

**Министерство образования и науки Российской
Федерации**

Федеральное государственное бюджетное образовательное
учреждение высшего образования

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)**

Кафедра технологий электронного обучения (ТЭО)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Методические указания к лабораторным и организации
самостоятельной работы для студентов по направлениям
подготовки (уровень бакалавра):

- 27.03.05 «Иноватика», профиль «Управление
инновациями в электронной технике»
- 27.03.02 «Управление качеством», профиль "Управление
качеством в информационных системах"

2018

Гураков Алексей Валерьевич

Информатика: Методические указания к лабораторным работам и организации самостоятельной работы для студентов направлений «Инноватика» и «Управление качеством» (уровень бакалавриата) / А.В. Гураков. – Томск, 2018. – 33 с.

Оглавление

1 Введение.....	4
2 Методические указания к проведению лабораторных работ.....	7
3. Методические указания для организации самостоятельной работы.....	28
4. Рекомендуемая литература.....	32

1 Введение

Дисциплина «Информационные технологии» является одними из важнейших общих математических и естественнонаучных дисциплин. Современный уровень развития вычислительной техники требует от специалистов высокого уровня знаний и навыков работы с компьютером для решения вопросов получения, хранения, преобразования, передачи и использования информации.

Целью изучения дисциплины является получение теоретических, практических знаний, основных навыков работы на персональном компьютере (ПК), а также освоение программных средств реализации информационных процессов, базового программного обеспечения (ПО).

Основными задачами изучения дисциплины «Информационные технологии» являются:

- привить студентам понимание принципов работы программ;
- научить студентов системному подходу к решению простых алгоритмических задач;
- дать студентам представление о современных технических и программных средствах;
- подготовить студентов к самостоятельному решению учебных и профессиональных задач средствами вычислительной техники.

В результате изучения дисциплин студенты должны:

знать:

- принципы алгоритмизации и программирования;
- процесс разработки программного обеспечения;
- принципы построения алгоритмов
- основные управляющие структуры и способы описания алгоритмов с использованием различных нотаций;

- понятие типа данных, форматы представления данных при решении задач с помощью компьютера;
- синтаксис и семантику алгоритмического языка программирования высокого уровня;
- основные алгоритмы типовых численных методов решения математических задач.
- основные алгоритмы сортировки и поиска данных;
- основные методы и источники самостоятельного приобретения новых знаний и умений.

уметь:

- работать в современных средах разработки, составлять блок-схемы алгоритмов, создавать программы на структурном языке программирования высокого уровня
- решать задачи, используя различные методы разработки алгоритмов и выбирая наиболее подходящие алгоритмы и средства их реализации в зависимости от постановки задачи;
- разрабатывать программы на языке программирования высокого уровня с использованием основных управляющих конструкций и стандартных типов данных;
- использовать языки и системы программирования для решения профессиональных задач;
- самостоятельно определять формирующиеся дефициты знаний, умений и навыков в ходе обучения;
- сформулировать проблемы, связанные с недостатком знаний и навыков, и выбрать подходы к их решению;
- самостоятельно организовывать свою деятельность, направленную на получение новых знаний и навыков основных методов и источников самостоятельного приобретения новых знаний и умений;

- оценивать и отбирать наиболее важную информацию, максимально полезную для решения поставленных задач при выполнении домашних заданий, при подготовке к контрольным мероприятиям.

владеть:

- навыками разработки и анализа алгоритмов решения типовых задач (сортировки и поиска данных и пр.), исследования их свойств;
- методами и инструментальными средствами разработки программ на языке программирования высокого уровня, их тестирования и отладки;
- навыками использования инструментальных программных средств в процессе разработки и сопровождения программных продуктов;
- навыками уверенного владения средствами поиска информации в различных источниках, рекомендованных для самостоятельного изучения;
- способностью применять полученные знания для решения новых задач в различных областях;
- навыками самостоятельного приобретения и использования новых знаний и умений.

2 Методические указания к проведению лабораторных работ

Лабораторные работы выполняются в компьютерном классе. Так как занятия проводятся в аудитории со свободным доступом, студентам рекомендуется приобрести flash-накопитель для сохранения своих работ.

2.1 Лабораторная работа «Основные структуры языка Free Pascal»

Цель работы

Изучить основные лексемы языка программирования, правила составления идентификаторов, скалярные типы данных. Изучить способы ввода и вывода данных. Научиться решать элементарные задачи.

