианта а) – поиск на совпадение,
- для варианта б) – поиск на вхождение,
- для варианта в) – поиск на принадлежность к диапазону.

Способы корректировки файла:

1. Вставка новой записи после указанного номера существующей записи.
2. Удаление записи с указанным номером.
3. Замена записи с указанным номером.
4. Обмен местами двух записей с указанными номерами.
5. Вставка записи в начало файла.
6. Добавление записи в конец файла.
7. Удаление записи после указанного номера.
8. Восстановление только что удаленной записи.
9. Замена любого выбранного пользователем поля в записи с заданным номером.

Таблица 6.1

Обработка файлов

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
1	Ведомость для начисления заработной платы	1. Цех, участок 2. Ф.И.О. 3. Объем выполненной работы 4. Расценки на единицу продукции 5. Начисленный заработок	а) по номеру б) по фамилии в) по величине заработка
2	Список оборудования	1. Кафедра 2. Наименование 3. Количество 4. Стоимость единицы продукции 5. Общая стоимость	а) по количеству б) по наименованию в) по стоимости
3	Библиотечный каталог	1. УДК (универсальный десятичный код) 2. Ф.И.О. автора 3. Наименование 4. Стоимость 5. Количество	а) по количеству б) по фамилии автора в) по стоимости

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
4	Железнодорожное расписание	1. Номер поезда 2. Наименование 3. Время прихода 4. Время отправления 5. Категория (скорый, пассажирский)	а) по номеру поезда б) по пункту назначения в) по времени отправления
5	Протокол соревнований	1. Организация 2. Ф.И.О. 3. Год рождения 4. Результат 5. Занятое место	а) по занятому месту б) по фамилии в) по результату
6	Записная книжка-ежедневник	1. День недели 2. Время суток 3. Мероприятие 4. Ф.И.О. 5. Телефон	а) по телефону б) по мероприятию в) по времени
7	Географический справочник	1. Страна 2. Площадь, млн км ² 3. Население, млн человек 4. Континет 5. Столица	а) по континенту б) по названию в) по площади
8	Телефонный справочник	1. Ф.И.О. 2. Улица 3. Номер дома 4. Организация 5. Телефон	а) по номеру дома б) по телефону в) по названию организации
9	Результаты обмена емкостей нефтебазы	1. Типовое название емкости 2. Дата измерения 3. Уровень, м 4. Вид топлива 5. Объем топлива	а) по виду топлива б) по объему топлива в) по дате измерения
10	Справочник автоинспектора	1. Марка автомобиля 2. Номер 3. Цвет 4. Год выпуска 5. Пробег, км	а) по году выпуска б) по цвету в) по марке автомобиля

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
11	Записная книжка продавца	1. Дата 2. Вид товара 3. Количество 4. Цена за единицу 5. Общая стоимость	а) по дате б) по виду товара в) по цене
12	Памятка агронома-химика	1. Название культуры 2. Название удобрения 3. Норма внесения на 100 м ² 4. Дата внесения (по месяцам) 5. Содержание, единица объема (в одном метре)	а) по дате внесения б) по виду удобрений в) по названию культуры
13	Памятка Шерлока Холмса	1. Фамилия субъекта 2. Год рождения 3. Вид правонарушения 4. Дата нарушения 5. Сумма вознаграждения	а) по дате нарушения б) по виду правонарушения в) по фамилии
14	Протокол технического эксперимента	1. Дата проведения эксперимента 2. Вид оборудования 3. Время начала эксперимента 4. Время окончания эксперимента	а) по дате проведения б) по виду оборудования в) по затратам
15	Справочник кинолога	1. Порода собаки 2. Отец 3. Мать 4. Дата рождения 5. Рост	а) по породе б) по дате рождения в) по отцу
16	Памятка туриста	1. Район маршрута 2. Вид туризма 3. Категория (от 1 до 6) 4. Стоимость проезда 5. Вес снаряжения на одного участника	а) по категории б) по району в) по виду туризма

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
17	Памятка дачника-овощевода	1. Вид овощей 2. Название сорта 3. Дата посадки 4. Дата уборки урожая 5. Урожай, кг на м ²	а) по дате уборки б) по названию сорта в) по урожаю
18	Памятка преподавателя	1. Название дисциплины 2. Объем лекций, ч 3. Объем лабораторных занятий, ч 4. Вид контроля – зачет, экзамен 5. Количество студентов	а) по виду контроля б) по дисциплине в) по объему занятий
19	Дневник метеонаблюдений	1. Дата 2. Температура 3. Давление 4. Облачность (ясно, слабая, сильная, дождь) 5. Направление ветра (азимут)	а) по дате б) по температуре в) по направлению ветра
20	Справочник по транзисторам	1. Тип транзистора 2. Напряжение питания 3. Допустимый ток 4. Коэффициент усиления 5. Стоимость	а) по типу транзистора б) по напряжению питания в) по допустимому току
21	Справочник по оборудованию	1. Наименование 2. Страна-изготовитель 3. Стоимость 4. Вес, кг 5. Объем, дм ³	а) по наименованию б) по стране-изготовителю в) по стоимости
22	Великие люди	1. Ф.И.О. 2. Область деятельности (ученый, полководец, общественный деятель, поэт, художник и т. д.) 3. Год рождения 4. Страна 5. Продолжительность жизни	а) по названию страны б) по фамилии в) по году рождения

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
23	Памятка любителя музыки	1. Название группы 2. Страна 3. Фамилия руководителя 4. Количество записанных дисков 5. Общий тираж дисков	а) по названию группы б) по фамилии руководителя в) по количеству дисков
24	Великие даты	1. Дата 2. Страна 3. Вид события (война, революция, образование государства и т. д.) 4. Фамилия видного деятеля 5. Примерное число жертв	а) по названию страны б) по виду события в) по дате
25	Памятка альпиниста	1. Название вершины 2. Высота 3. Страна расположения 4. Год покорения 5. Количество восхождений	а) по году покорения б) по стране расположения в) по высоте
26	История персональных ЭВМ	1. Тип микропроцессора 2. Дата появления 3. Фирма-изготовитель 4. Тактовая частота 5. Объем адресуемой памяти, Кб	а) по тактовой частоте б) по типу микропроцессора в) по объему памяти
27	Живая планета	1. Наименование животного 2. Рост (длина), м 3. Вес (средний), кг 4. Скорость передвижения, км/ч 5. Континент (океан)	а) по континенту б) по названию в) по весу
28	Медицинская карта	1. Ф.И.О. 2. Год рождения 3. Рост 4. Вес 5. Группа крови	а) по фамилии б) по группе крови в) по возрасту

Вариант	Область применения информационной системы	Наименование полей записи	Поля для поиска информации
29	Химические элементы	1. Наименование 2. Атомный вес 3. Валентность 4. Плотность 5. Цвет	а) по наименованию б) по цвету в) по атомному весу
30	Материалы	1. Наименование 2. Плотность 3. Агрегатное состояние (твердое, жидкое, газообразное) 4. Электропроводность 5. Модуль упругости	а) по наименованию б) по агрегатному состоянию в) по плотности

Контрольные вопросы для защиты работ

1. Какие типы файлов Вы знаете?
2. Что такое текстовый файл?
3. Что такое структурированный файл?
4. Как описывается файловая переменная?
5. Как создать файл?
6. Как открыть файл только для чтения?
7. Как добавить информацию в конец файла?
8. Почему при окончании работы с файлом его обязательно необходимо закрыть?
9. Какие стандартные функции обработки файлов Вы знаете?
10. Что такое конец файла?