Форма проведения

Лабораторная работа проводится в компьютерном классе. Задание состоит из двух частей. В первой части необходимо познакомиться с основными понятиями языка программирования и ответить на несколько вопросов. Изучить работу процедур ввода и вывода.

В второй части следует решить четыре задачи. Первая задача простая с линейным алгоритмом решения. Вторая задача с ветвлением. Третья и четвертая задачи с циклами.

Отчет по лабораторной работе должен включать в себя следующие пункты:

- титульный лист;
- цель работы;
- формулировку задачи;

- решение и ответ¹.

В случаях, когда необходимо написать программу, в отчете необходимо придерживаться следующего порядка описания:

1. формулировка задачи;
2. анализ задачи и решение;
3. алгоритм решения задачи в виде блок-схемы;
4. разработка структуры программы;
5. код программы с необходимыми комментариями в текстовом формате;
6. результаты работы программы (скрин).

Варианты заданий

Задание №1

1. Перечислите все символы языка программирования Free Pascal.
2. Сформулируйте правила составления идентификаторов. Придумайте 5 идентификаторов в соответствии с этими правилами. Придумайте не менее 5 примеров идентификаторов, составленных с ошибкой.
3. Создайте программу и опишите в ней не менее 12 переменных, относящихся к различным скалярным типам данных. Сохраните программу с именем, которое формируется по правилу²:

инициалы+средняя_цифра_группы +вариант+_+номер_программы.pas

#Пример: gav103_01.pas

¹ Внимание! Решение и ответ на задачу должны быть написаны сразу после текста задания.

² В дальнейшем все программы сохранять, используя этот способ формирования имени файла.

С помощью оператора присваивания присвойте этим переменным следующие значения:

**V; 254*V; 1-254*V; -1*V; 65536; 3.5*V; 3.5E-45/V; 3.5E100;
3.5E100*V; 2¹³; 'G'; TRUE**

где V – номер варианта (переменная типа byte)

4. Написать программу, которая реализует следующий диалог:

- *What's your name?*
- *Name*
- *Hi, Name!*

Name – собственное имя. Вводится с клавиатуры во второй строке и выводится на экран в третьей. Что бы ввести имя используйте переменную типа **string**. Все символы и пунктуация должны быть сохранены.

5. Наберите программу¹:

```
Uses CRT;  
Var  
    a, b: byte;  
    c, d: byte;  
Begin  
    ClrScr;  
    Read(a, b);  
    Read(c, d);  
    Writeln (' a=', a, ' b=', b, ' c=', c, ' d=', d);  
End.
```

Запустите эту программу несколько раз и введите значения для переменных различными способами:

- *четыре значения в одну строку через пробел;*
- *по два значения в строке;*
- *в «столбик».*

¹ Внимание! Вносить изменения в этот код и кода программ следующих заданий строго запрещается. Запускать программу следует только в IDE Free Pascal

Отразите в отчете результаты эксперимента (скрин) и объясните каждый из них.

Поменяйте все процедуры Read на Readln и повторите эксперимент. Сравните результаты.

6. Наберите программу:

```
Uses CRT;  
Var  
    a, b: byte;  
    c, d: char;  
Begin  
    ClrScr;  
    Read(a, b);  
    Read(c, d);  
    Writeln (' a=', a, ' b=', b, ' c=', c, ' d=', d);  
End.
```

Запустите эту программу несколько раз и введите значения для переменных различными способами:

- *четыре значения в одну строку через пробел;*
- *четыре значения в одну строку без пробела;*
- *по два значения в строке;*
- *по два значения в строке без пробелов;*
- *в «столбик».*

Отразите в отчете результаты эксперимента и объясните их.

Поменяйте все процедуры Read на Readln и повторите эксперимент. Сравните результаты.

7. Наберите программу

```
Uses CRT;  
Var  
    a: byte;  
    c: char;  
    b: real;  
    d: real;
```

Begin

```
ClrScr;  
a:=23; b:=1.28; c:='d'; d:=1.216E-3;  
Writeln (a:M:N);  
Writeln (b:M:N);  
Writeln (c:M:N);  
Writeln (d:M:N);
```

End.

Числа M и N берутся из таблицы. В некоторых случаях значение для N отсутствует. В этом случае необходимо удалить из программы и N и двоеточие перед ним.

M	N
4	2
5	3
7	2
3	1
1	4
1	-
2	-
3	-
5	-

Запустите программу для разных значений M и N¹. Все результаты отразить в отчете и объяснить.