Лабораторная работа № 7

ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

Цель работы: изучить методы создания и обработки динамических структур данных на основе связанных списков.

Краткие теоретические сведения

Все команды и данные программы во время ее выполнения размещаются в определенных ячейках оперативной памяти. При этом часть данных размещается в ячейки памяти еще на этапе компиляции и в процессе работы программы их адреса относительно начала программы не изменяются. Такое размещение данных и команд называется **статическим** и соответствующие этим данным переменные называются **статическими переменными**.

В языке Object Pascal возможна также организация **динамического** размещения данных, при котором под некоторые данные и программы память выделяется непосредственно во время выполнения по мере надобности, а после решения требуемой задачи память освобождается для других данных. Соответствующие таким данным переменные называются **динамическими переменными**.

Для организации динамического распределения памяти используются переменные специального типа – **указатели**, которые обеспечивают работу непосредственно с адресами ячеек памяти.

Указатель описывается следующим образом:

Туре
 $P_k = \wedge \langle \text{тип} \rangle;$

Списки

В практике программирования довольно часто встречается задача обработки набора однотипных данных, количество которых меняется в зависимости от ситуации. Их надо разместить в оперативной памяти и по мере надобности с ними работать: добавлять новые данные в список, освобождаться от старых данных, находить нужную информацию. При решении подобных задач программисты вводят понятие списка.

Список – это последовательность однотипных элементов (данных), с которыми надо работать в оперативной памяти: добавлять новые элементы в группу, удалять использованные, сортировать, находить нужные. В процессе работы список может возрасти и уменьшиться.

Связанный список – это структура данных в виде списка, элементы которого связаны друг с другом с помощью указателей.

Как правило, каждый элемент этого списка имеет тип record. Данные хранятся в информационном поле записи. Для адресации

элементов в структуру включается указатель (адресное поле) на область размещения следующего элемента. Такой список называют **однаправленным (односвязным)**. Если добавить в каждый элемент ссылку на предыдущий, получится двунаправленный список (двусвязный).

```

Type TData=integer;//Описание типа информационной части
TNode1=^Node1;//Описание типа указателя на элемент
Node1=record//односвязных списков
data:TData;
next:TNode;
end;
TNode2=^Node2;//Описание типа указателя на элемент
Node2=record//двусвязных списков
data:TData;
next:TNode2;
prev:TNode2;
end;

```

При организации списка сперва создается элемент, формируется его информационная часть, а затем он связывается с предыдущим. Если на момент создания элемента список пуст, то в адресное поле, указывающее на следующий элемент, заносится nil. При удалении элемента из списка сперва перезаписываются связи между элементами, а затем освобождается память, выделенная под элемент списка, вызовом процедуры **dispose**.

Практическая часть

Задание. Написать программу для создания и просмотра динамической структуры на основе связанных списков, добавления элементов в список и решения указанной задачи (табл. 7.1).

Таблица 7.1

Обработка списков

Вариант	Задание
1	Создать список из случайных целых чисел, лежащих в диапазоне от -50 до +50 и преобразовать его в два списка. Первый должен содержать только положительные числа, а второй – только отрицательные. Порядок следования чисел должен быть сохранен

Вариант	Задание
2	Создать список из случайных целых чисел и удалить из него записи с четными числами
3	Создать список из случайных положительных и отрицательных целых чисел (от -10 до 10) и удалить из него отрицательные элементы
4	Создать список из случайных целых чисел и поменять местами крайние элементы
5	Создать список из случайных целых чисел и удалить элементы, заканчивающиеся на цифру 5
6	Создать список из случайных целых чисел и поменять местами элементы, содержащие максимальное и минимальное значения
7	Создать список из случайных целых чисел. Перенести в другой список все элементы, находящиеся между вершиной и элементом с максимальным значением
8	Создать список из случайных целых чисел. Перенести в другой список все элементы, находящиеся между вершиной и элементом с минимальным значением
9	Создать список из случайных чисел, определить количество элементов, находящихся между минимальным и максимальным элементами, и удалить их
10	Создать список из случайных чисел и определить количество элементов, имеющих значения меньше среднего значения от всех элементов, и удалить эти элементы
11	Создать список из случайных чисел, вычислить среднее арифметическое и заменить им первый элемент
12	Создать список из случайных целых чисел, разделить его на два: в первый поместить все четные, а во второй – нечетные числа
13	Создать список из случайных целых чисел в диапазоне от 1 до 10, определить наиболее часто встречающееся число и удалить его
14	Создать список из случайных целых чисел и удалить из него каждый второй элемент
15	Создать список из случайных целых чисел и удалить из него каждый нечетный элемент
16	Создать список из случайных целых чисел. Найти минимальный элемент и сделать его первым
17	Создать два списка из случайных целых чисел. В первом найти максимальный элемент и за ним вставить элементы второго
18	Создать список из случайных целых чисел. Удалить из списка все элементы, находящиеся между максимальным и минимальным элементами

Вариант	Задание
19	Упорядочить элементы списка случайных целых чисел в порядке возрастания
20	Создать список из случайных целых чисел. Удалить из списка все элементы, находящиеся до максимального элемента
21	Создать список из случайных целых чисел. Удалить из списка все элементы, находящиеся после минимального элемента
22	Создать список из случайных целых чисел. Из элементов, расположенных между максимальным и минимальным элементами, создать второй список, а из остальных – третий
23	Создать список из случайных положительных и отрицательных целых чисел. Образовать из него два списка, первый должен содержать отрицательные числа, а второй – положительные
24	Создать список из случайных целых чисел. Удалить из списка все элементы, находящиеся после максимального элемента
25	Создать два списка из случайных целых чисел. Вместо элементов первого списка, заключенных между максимальным и минимальным элементами, вставить второй список
26	Создать список из случайных целых чисел. Удалить из списка элементы с повторяющимися более одного раза значениями
27	Создать список из случайных целых чисел и удалить все элементы, кратные 5
28	Создать список из случайных целых чисел. Удалить из списка все элементы больше среднего арифметического
29	Создать список из случайных чисел. Преобразовать его в кольцо. Предусмотреть возможность движения по кольцу в обе стороны с отображением места положения текущего элемента
30	Создать список из случайных целых чисел. Удалить из списка все элементы, находящиеся между максимальным и минимальным элементами

Контрольные вопросы для защиты работ

1. Что такое связанный список?
2. В чем отличие двусвязанного списка?
3. В чем преимущество использования списков?
4. В чем недостатки использования списков?
5. Что такое динамический массив?
6. Что такое указатель?
7. Как обратиться к элементам динамического массива через указатель?