Задание №2

Напишите программы, которые решают следующие задачи.

ВАРИАНТ 1.

1. Вводятся два числа a и b. Получить их сумму, разность и произведение.

¹ В некоторых случаях программа будет выдавать ошибку. Необходимо определить строку с ошибкой и закомментировать её для данного случая. Переходя к следующей паре M и N комментарий нужно убрать.

- С клавиатуры вводится значение переменных X1 и X2 типа REAL, и R1 и R2 типа BYTE. Необходимо определить значение $Y = \frac{\cos(X1)+R1 \cdot 4}{\ln(R2)-X2}$. Полученное значение Y необходимо вывести на экран.
- Составить программу вычисления суммы конечного ряда

$$\sum_{n=1}^5 \frac{1}{n!}.$$

- Составить программу вычисления суммы бесконечного ряда с точность EPS. Значение EPS вводится с клавиатуры.

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{(2 \cdot n)!}.$$

ВАРИАНТ 2.

- Вводится длина ребра куба. Найти объем куба и площадь его боковой поверхности.
- С клавиатуры вводится значение переменных X1 и X2 типа REAL, и R1 и R2 типа BYTE. Необходимо определить значение $Y = e^{X1-R1} \cdot \frac{|X2-R1|}{\sin(X2)}$. Полученное значение Y необходимо вывести на экран.
- Составить программу вычисления суммы конечного ряда

$$\sum_{n=1}^5 \frac{1}{(2n)!}.$$

- Составить программу вычисления суммы бесконечного ряда с точность EPS. Значение EPS вводится с клавиатуры.

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^2}.$$

ВАРИАНТ 3.

1. Вводятся два положительных числа. Найти среднее арифметическое и среднее геометрическое этих чисел.
2. С клавиатуры вводится значение переменных X1 и X2 типа REAL, и R1 и R2 типа BYTE. Необходимо определить значение $Y = \ln|\operatorname{arctg}(X1)| \cdot \left(\frac{X2-R2}{R1}\right)^2$. Полученное значение Y необходимо вывести на экран.
3. Составить программу вычисления суммы конечного ряда

$$\sum_{n=1}^5 \frac{1}{(2n-1)!}$$

4. Составить программу вычисления суммы бесконечного ряда с точность EPS. Значение EPS вводится с клавиатуры.

$$\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n+3}.$$

ВАРИАНТ 4.

1. Вводятся два числа. Найти среднее арифметическое этих чисел и среднее геометрическое их модулей.
2. С клавиатуры вводится значение переменных X1 и X2 типа REAL, и R1 и R2 типа BYTE. Необходимо определить значение $Y = e^{\cos(X1)} \cdot \frac{(\ln(R1)-R2)^2}{\operatorname{tg}(X2)}$. Полученное значение Y необходимо вывести на экран.

3. Составить программу вычисления суммы конечного ряда

$$\sum_{n=1}^5 \frac{1}{(2n)! + 2}$$

4. Составить программу вычисления суммы бесконечного ряда с точность EPS. Значение EPS вводится с клавиатуры.

$$\sum_{n=1}^{\infty} \frac{(-1)^n}{(n+1)(n+2)}$$

ВАРИАНТ 5.

1. Вводятся катеты прямоугольного треугольника. Найти его гипотенузу и площадь.
2. С клавиатуры вводится значение переменных X1 и X2 типа REAL, и R1 и R2 типа BYTE. Необходимо определить значение $Y = \sqrt{\frac{R2+R1}{\ln(R2)} + \frac{(X1-X2)^2}{|X2|}}$. Полученное значение Y необходимо вывести на экран.
3. Составить программу вычисления суммы конечного ряда

$$\sum_{n=1}^5 \frac{1}{n! + 4}$$

4. Составить программу вычисления суммы бесконечного ряда с точность EPS. Значение EPS вводится с клавиатуры.

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{(n+2)n}$$

Контрольные вопросы

1. С помощью какой процедуры выводится информация на экран?
2. Зачем нужен форматированный вывод? Как он реализуется во Free Pascal?
3. Перечислите этапы разработки программы.
4. Какие виды алгоритмов Вы знаете?
5. Какие виды циклов Вам знакомы?

2.2 Лабораторная работа «Обработка данных строкового типа»

Цель работы

Научиться обрабатывать данные строкового типа средствами языка программирования Free Pascal.