8. Как добавить элемент в начало списка?
9. Как удалить элемент из произвольного места?
10. Что из себя представляет элемент списка?

Лабораторная работа № 8 **ВЫЧИСЛЕНИЕ ИНТЕГРАЛОВ**

Цель работы: изучить численные методы нахождения интегралов.

Краткие теоретические сведения

При решении интегралов стандартными методами высшей математики необходимо найти первообразную подынтегральной функции. Однако во многих случаях найти первообразную невозможно. В таких случаях приходится применять методы численного интегрирования.

Основной принцип построения всех приближенных формул численного интегрирования состоит в том, что интервал интегрирования разбивается на множество меньших отрезков, внутри которых подынтегральная кривая $y = f(x)$ заменяется с некоторой степенью точности более простыми функциями, интегралы от которых можно вычислить.

Если разбить интервал интегрирования на n равных частей, каждая длиной $h = \frac{b-a}{n}$, можно получить следующие основные формулы численного интегрирования:

Метод «левых» прямоугольников:

$$S = \int_a^b f(x)dx \approx h \sum_{i=0}^{n-1} f(x_i).$$

Метод «правых» прямоугольников:

$$S = \int_a^b f(x)dx \approx h \sum_{i=1}^n f(x_i).$$

Метод «средних» прямоугольников:

$$S = \int_a^b f(x)dx \approx h \sum_{i=1}^n f\left(\frac{x_i + x_{i-1}}{2}\right).$$

Метод трапеций:

$$S = \int_a^b f(x)dx \approx h \left(\frac{f(x_0)}{2} + \frac{f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i) \right).$$

Метод парабол (Симпсона):

$$\int_a^b f(x)dx \approx \frac{h}{3} (Y_{\text{кр}} + 4Y_{\text{неч}} + 2Y_{\text{чет}}).$$

Практическая часть

Задание. Разработать алгоритм и программу вычисления определенного интеграла численным методом, указанным в табл. 8.2. Вычислить три интеграла (табл. 8.1) в соответствии с вариантом.

Результаты работы программы проверить решением заданных интегралов в системе компьютерной математики.

Таблица 8.1

Определенные интегралы

Номер варианта	Интеграл	Номер варианта	Интеграл	Номер варианта	Интеграл
1	$\int_{5.1}^{8.3} \frac{xdx}{x+2.5}$	2	$\int_{0.5}^{\pi/2} \frac{\cos^2 x}{\sin x} dx$	3	$\int_3^{6.1} \frac{x+6.25}{(x+1.5)^2} dx$
4	$\int_3^{4.5} (5-x^2)^{0.5} dx$	5	$\int_{0.1}^{1.5} \frac{dx}{\sin x \cos x}$	6	$\int_{3.1}^{4.6} \frac{xdx}{\sqrt{x^4-x}}$
7	$\int_{0.5}^{6.1} \frac{dx}{x^2(x+1.3)}$	8	$\int_{0.1}^{1.6} \frac{2-\sin x}{1+\cos x} dx$	9	$\int_{-1.3}^3 \frac{x^3 dx}{(9+x^2)^{3/2}}$
10	$\int_{1.1}^{3.6} \frac{dx}{(x+1)\sqrt{x+x^4}}$	11	$\int_{0.1}^{4.2} \frac{dx}{x^2+3x}$	12	$\int_2^{4.5} \frac{dx}{\sqrt{(x+2)(x+0.5)}}$
13	$\int_{2.5}^{7.5} \frac{dx}{x(x+1.6)^2}$	14	$\int_{2.6}^{7.1} \frac{(x^2+3)^{3/2}}{x^3} dx$	15	$\int_{0.56}^{1.5} \frac{dx}{\cos^3 x \sin^3 x}$
16	$\int_{-1.5}^{1.5} \frac{x+2.5}{x+6.1} dx$	17	$\int_2^{6.5} \frac{xdx}{x^2+6x+1}$	18	$\int_{0.5}^{\pi} \sin^4 x dx$
19	$\int_0^{2.1} \frac{dx}{(x+3)\sqrt{9-x^2}}$	20	$\int_{\pi/2}^{3.2} \cos^3 x dx$	21	$\int_{0.8}^{\pi/2} \frac{\sin x dx}{\sqrt{3+2\sin x}}$

Окончание табл. 8.1

Номер варианта	Интеграл	Номер варианта	Интеграл	Номер варианта	Интеграл
22	$\int_{-1.23}^3 x \frac{x+1.5}{x+0.6} dx$	23	$\int_{-1.5}^{4.5} \frac{x^2 dx}{x^2 - 3x + 10}$	24	$\int_7^9 \sqrt{\frac{x-6}{(x+7)(x-3)}} dx$
25	$\int_{0.1}^3 \frac{dx}{\sin^3 x}$	26	$\int_{7.1}^{9.9} x^3 (7+x^2)^{0.5} dx$	27	$\int_{3.1}^{6.9} \frac{x^3 dx}{2x^2 + 6x + 1}$
28	$\int_{0.6}^{1.1} \frac{\sin^4 x}{\cos x} dx$	29	$\int_{0.1}^{1.5} \frac{\cos^3 x}{\sin^3 x} dx$	30	$\int_{1.1}^{3.6} \left(\frac{x+3}{x+4} \right)^2 dx$

Таблица 8.2

Варианты заданий

Вариант	Интеграл (из табл. 8.1)	Численный метод
1	1, 5, 10	Метод прямоугольников
2	2, 6, 11	Метод трапеций
3	3, 7, 12	Метод Симпсона
4	4, 8, 13	Метод прямоугольников
5	5, 9, 14	Метод трапеций
6	6, 10, 15	Метод Симпсона
7	7, 11, 16	Метод прямоугольников
8	8, 12, 17	Метод трапеций
9	9, 13, 18	Метод Симпсона
10	10, 14, 19	Метод прямоугольников
11	11, 15, 20	Метод трапеций
12	12, 16, 21	Метод Симпсона
13	13, 17, 22	Метод прямоугольников
14	14, 18, 23	Метод трапеций
15	15, 19, 24	Метод Симпсона
16	16, 20, 25	Метод прямоугольников
17	17, 21, 26	Метод трапеций
18	18, 22, 27	Метод Симпсона
19	19, 23, 28	Метод прямоугольников
20	20, 24, 29	Метод трапеций
21	21, 25, 30	Метод Симпсона
22	10, 14, 19	Метод прямоугольников
23	11, 15, 20	Метод трапеций

Прямой ход приводит систему (9.1) к эквивалентной ей системе вида (9.2):

$$\begin{cases} x_1 + \tilde{a}_{12}x_2 + \dots + \tilde{a}_{1n}x_n = \tilde{b}_1; \\ x_2 + \dots + \tilde{a}_{2n}x_n = \tilde{b}_2; \\ \dots\dots\dots \\ \tilde{a}_{nn}x_n = \tilde{b}_n. \end{cases} \quad (9.2)$$

Для этого сначала первое неизвестное исключают из второго и последующих уравнений системы, затем второе неизвестное исключают из третьего и последующих уравнений и т. д. Таким образом, в последнем уравнении остается только одно неизвестное. Для реализации прямого хода используют следующие известные правила:

- любое уравнение системы можно умножить на постоянный коэффициент;
- можно сложить два любых уравнения системы и результат записать вместо одного из этих уравнений.

Переход от системы (9.1) к системе (9.2) возможен при выполнении следующих преобразований. Пусть $a_{11} \neq 0$ (если это не так, то можно поменять местами два уравнения системы). Разделим все члены первого уравнения системы (9.1) на a_{11} , все члены второго уравнения – на a_{21} , третьего – на a_{31} и т. д. Если какой-то из этих коэффициентов равен нулю, то соответствующее уравнение не преобразовывается. Затем вычтем из второго, третьего, ..., n -го уравнения соответствующие части первого, получим:

$$\begin{cases} x_1 + a'_{12}x_2 + \dots + a'_{1n}x_n = b'_1; \\ a'_{22}x_2 + \dots + a'_{2n}x_n = b'_2; \\ \dots\dots\dots \\ a'_{n2}x_2 + \dots + a'_{nn}x_n = b'_n. \end{cases}$$

Первое уравнение оставим без изменений, а оставшиеся преобразуем аналогично. Так, последовательно систему (9.1) приводим к виду (9.2).

Обратный ход: последовательно вычисляют значения всех неизвестных, начиная с последнего.

Отсюда найдем:

$$x_2 = \frac{-c_2 x_3 + d_2 - a_2 B_1}{a_2 A_1 + b_2},$$

или

$$x_2 = A_2 x_3 + B_2;$$

$$A_2 = -\frac{c_2}{e_2}; \quad B_2 = \frac{d_2 - a_2 B_1}{e_2}; \quad e_2 = a_2 A_1 + b_2.$$

Аналогично можно вычислить прогоночные коэффициенты для любого номера i :

$$A_i = -\frac{c_i}{e_i}; \quad B_i = \frac{d_i - a_i B_{i-1}}{e_i}; \quad e_i = a_i A_{i-1} + b_i; \quad i = 2, 3, \dots, n-1. \quad (9.6)$$

Обратная прогонка состоит в последовательном вычислении неизвестных x_i . Сначала необходимо найти x_n . Для этого используют выражение (9.4) при $i = n-1$ и последнее уравнение системы (9.3). Исключая x_{i-1} , находят:

$$x_n = \frac{d_n - a_n B_{n-1}}{a_n A_{n-1} + b_n}.$$

Далее, используя формулы (9.4) и выражения для прогоночных коэффициентов (9.5), (9.6), последовательно вычисляют все неизвестные $x_{n-1}, x_{n-2}, \dots, x_1$.

Метод итераций (метод последовательных приближений)

Приближенные методы решения СЛАУ позволяют получать значения корней системы с заданной точностью в виде предела последовательности некоторых векторов. Процесс построения такой последовательности называется итерационным (повторяющимся). Эффективность применения приближенных методов зависит от выбора начального приближения и быстроты сходимости процесса.

Рассмотрим СЛАУ (7), состоящую из n уравнений с n неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1; \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2; \\ \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n. \end{cases} \quad (9.7)$$

Предполагаем, что диагональные коэффициенты $a_{ii} \neq 0$ ($i = 1, 2, \dots, n$); выразим x_1 через первое уравнение системы, x_2 — через второе уравнение и т. д. В результате получим систему (9.8), эквивалентную системе (9.7):

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3 - \dots - \frac{a_{1n}}{a_{11}}x_n; \\ x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 - \dots - \frac{a_{2n}}{a_{22}}x_n; \\ \dots \dots \dots \\ x_n = \frac{b_n}{a_{nn}} - \frac{a_{n1}}{a_{nn}}x_1 - \frac{a_{n2}}{a_{nn}}x_2 - \dots - \frac{a_{n,n-1}}{a_{nn}}x_{n-1}. \end{cases} \quad (9.8)$$

Обозначим $\frac{b_i}{a_{ii}} = \beta_i$; $-\frac{a_{ij}}{a_{ii}} = \alpha_{ij}$, где $i = 1, 2, \dots, n$; $j = 1, 2, \dots, n$.

Тогда система (9.8) запишется таким образом:

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \dots + \alpha_{1n}x_n; \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + \dots + \alpha_{2n}x_n; \\ \dots \dots \dots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \dots + \alpha_{n,n-1}x_{n-1}. \end{cases} \quad (9.9)$$

Система (9.9) называется системой, приведенной к **нормальному виду**.

За начальное (нулевое) приближение к точному решению СЛАУ примем столбец свободных членов:

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \dots \\ x_n^{(0)} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{pmatrix} \text{ — нулевое приближение.}$$

Пояснение к обозначению: $x_j^{(i)}$ – верхний индекс (i) обозначает номер итерации (приближения); нижний j – номер неизвестного в уравнении.

Далее находим последовательные приближения к точному решению СЛАУ.