Форма проведения

Лабораторная работа проводится в компьютерном классе. Задание состоит из двух задач.

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. титульный лист;
2. цель работы;
3. формулировка задачи;
4. анализ задачи и решение;
5. алгоритм решения задачи в виде блок-схемы;
6. разработка структуры программы;
7. код программы с необходимыми комментариями в текстовом формате;
8. результаты работы программы (скрин);
9. выводы.

Пункты 3-8 повторяются для каждой задачи.

Варианты заданий

ВАРИАНТ 1.

1. Дана строка символов. Подсчитать сколько раз среди данных символов встречается символ '+' и сколько раз символ '*'.
2. Составить программу, заменяющую в исходной строке символов все единицы нулями и все нули единицами. Замена должна выполняться с заданного символа.

ВАРИАНТ 2.

1. Дана строка символов. Подсчитать сколько раз среди данных символов встречается символ '/' и сколько раз символ 'k'.
2. Составит программу "сжатия" исходной строки символов: каждая подстрока, состоящая из нескольких вхождений одного и того же символа, должна быть заменена на текст «x (k)», где x - символ, а k - строка, являющаяся записью числа вхождений символа в исходную строку.

ВАРИАНТ 3.

1. Дана строка символов длиной n. Удалить из строки все группы букв вида abcd.
2. Группы символов, разделенные пробелами и не содержащие пробелов внутри себя, будем называть словами. Выравнивание строки заключается в том, что между ее отдельными словами дополнительно вносятся пробелы так, чтобы длина строки стала равной заданной длине (предполагается, что требуемая длина не меньше исходной), а последнее слово строки сдвинулось к ее правому краю. Составить программу выравнивания заданной строки текста по ширине экрана.

ВАРИАНТ 4.

1. Дана строка символов длиной n . Удалить из строки все группы букв вида $adbe$.
2. Дана строка символов. Известно, что первый символ строки отличен от пробела и что в строке имеется хотя бы один пробел. Рассматривается подстрока символов, предшествующих первому пробелу (местоположение первого пробела заранее неизвестно), преобразовать данную подстроку, удалив из нее все символы, не являющиеся буквами.

ВАРИАНТ 5.

1. Дана строка символов длиной n . Подсчитать число вхождений в эту строку группы букв abd .
2. Дана строка символов. Известно, что первый символ строки отличен от пробела и что в строке имеется хотя бы один пробел. Рассматривается подстрока символов, предшествующих первому пробелу (местоположение первого пробела заранее неизвестно). Преобразовать данную подстроку, заменив все малые буквы одноименными большими.

Контрольные вопросы

1. Перечислите операции, которые можно производить с данными типа *string*?
2. Какое значение хранит нулевой символ (байт) строки после ее инициализации (до присвоения какого-либо значения строке)?

2.3 Лабораторная работа «Подпрограммы»

Цель работы

Изучить особенности описания и использования подпрограмм в языке программирования Free Pascal

Форма проведения

Лабораторная работа проводится в компьютерном классе. Студенту выдается две задачи в соответствии с вариантом. Первая задача основная. Вторая вспомогательная и оформляется в виде подпрограммы.

Порядок выполнения работы.

Шаг 1 Составить алгоритм решения основной задачи. Написать программу без использования подпрограмм.

Шаг 2 Решить вторую задачу. Записать алгоритм решения в виде функции.

Шаг 3 Решить основную задачу с использованием созданной на предыдущем этапе функции. Написать программу.

Шаг 4 Алгоритм решения второй задачи, разработанный на втором шаге, записать в виде процедуры.

Шаг 5 Решить основную задачу с использованием созданной на предыдущем этапе процедуры. Написать программу.

Шаг 6 Написать вывод, в котором обязательно объяснить какие изменения понадобилось внести в каждую программу и ответить на вопрос: «Какой вид подпрограмм лучше всего использовать для решения основной задачи и почему?»

В результате должно получиться три программы, которые нужно выслать вместе с отчетом на проверку.

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. титульный лист;
2. цель работы;
3. подробное описание решения основной задачи:
 - а) формулировка задачи;
 - б) анализ задачи и решение;

- в) алгоритм решения задачи в виде блок-схемы;
 - г) разработка структуры программы;
 - д) код программы с необходимыми комментариями в текстовом формате;
 - е) результаты работы программы (скрин);
4. подробное решение вспомогательной задачи, которое содержит те же пункты, что перечислены в предыдущем пункте. За исключением пунктов г) и д), в которых следует разработать и привести код функции;
 5. решение основной задачи с использованием функции;
 6. разработка процедуры на основании решения, приведенного в пункте 4.
 7. решение основной задачи с использованием процедуры;
 8. выводы.

Варианты заданий

1.	<p>Сколькими способами можно распределить уроки в N классах (N – четное число) между M ($M=N \text{ div } 2$) учителями, если каждый учитель будет преподавать в двух классах? Определить подпрограмму подсчета факториала.</p> <p>Общее количество различных наборов при выборе k элементов из n без возвращения и без учёта порядка равняется</p> $C_n^k = \frac{n!}{k!(n-k)!}$
2.	<p>Задан одномерный массив, заполненный случайными числами. Определить среднее геометрическое модулей минимального и максимального значений. Написать подпрограмму, которая определяет индекс минимального или максимального значения.</p>
3.	<p>Заполнить массив из N ячеек случайным образом неповторяющимися целыми числами в диапазоне от -20 до 20. Написать подпрограмму, которая находит в массиве адрес</p>

	числа N в интервале от k_1 -й позиции до k_2 -й.
4.	Задан двумерный массив, в котором хранятся координаты точек, лежащих на плоскости (x – первая строка массива, y – вторая строка). Определить число точек лежащих в N -ой четверти. Описать подпрограмму, определяющую номер координатной четверти, в которой находится точка с координатами (x, y) .
5.	Написать подпрограмму, которая подсчитывает, сколько раз в строке встречается определенный символ. В строке символов определить, какой символ '=' или '*' встречается чаще.

Контрольные вопросы

1. Что такое формальные и фактические параметры?
2. Чем отличаются процедуры от функций?
3. Опишите структуру функции?
4. Опишите структуру процедуры?
5. Как обращаться к подпрограмме из основного блока программы?

2.4 Лабораторная работа «Создание пользовательских модулей»

Цель работы

Научиться создавать пользовательские модули и использовать подпрограммы, описанные в них, в своих программах.

Форма проведения

Лабораторная работа проводится в компьютерном классе. Задание состоит из двух частей. В первой необходимо решить несколько задач, объединенных одной темой. Разработать алгоритмы и оформить их в виде подпрограмм. Последние скомпоновать в модуль.

Во второй части задания требуется решить задачу с использованием подпрограмм, которые размещены в модуле.

В результате должно получиться программа, файл с кодом модуля, сам модуль. Их нужно выслать вместе с отчетом на проверку.

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. титульный лист;
2. цель работы;
3. подробное описание решения всех задач, которые будут размещены в модуле;
4. код модуля;
5. подробное описание решения основной задачи.
6. выводы.

Описание решения всех задач должно включать в себя следующие пункты:

1. формулировка задачи;
2. анализ задачи и решение;
3. алгоритм решения задачи в виде блок-схемы;
4. разработка структуры подпрограммы;
5. код подпрограммы с необходимыми комментариями в текстовом формате.

Варианты заданий

№	Задание
1	<p>. Найти значение переменной</p> $z = \begin{cases} \frac{x+y}{x \cdot y}, & \text{если } x > y \\ \frac{y-x}{y}, & \text{если } x \leq y \end{cases}$ <p>Необходимо создать: несколько подпрограмм, которые</p>

	<p>позволяют складывать, вычитать, умножать и делить комплексные числа; функции, возвращающие аргумент и модуль комплексного числа; процедуры, позволяющие ввести комплексное число с клавиатуры и вывести его на экран. Также потребуется описать пользовательский тип <i>TComplex</i> (запись).</p>
2	<p>Доказать верность следующих тождеств для четырех значений углов, принадлежащих различным четвертям:</p> $tg \alpha - ctg \alpha = -2ctg 2\alpha$ $tg \alpha + ctg \alpha = 2cosec 2\alpha$ $1 \pm tg \alpha = \frac{\sqrt{2} \sin\left(\frac{\pi}{2} \pm \alpha\right)}{\cos \alpha}$ <p>Написать функции, которые определяют синус, косинус, тангенс, котангенс, секанс и косеканс угла, заданного в градусах. Создать функцию, которая переводит радианы в градусы.</p> <p>Ответ вывести в виде таблицы.</p> <p><i>Нельзя использовать стандартные тригонометрические функции, а также одни пользовательские функции при расчете других (т.е. нельзя, например, при расчете котангенса использовать тангенс).</i></p>
3	<p>Задан двумерный массив размерностью $M \times (N+1)$, в который помещены результаты M экспериментов по N опытов в каждом. В первом столбце помещены значения аргумента исследуемой функции. Зная, что график функции есть прямая, рассчитать коэффициенты этой прямой методом наименьших квадратов.</p> <p>Для решения понадобится написать подпрограммы определения среднего арифметического, суммы нескольких чисел и их квадратов. Коэффициенты k и b</p>