1-я итерация:

$$\begin{cases} x_1^{(1)} = \beta_1 + \alpha_{12}x_2^{(0)} + \alpha_{13}x_3^{(0)} + \dots + \alpha_{1n}x_n^{(0)}; \\ x_2^{(1)} = \beta_2 + \alpha_{21}x_1^{(0)} + \alpha_{23}x_3^{(0)} + \dots + \alpha_{2n}x_n^{(0)}; \\ \dots \\ x_n^{(1)} = \beta_n + \alpha_{n1}x_1^{(0)} + \alpha_{n2}x_2^{(0)} + \dots + \alpha_{n,n-1}x_{n-1}^{(0)}. \end{cases} \quad x^{(1)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \dots \\ x_n^{(1)} \end{pmatrix}.$$

2-я итерация:

$$\begin{cases} x_1^{(2)} = \beta_1 + \alpha_{12}x_2^{(1)} + \alpha_{13}x_3^{(1)} + \dots + \alpha_{1n}x_n^{(1)}; \\ x_2^{(2)} = \beta_2 + \alpha_{21}x_1^{(1)} + \alpha_{23}x_3^{(1)} + \dots + \alpha_{2n}x_n^{(1)}; \\ \dots \\ x_n^{(2)} = \beta_n + \alpha_{n1}x_1^{(1)} + \alpha_{n2}x_2^{(1)} + \dots + \alpha_{n,n-1}x_{n-1}^{(1)}. \end{cases} \quad x^{(2)} = \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \\ \dots \\ x_n^{(2)} \end{pmatrix}.$$

И так далее.

(k + 1)-я итерация:

$$\begin{cases} x_1^{(k+1)} = \beta_1 + \alpha_{12}x_2^{(k)} + \alpha_{13}x_3^{(k)} + \dots + \alpha_{1n}x_n^{(k)}; \\ x_2^{(k+1)} = \beta_2 + \alpha_{21}x_1^{(k)} + \alpha_{23}x_3^{(k)} + \dots + \alpha_{2n}x_n^{(k)}; \\ \dots \\ x_n^{(k+1)} = \beta_n + \alpha_{n1}x_1^{(k)} + \alpha_{n2}x_2^{(k)} + \dots + \alpha_{n,n-1}x_{n-1}^{(k)}. \end{cases} \quad x^{(k+1)} = \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \dots \\ x_n^{(k+1)} \end{pmatrix}.$$

Если выполнены условия сходимости итерационного процесса (см. п. 2.4.3), то можно доказать, что последовательность $x^{(0)}, x^{(1)}, \dots, x^{(k)}, \dots$ сходится к точному решению системы (9.2).

Метод Зейделя

Метод Зейделя представляет собой модификацию метода последовательных приближений. Процесс Зейделя сходится к точному решению СЛАУ быстрее метода итераций.

Начальное приближение выбираем аналогично методу итераций:

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \dots \\ x_n^{(0)} \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{pmatrix}.$$

В методе Зейделя при вычислении $(k + 1)$ -го приближения неизвестного x_i ($i > 1$) учитываются уже найденные ранее $(k + 1)$ -е приближения неизвестных x_1, x_2, \dots, x_{i-1} .

1-я итерация:

$$\begin{cases} x_1^{(1)} = \beta_1 + \alpha_{12}x_2^{(0)} + \alpha_{13}x_3^{(0)} + \dots + \alpha_{1n}x_n^{(0)}; \\ x_2^{(1)} = \beta_2 + \alpha_{21}x_1^{(1)} + \alpha_{23}x_3^{(0)} + \dots + \alpha_{2n}x_n^{(0)}; \\ \dots \\ x_n^{(1)} = \beta_n + \alpha_{n1}x_1^{(1)} + \alpha_{n2}x_2^{(1)} + \dots + \alpha_{n,n-1}x_{n-1}^{(1)}. \end{cases} \quad x^{(1)} = \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \dots \\ x_n^{(1)} \end{pmatrix}.$$

К моменту нахождения $x_2^{(1)}$ уже найдено $x_1^{(1)}$, его и используем при вычислении, а не $x_1^{(0)}$. К моменту нахождения $x_n^{(1)}$ уже найдены $x_1^{(1)}, x_2^{(1)}, \dots, x_{n-1}^{(1)}$, их и используем вместо $x_1^{(0)}, x_2^{(0)}, \dots, x_{n-1}^{(0)}$.

И так далее.

$(k + 1)$ -я итерация:

$$\begin{cases} x_1^{(k+1)} = \beta_1 + \alpha_{12}x_2^{(k)} + \alpha_{13}x_3^{(k)} + \dots + \alpha_{1n}x_n^{(k)}; \\ x_2^{(k+1)} = \beta_2 + \alpha_{21}x_1^{(k+1)} + \alpha_{23}x_3^{(k)} + \dots + \alpha_{2n}x_n^{(k)}; \\ \dots \\ x_n^{(k+1)} = \beta_n + \alpha_{n1}x_1^{(k+1)} + \alpha_{n2}x_2^{(k+1)} + \dots + \alpha_{n,n-1}x_{n-1}^{(k+1)}. \end{cases} \quad x^{(k+1)} = \begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \dots \\ x_n^{(k+1)} \end{pmatrix}.$$

И так далее.

Таким образом, на каждой итерации используется самая свежая информация, только что полученная на этой же итерации.

Практическая часть

Задание. Разработать алгоритм и программу решения СЛАУ методом, указанным в табл. 9.2. Решить систему уравнений в соответствии с вариантом (табл. 9.1).

Результаты работы программы проверить подстановкой найденного решения в исходную систему для получения тождества.