	<p>рассчитываются с помощью функции.</p> <p>Ответ вывести в виде: $\mathbf{y(x)=kx+b}$.</p> <p>Считать, что результаты эксперимента хранятся в файле.</p>
4	<p>Одномерный массив произвольной длины упорядочить по возрастанию и по убыванию разными методами.</p> <p>Написать несколько процедур и функций, с помощью которых можно сделать следующее: заполнить массив (двумя способами); вывести на экран; найти минимальный и максимальный элементы; найти индекс минимального и максимального элемента; сортировку методами простого выбора, простой замены, простого обмена.</p> <p><i>Тип массива можно определить самостоятельно. Если процедуры и функции смогут работать с массивами разного типа, то +4 балла</i></p>
5	<p>Коллинеарны ли векторы c_1 и c_2, построенные по векторам a (a_x, a_y) и b (b_x, b_y):</p> $\vec{c}_1 = \gamma_1 \vec{a} + \gamma_2 \vec{b}$ $\vec{c}_2 = \sigma_1 \vec{a} + \sigma_2 \vec{b}$ <p>Необходимо написать следующие функции и процедуры:</p> <ul style="list-style-type: none"> • определение длины вектора; • сложение двух векторов; • скалярное произведение двух векторов; • векторное произведение двух векторов; • длина проекции вектора на оси; • угол между векторами, • проверка на параллельность и перпендикулярность.

Использовать пользовательский тип <i>TVector</i> (запись).
--

Контрольные вопросы

1. Зачем необходимы модули?
2. Опишите структуру модуля.
3. Как подключаются модули к программе?

2.5 Лабораторная работа «Обработка одномерных и двумерных массивов»

Цель работы

Изучение основных алгоритмов обработки одномерных массивов и средств реализации этих алгоритмов на языке Free Pascal.

Форма проведения

Лабораторная работа проводится в компьютерном классе. Задание состоит из двух задач, связанных с обработкой одномерного и двумерного массивов соответственно. Необходимо решить задачи и написать программы. На проверку преподавателю высылаются две программы и отчет

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. титульный лист;
2. цель работы;
3. для каждой задачи привести
 - а) текст задачи;
 - б) анализ задачи и решение;
 - в) алгоритм решения задачи в виде блок-схемы;
 - г) разработка структуры подпрограммы;

- д) код программы с необходимыми комментариями в текстовом формате;
- е) результат выполнения программы (скрин).

Варианты первого задания

1. Заполнить массив из N ячеек случайными целыми числами в диапазоне от -120 до 250.
Определить сумму элементов массива с k_1 -го по k_2 -й. (значения k_1 и k_2 вводятся с клавиатуры).
2. Заполнить массив из N ячеек случайными вещественными числами в диапазоне от -7 до 9.
Определить среднее арифметическое элементов массива с k_1 -го по k_2 -й. (значения k_1 и k_2 вводятся с клавиатуры).
3. Заполнить массив из N ячеек символами латинского алфавита (заглавными и прописными).
Проверить, можно ли из этих символов составить слово, введенное с клавиатуры.
4. Заполнить массив из N ячеек символами русского алфавита (заглавными и прописными).
Проверить, можно ли из этих символов составить слово, введенное с клавиатуры.
5. Заполнить массив десятью первыми членами арифметической прогрессии с известным первым членом a и её разностью p . Вывести на экран все элементы, следующие за первым элементом, большего заданного числа k . Если таких элементов нет, вывести на экран соответствующее сообщение.

Варианты второго задания

1. Заполнить двумерный массив $M \times N$ цифрами 1, 2, 3, 4 и т.д. по спирали.

2. Заполнить двумерный массив $M \times N$ следующим образом: в первую строку и первый столбец поместить единицы, в остальные ячейки поместить сумму значений в смежных ячейках, расположенных слева и выше.
3. Заполнить двумерный массив $M \times N$ «змейкой» начиная с первой строки цифрами 1, 2, 3 и т.д.
4. Двумерный массив $M \times N$ заполнить случайными целыми числами в диапазоне от 0 до 150. Вывести на экран последовательность чисел, получившуюся при чтении этого массива по спирали.
5. Двумерный массив $M \times N$ заполнить случайными неповторяющимися целыми числами в диапазоне от 100 до 200. Найти индексы максимального элемента массива.

Контрольные вопросы

1. Как получить доступ к элементам массива?
2. Что такое массив?
3. Что такое индекс массива?
4. Как описать массив на языке Free Pascal?
5. Каким образом можно составить выражение для генерации массива случайными целыми числами на заданном промежутке?

2.6 Лабораторная работа «Сортировка одномерных массивов»

Цель работы

Изучить простые методы сортировки одномерных массивов.

Форма проведения

Лабораторная работа проводится в компьютерном классе. Требуется разработать программу сортировки одномерного массива, согласно своему варианту.

Требования к программе:

- сортировка массива должна быть оформлена в виде подпрограммы;
- массив должен передаваться в подпрограмму в качестве параметра;
- размерность обрабатываемого массива должна задаваться пользователем с клавиатуры;
- массив должен заполняться случайными значениями;
- вывод значений отсортированного массива на экран компьютера должен быть также оформлен в виде подпрограммы;
- для сортируемого массива необходимо организовать вывод его значений на экран компьютера до, и после сортировки.

Отчет по лабораторной работе должен включать в себя следующие пункты:

1. титульный лист;
2. цель работы;
3. формулировка задачи;
4. анализ задачи и решение;
5. алгоритм решения задачи в виде блок-схемы;
6. разработка структуры программы;
7. код программы с необходимыми комментариями в текстовом формате;
8. результаты работы программы (скрин);
9. выводы.

Варианты заданий.

Метод сортировки	Тип массива			
	<i>Цельный</i>	<i>Вещественный</i>	<i>Символьный</i>	<i>Строковый</i>
<i>Простого включения</i>	1	6	11	16
<i>Двоичного включения</i>	2	7	12	17
<i>Простого выбора</i>	3	8	13	18
<i>Простого обмена</i>	4	9	14	19
<i>Шейкерная</i>	5	10	15	20

Контрольные вопросы.

1. Что означает отсортировать одномерный массив по возрастанию?
2. На какие два этапа разбивается весь процесс сортировки?
3. В чем заключается суть сортировки с помощью прямого включения?
4. В чем заключается суть сортировки с двоичным включением?
5. В чем заключается суть сортировки с помощью прямого выбора?
6. В чем заключается суть пузырьковой сортировки?
7. В чем заключается суть шейкерной сортировки?

3. Методические указания для организации самостоятельной работы

3.1 Общие положения

Целью самостоятельной работы является систематизация, расширение и закрепление теоретических знаний, использование материала, собранного и полученного

в ходе самостоятельной подготовки к лабораторным работам и практическим занятиям.

Самостоятельная работа включает в себя подготовку к лабораторным работам и практическим занятиям, проработку лекционного материала и подготовку к контрольным работам, изучение тем дисциплины, вынесенных на самостоятельное изучение.

3.2 Проработка лекционного материала

Изучение теоретической части дисциплин призвано не только углубить и закрепить знания, полученные на аудиторных занятиях, но и способствовать развитию у студентов творческих навыков, инициативы и организовать свое время.

Проработка лекционного материала включает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;
- знакомство с Интернет-источниками;
- подготовку к различным формам контроля (контрольные работы);
- подготовку ответов на вопросы по различным темам дисциплины в той последовательности, в какой они представлены.

Планирование времени, необходимого на изучение дисциплин, студентам лучше всего осуществлять весь семестр, предусматривая при этом регулярное повторение материала.

Материал, законспектированный на лекциях, необходимо регулярно прорабатывать и дополнять сведениями из других источников литературы, представленных не только в программе дисциплины, но и в периодических изданиях.

При изучении дисциплины сначала необходимо по каждой теме прочитать рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.

Задачи, стоящие перед студентом при подготовке и написании контрольной работы:

- закрепление полученных ранее теоретических знаний;
- выработка навыков самостоятельной работы;
- выяснение подготовленности студентов к зачету.