Таблица 9.1

Система линейных алгебраических уравнений

Вариант	Система уравнений	Вариант	Система уравнений
1	$\begin{cases} 2x_1 + x_2 + 3x_3 = 7 \\ 2x_1 + 3x_2 + x_3 = 1 \\ 3x_1 + 2x_2 + x_3 = 6 \end{cases}$	2	$\begin{cases} 2x_1 - x_2 + 2x_3 = 3 \\ x_1 + x_2 + 2x_3 = -4 \\ 4x_1 + x_2 + 4x_3 = -3 \end{cases}$
3	$\begin{cases} 3x_1 - x_2 + x_3 = 12 \\ x_1 + 2x_2 + 4x_3 = 6 \\ 5x_1 + x_2 + 2x_3 = 3 \end{cases}$	4	$\begin{cases} 2x_1 - x_2 + 3x_3 = -4 \\ x_1 + 3x_2 - x_3 = 11 \\ x_1 - 2x_2 + 2x_3 = -7 \end{cases}$
5	$\begin{cases} 3x_1 - 2x_2 + 4x_3 = 12 \\ 3x_1 + 4x_2 - 2x_3 = 6 \\ 2x_1 - x_2 - x_3 = -9 \end{cases}$	6	$\begin{cases} 8x_1 + 3x_2 - 6x_3 = -4 \\ x_1 + x_2 - x_3 = 2 \\ 4x_1 + x_2 - 3x_3 = -5 \end{cases}$
7	$\begin{cases} 4x_1 + x_2 - 3x_3 = 9 \\ x_1 + x_2 - x_3 = -2 \\ 8x_1 + 3x_2 - 6x_3 = 12 \end{cases}$	8	$\begin{cases} 3x_1 - 2x_2 + 4x_3 = 21 \\ 3x_1 + 4x_2 - 2x_3 = 9 \\ 2x_1 - x_2 - x_3 = 10 \end{cases}$
9	$\begin{cases} 3x_1 - 2x_2 - 5x_3 = 5 \\ 2x_1 + 3x_2 - 4x_3 = 12 \\ x_1 - 2x_2 + 3x_3 = -1 \end{cases}$	10	$\begin{cases} 4x_1 + x_2 + 4x_3 = 19 \\ 2x_1 - x_2 + 2x_3 = 11 \\ x_1 + x_2 + 2x_3 = 8 \end{cases}$
11	$\begin{cases} 2x_1 - x_2 + 2x_3 = 0 \\ 4x_1 + x_2 + 4x_3 = 6 \\ x_1 + x_2 + 2x_3 = 4 \end{cases}$	12	$\begin{cases} 2x_1 - x_2 - 3x_3 = 0 \\ 3x_1 + 4x_2 + 2x_3 = 1 \\ x_1 + 5x_2 + x_3 = -3 \end{cases}$
13	$\begin{cases} 2x_1 - x_2 + 2x_3 = 8 \\ x_1 + x_2 + 2x_3 = 11 \\ 4x_1 + x_2 + 4x_3 = 22 \end{cases}$	14	$\begin{cases} 2x_1 - x_2 - 3x_3 = -9 \\ x_1 + 5x_2 + x_3 = 20 \\ 3x_1 + 4x_2 + 2x_3 = 15 \end{cases}$
15	$\begin{cases} -3x_1 + 5x_2 + 6x_3 = -8 \\ 3x_1 + x_2 + x_3 = -4 \\ x_1 - 4x_2 - 2x_3 = -9 \end{cases}$	16	$\begin{cases} 3x_1 + x_2 + x_3 = -4 \\ -3x_1 + 5x_2 + 6x_3 = 36 \\ x_1 - 4x_2 - 2x_3 = -19 \end{cases}$
17	$\begin{cases} 3x_1 - x_2 + x_3 = -11 \\ 5x_1 + x_2 + 2x_3 = 8 \\ x_1 + 2x_2 + 4x_3 = 16 \end{cases}$	18	$\begin{cases} 3x_1 - x_2 + x_3 = 9 \\ 5x_1 + x_2 + 2x_3 = 11 \\ x_1 + 2x_2 + 4x_3 = 19 \end{cases}$

Вариант	Система уравнений	Вариант	Система уравнений
19	$\begin{cases} 2x_1 + 3x_2 + x_3 = 4 \\ 2x_1 + x_2 + 3x_3 = 0 \\ 3x_1 + 2x_2 + x_3 = 1 \end{cases}$	20	$\begin{cases} 2x_1 + 3x_2 + x_3 = 12 \\ 2x_1 + x_2 + 3x_3 = 16 \\ 3x_1 + 2x_2 + x_3 = 8 \end{cases}$
21	$\begin{cases} x_1 - 2x_2 + 3x_3 = 14 \\ 2x_1 + 3x_2 - 4x_3 = -16 \\ 3x_1 - 2x_2 - 5x_3 = -8 \end{cases}$	22	$\begin{cases} 3x_1 - 4x_2 - 2x_3 = 11 \\ 2x_1 - x_2 - 3x_3 = 4 \\ 3x_1 - 2x_2 + 4x_3 = 11 \end{cases}$
23	$\begin{cases} x_1 + 5x_2 - 6x_3 = -15 \\ 3x_1 + x_2 + 4x_3 = 13 \\ 2x_1 - 3x_2 + x_3 = 9 \end{cases}$	24	$\begin{cases} 7x_1 + 4x_2 - x_3 = 13 \\ 3x_1 + 2x_2 + 3x_3 = 3 \\ 2x_1 - 3x_2 + x_3 = -10 \end{cases}$
25	$\begin{cases} 2x_1 + x_2 + x_3 = 7 \\ x_1 + 2x_2 + x_3 = 8 \\ x_1 + x_2 + 2x_3 = 9 \end{cases}$	26	$\begin{cases} x_1 + 2x_2 + 3x_3 = 3 \\ 3x_1 + x_2 + 2x_3 = 7 \\ 2x_1 + 3x_2 + x_3 = 2 \end{cases}$
27	$\begin{cases} 5x_1 + 2x_2 - 4x_3 = -16 \\ x_1 + 3x_3 = -6 \\ 2x_1 - 3x_2 + x_3 = 9 \end{cases}$	28	$\begin{cases} x_1 + 4x_2 - x_3 = -9 \\ 4x_1 - x_2 + 5x_3 = -2 \\ 3x_2 - 7x_3 = -6 \end{cases}$
29	$\begin{cases} 2x_1 + 3x_2 + 4x_3 = 12 \\ 7x_1 - 5x_2 + x_3 = -33 \\ 4x_1 + x_3 = -7 \end{cases}$	30	$\begin{cases} x_1 + 4x_2 - x_3 = 6 \\ 5x_2 + 4x_3 = -20 \\ 3x_1 - 2x_2 + 5x_3 = -22 \end{cases}$

Таблица 9.2

Варианты заданий

Вариант	Численный метод
1	Метод Гаусса
2	Метод прогонки
3	Метод последовательных приближений
4	Метод Зейделя
5	Метод Гаусса
6	Метод прогонки
7	Метод последовательных приближений
8	Метод Зейделя
9	Метод Гаусса
10	Метод прогонки
11	Метод последовательных приближений
12	Метод Зейделя
13	Метод Гаусса

Вариант	Численный метод
14	Метод прогонки
15	Метод последовательных приближений
16	Метод Зейделя
17	Метод Гаусса
18	Метод прогонки
19	Метод последовательных приближений
20	Метод Зейделя
21	Метод Гаусса
22	Метод прогонки
23	Метод последовательных приближений
24	Метод Зейделя
25	Метод Гаусса
26	Метод прогонки
27	Метод последовательных приближений
28	Метод Зейделя
29	Метод Гаусса
30	Метод прогонки

Контрольные вопросы для защиты работ

1. Что из себя представляет система линейных алгебраических уравнений?
2. Какие численные методы используются для вычисления СЛАУ?
3. В чем суть итерационных методов решения СЛАУ?
4. В чем суть прямых методов решения СЛАУ?
5. Что выполняется на обратном ходе метода Гаусса?
6. Что вычисляется на первом этапе метода прогонки?
7. В чем отличие метода Зейделя от метода последовательных приближений?
8. Как определить условие сходимости итерационного процесса?
9. Как определить условие окончания итерационного процесса?

Лабораторная работа № 10

РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Цель работы: изучить численные методы решения нелинейных уравнений.

Краткие теоретические сведения

Пусть дано уравнение:

$$f(x) = 0, \quad (10.1)$$

где 1) функция $f(x)$ непрерывна на отрезке $[a, b]$ вместе со своими производными 1-го и 2-го порядка; 2) значения $f(x)$ на концах отрезка имеют разные знаки ($f(a) \cdot f(b) < 0$); 3) первая и вторая производные $f'(x)$ и $f''(x)$ сохраняют определенный знак на всем отрезке.

Условия 1) и 2) гарантируют, что на интервале $[a, b]$ находится хотя бы один корень, а из 3) следует, что $f(x)$ на данном интервале монотонна, и поэтому корень будет единственным.

Численное решение нелинейных (алгебраических или трансцендентных) уравнений (10.1) заключается в нахождении значений x , удовлетворяющих (с заданной точностью) данному уравнению.

Метод половинного деления

Метод половинного деления имеет и другие названия: метод деления отрезка пополам, метод дихотомии, метод проб, метод бисекций.

Процесс уточнения корня уравнения (10.1) на отрезке $[a, b]$, при условии, что функция $f(x)$ непрерывна на этом отрезке, заключается в следующем.

Исходный отрезок делится пополам. Если $f\left(\frac{a+b}{2}\right) = 0$, то $x^{(*)} = \left(\frac{a+b}{2}\right)$ является корнем уравнения (10.1). Если $f\left(\frac{a+b}{2}\right) \neq 0$,

то выбирается та из половин $\left[a, \frac{a+b}{2}\right]$ или $\left[\frac{a+b}{2}, b\right]$, на концах которой функция $f(x)$ имеет противоположные знаки. Новый суженный отрезок $[a^{(1)}, b^{(1)}]$ снова делится пополам и действия повторяются. В результате на каком-то этапе находится точный корень уравнения (10.1).

Если требуется найти корень с точностью ε , то деление отрезка пополам продолжается до тех пор, пока длина отрезка не станет меньше 2ε . Тогда середина последнего отрезка даст значение корня с требуемой точностью.

Метод Ньютона (метод касательных)

При нахождении корня уравнения (10.1) методом Ньютона, итерационный процесс определяется формулой

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (10.2)$$

Для начала вычислений требуется задание начального приближения x^0 .

В качестве условия окончания итераций в практических вычислениях часто используется правило $|x^{(k+1)} - x^{(k)}| < \varepsilon$, где ε – заданная точность. Тогда $x^{(*)} \approx x^{(k+1)}$ с точностью ε .

Метод секущих

Использование метода Ньютона предполагает вычисление на каждой итерации значения функции и ее производной. Заменяя производную функции приближенным разностным отношением:

$$f'(x^{(k)}) \approx \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

и подставляя его в (10.2), получаем итерационную формулу метода секущих:

$$x^{(k+1)} \approx x^{(k)} - \frac{f(x^{(k)})(x^{(k)} - x^{(k-1)})}{f(x^{(k)}) - f(x^{(k-1)})}. \quad (10.3)$$

Использование этого метода избавляет от необходимости расчета производной функции в процессе вычислений.

Метод является двухшаговым; как видно из формулы (10.3), результат $(k + 1)$ -го шага зависит от результатов (k) -го и $(k - 1)$ -го шагов.

Для выполнения первой итерации требуется задание двух начальных точек $x^{(0)}$ и $x^{(1)}$. Выбор начальной точки $x^{(0)}$ осуществляется по тому же принципу, что и в методе касательных, например, используя условие (10.2). Вторая начальная точка $x^{(1)}$ выбирается в непосредственной близости от $x^{(0)}$, желательно между точкой $x^{(0)}$ и искомым корнем.

Окончание счета по методу секущих, учитывая его быструю сходимость, можно контролировать путем проверки на малость модуля $|x^{(k+1)} - x^{(k)}|$.

Практическая часть

Задание. Разработать алгоритм и программу решения нелинейных уравнений методом, указанным в табл. 10.2. Решить указанное уравнение с заданным начальным приближением (табл. 10.1).

Результаты работы программы проверить подстановкой найденного решения в исходное уравнение для получения тождества.

Таблица 10.1

Нелинейные уравнения

Вариант	Вид уравнения	Начальное приближение корня
1	$\sqrt{2x+5} + \sqrt{5x+6} = \sqrt{12x+25}$	$x \approx 1$
2	$2(\cos(4x) - \sin(x)\cos(3x)) = \sin(4x) + \sin(2x)$	$x \approx 2$
3	$\left(1 + \frac{1}{2x}\right)\lg(3) + \lg(2) = \lg(27 - 3^{\frac{1}{x}})$	$x \approx 0,5$
4	$x^2 - 4x - 6 = \sqrt{2x^2 - 8x + 12}$	$x \approx 0$
5	$\sin\left(\frac{x}{2}\right)\cos\left(\frac{3x}{2}\right) - \frac{1}{\sqrt{3}}\sin(2x) = \sin\left(\frac{3x}{2}\right)\cos\left(\frac{x}{2}\right)$	$x \approx 7$
6	$\sqrt{5 + \sqrt[3]{x}} + \sqrt{5 - \sqrt[3]{x}} = \sqrt[3]{x}$	$x \approx 50$
7	$\lg(5) + \lg(x+10) = 1 - \lg(2x-1) + \lg(21x-20)$	$x \approx 1$
8	$\sqrt{3x^2+1} + \sqrt{x^2+3} = \sqrt{6x^2+10}$	$x \approx 1$
9	$5(1 + \cos(x)) = 2 + \sin^4 x - \cos^4 x$	$x \approx 2$
10	$x\lg(\sqrt[5]{5^{2x-8}}) - \lg(25) = 0$	$x \approx 5$
11	$\sqrt{x^2 - x - 1} + \sqrt{x^2 + x + 3} = \sqrt{2x^2 + 8}$	$x \approx 5$
12	$1 + \sin(2x) = (\cos(3x) + \sin(3x))^2$	$x \approx 3$
13	$\lg(3^x - 2^{4-x}) = 2 + 0,25\lg(16) - 0,5x\lg(4)$	$x \approx 3$
14	$\sqrt{x + 2\sqrt{x-1}} + \sqrt{x - 2\sqrt{x-1}} = x - 1$	$x \approx 3$
15	$\arcsin(x) - 0,2x - 0,1 = 0$	$x \approx 0$