Используются компьютерные контрольные работы, которые находятся в электронном курсе «Информационные технологии». Выполняются онлайн. Темы контрольных работ:

1. Понятие языка высокого уровня
2. Алгоритмы и алгоритмизация
3. Структура программы
4. Целочисленные и вещественные типы
5. Ввод и вывод данных
6. Организация ветвлений
7. Операторы цикла
8. Подпрограммы
9. Файлы
10. Множества
11. Строки
12. Массивы
13. Сортировка и поиск
14. Записи

3.3 Подготовка к лабораторным работам

Проведение лабораторных работ включает в себя следующие этапы:

- постановку темы занятий и определение задач лабораторной работы;
- определение порядка выполнения лабораторной работы или отдельных её этапов;
- непосредственное выполнение лабораторной работы студентам и контроль за ходом занятий;
- подведение итогов лабораторной работы и формулирование основных выводов;
- оформление отчета;
- отправка отчета на проверку;
- защита лабораторной работы (для текстовых работ).

При подготовке к лабораторным занятиям необходимо заранее изучить методические рекомендации по его проведению. Обратит внимание на цель занятия, на основные вопросы для подготовки к занятию, на содержание темы занятия.

Если в процессе выполнения лабораторной работы или при изучении теоретического материала у студента возникают вопросы, разрешить которые самостоятельно не удастся, необходимо обратиться к преподавателю для получения у него разъяснений или указаний. Сделать это можно лично на занятиях или в часы консультаций, по электронной почте или в специальном форуме «Консультация», который находится в электронном курсе.

4. Рекомендуемая литература

1. Гураков А. В., Мещерякова О. И., Мещеряков П. С. Информатика II. — Томск: ТУСУР 2015 г.— 112 с. — Электронное издание. — ISBN tusur_2017_51 : В другом месте, <https://ibooks.ru/reading.php?productid=356501> (дата обращения: 5.07.2018)
2. Алексеев, Е.Р. Free Pascal и Lazarus: Учебник по программированию [Электронный ресурс] : учебник / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. — Электрон. дан. — Москва : ДМК Пресс, 2010. — 438 с.: В другом месте, <https://e.lanbook.com/book/1267> (дата обращения: 5.07.2018)
3. Трофимов, В. В. Информатика в 2 т. Том 1 : учебник для академического бакалавриата / В. В. Трофимов, М. И. Барабанова ; отв. ред. В. В. Трофимов. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 553 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02613-9.: В другом месте, <http://www.biblio-online.ru/book/F0FE998E-C747-4ABB-84E3-07A146765A50> (дата обращения: 5.07.2018)
4. Трофимов, В. В. Информатика в 2 т. Том 2 : учебник для академического бакалавриата / В. В. Трофимов ; отв. ред. В. В. Трофимов. — 3-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 406 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02615-3.: В другом месте, <http://www.biblio-online.ru/book/5A795D83-C63B-4210-93C5-B3AC5093CC91> (дата обращения: 5.07.2018)
5. Трофимов, В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М. : Издательство Юрайт, 2018. — 137 с. — (Серия : Бакалавр. Академический курс. Модуль.).

- ISBN 978-5-9916-9866-5. — Режим доступа : : В другом месте, <https://biblio-online.ru/viewer/B08DB966-3F96-4B5A-B030-E3CD9085CED4/algorithmizaciya-i-programmirovanie#page/1>
6. Гураков А.В. Информатика для студентов направлений «Инноватика» и «Управление качеством» (уровень бакалавриата) [Электронный ресурс]. – Режим доступа: <http://mooc.tusur.ru/course/view.php?id=3> (дата обращения: 5.07.2018).
 7. Гураков А.В. Информационные технологии для студентов направлений «Инноватика» и «Управление качеством» (уровень бакалавриата) [Электронный ресурс]. – Режим доступа: <http://mooc.tusur.ru/course/view.php?id=12> (дата обращения: 5.07.2018).
 8. Фаронов В.В. Турбо Паскаль 7.0: Практика программирования: Учебное пособие - 7-е изд., перераб.. - М.: Нолидж, 2001. - 416 с.: ил.
 9. Фигурнов А.Ф. TURBO PASCAL 6.0, 1992.
 10. Вирт Н. Алгоритмы и структуры данных./Пер.с англ. – 2-е изд. – СПб.: Невский Диалект, 2001. -352 с.
 11. Фаронов В.В. Turbo Pascal: Наиболее полное руководство. – СПб.: БХВ-Петербург, 2006. – 1037[1] с.: ил., табл. (49 экз. в библи.)
 12. Фигурнов В.Э. IBM PC для пользователя. – 6-е изд., перераб. И доп. – М.: Финансы и статистика, КомпьютерПресс, 1995.