Вариант	Вид уравнения	Начальное приближение корня
16	$\lg(x^2 + 1) = 2 \lg^{-1}(x^2 + 1) - \lg(6)$	$x \approx 3$
17	$\sin(3x) + 0,5x \cos(2x - 3) = \operatorname{tg}(x + 2)$	$x \approx 1$
18	$0,5(\lg(x^2 - 55x + 90) - \lg(x - 36)) = 0$	$x \approx 70$
19	$x^2 - 3x + 5 = \frac{1}{3}x + 3$	$x \approx 1$
20	$\sin(2x) - \sin(3x) + \sin(8x) = \cos\left(7x + \frac{3\pi}{2}\right)$	$x \approx 2$
21	$\sqrt{x-2} + 2\sqrt{3x-2} = -5x + 20$	$x \approx 2$
22	$\sqrt{x^2 - 19x + 204} - \sqrt{x^2 - 25x - 150} = 3\sqrt{\frac{x+5}{x-30}}$	$x \approx 0$
23	$\sin(2x) + \cos(2x) = \sqrt{x} \sin(3x)$	$x \approx 5$
24	$5^{x+6} - 3^{x+7} = 43 \cdot 5^{x+4} - 19 \cdot 3^{x+5}$	$x \approx 0$
25	$(x-1)x(x+1) + x(x+1)(x+2) = 3x^2 + x + 18x\sqrt{x} - 16$	$x \approx 0$
26	$\sin(3x) + \sin(5x) = 2(\cos^2 2x - \sin^2 3x)$	$x \approx 0$
27	$\operatorname{arctg}(x) - 2x + 1 = 0$	$x \approx 0$
28	$\sqrt{3x^2 + 1} + \sqrt{x^2 + 3} = \sqrt{6x^2 + 10}$	$x \approx 0$
29	$6\sin^2 x + \sin x \cos x = 2 + \cos^2 x$	$x \approx 1$
30	$2^{x+2} \cdot 3^{x+1} = 5^{-x+3}$	$x \approx 1$

Таблица 10.2

Вариант	Численный метод
1	Метод половинного деления
2	Метод Ньютона
3	Метод секущих
4	Метод половинного деления
5	Метод Ньютона
6	Метод секущих
7	Метод половинного деления
8	Метод Ньютона
9	Метод секущих
10	Метод половинного деления
11	Метод Ньютона
12	Метод секущих

Вариант	Численный метод
13	Метод половинного деления
14	Метод Ньютона
15	Метод секущих
16	Метод половинного деления
17	Метод Ньютона
18	Метод секущих
19	Метод половинного деления
20	Метод Ньютона
21	Метод секущих
22	Метод половинного деления
23	Метод Ньютона
24	Метод секущих
25	Метод половинного деления
26	Метод Ньютона
27	Метод секущих
28	Метод половинного деления
29	Метод Ньютона
30	Метод секущих

Контрольные вопросы для защиты работ

1. Какие численные методы используются для решения нелинейных уравнений?
2. В чем суть метода половинного деления?
3. В чем суть метода касательных?
4. В чем суть метода секущих?
5. Какой критерий служит для определения условия окончания итераций?

Литература

1. Информатика. Базовый курс / под ред. С. В. Симоновича. – 3-е изд. – СПб. : Питер, 2012. – 637 с.
2. Фаронов, В. В. Delphi. Программирование на языке высокого уровня : учеб. для вузов / В. В. Фаронов. – СПб. : Питер, 2007. – 640 с.
3. Колдаев, В. Д. Численные методы и программирование : учеб. пособие / В. Д. Колдаев ; под ред. Л. Г. Гагариной. – М. : ФОРУМ : ИНФРА-М, 2014. – 336 с.
4. Водополова, Н. В. Основы алгоритмизации : практ. пособие к лаборатор. и контрол. работам по курсам «Информатика» и «Основы информатики и вычислительной техники» / Н. В. Водополова, В. И. Мисюткин, С. А. Чабуркина. – Гомель : ГГТУ им. П. О. Сухого, 2004. – 35 с.
5. Коробейникова, Е. В. Программирование в среде DELPHI : практ. пособие по курсу «Информатика» для студентов всех специальностей / Е. В. Коробейникова, В. И. Токочаков. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 32 с.
6. Токочаков, В. И. Программирование в среде Delphi : практ. пособие по курсу «Информатика» для студентов всех специальностей / В. И. Токочаков, Е. В. Коробейникова. – Гомель : ГГТУ им. П. О. Сухого, 2005. – 38 с.

Содержание

Введение.....	3
Общие требования к порядку выполнения работ и по содержанию отчетов.....	4
<i>Лабораторная работа № 1. Обработка двумерных массивов</i>	<i>5</i>
<i>Лабораторная работа № 2. Обработка символьной информации</i>	<i>9</i>
<i>Лабораторная работа № 3. Программирование с использованием подпрограмм</i>	<i>13</i>
<i>Лабораторная работа № 4. Алгоритмы сортировки</i>	<i>20</i>
<i>Лабораторная работа № 5. Обработка записей.....</i>	<i>25</i>
<i>Лабораторная работа № 6. Обработка файлов.....</i>	<i>28</i>
<i>Лабораторная работа № 7. Динамические структуры данных.....</i>	<i>36</i>
<i>Лабораторная работа № 8. Вычисление интегралов</i>	<i>41</i>
<i>Лабораторная работа № 9. Решение систем линейных алгебраических уравнений</i>	<i>44</i>
<i>Лабораторная работа № 10. Решение нелинейных уравнений.....</i>	<i>53</i>
Литература	59

Учебное электронное издание комбинированного распространения

Учебное издание

**Самовендюк Николай Владимирович
Чабуркина София Абелевна**

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

**Практикум
по выполнению лабораторных работ
по одноименной дисциплине
для студентов специальности 1-53 01 01
«Автоматизация технологических процессов
и производств (по направлениям)»
дневной формы обучения**

Электронный аналог печатного издания

Редактор *Т. Н. Мисюрова*
Компьютерная верстка *Н. Б. Козловская*

Подписано в печать 01.10.18.

Формат 60x84/16. Бумага офсетная. Гарнитура «Таймс».

Ризография. Усл. печ. л. 3,72. Уч.-изд. л. 4,05.

Изд. № 40.

<http://www.gstu.by>

Издатель и полиграфическое исполнение
Гомельский государственный
технический университет имени П. О. Сухого.
Свидетельство о гос. регистрации в качестве издателя
печатных изданий за № 1/273 от 04.04.2014 г.
пр. Октября, 48, 246746, г. Гомель