

АЛМАТИНСКИЙ ФИЛИАЛ НЕГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ ПРОФСОЮЗОВ»



Т.Г. ПЛОТНИКОВА

**ИНФОРМАТИКА И
ПРОГРАММИРОВАНИЕ
(РАЗДЕЛ «ПРОГРАММИРОВАНИЕ»)**

**ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ
К ПРАКТИЧЕСКИМ ЗАНЯТИЯМ**

Алматы
2013

РЕЦЕНЗЕНТЫ:

Баймулдина Н.С., кандидат педагогических наук,
старший преподаватель Казахского национального
педагогического университета им.Абая

Конева С.Н., кандидат педагогических наук, доцент
Алматинского филиала НОУ ВПО
«Санкт-Петербургский Гуманитарный университет профсоюзов»

Автор-составитель:
ПЛОТНИКОВА Т.Г.,
доцент Алматинского филиала НОУ ВПО
«Санкт-Петербургский Гуманитарный университет профсоюзов»

Рекомендовано к печати
Методическим советом Алматинского филиала НОУ ВПО
«Санкт-Петербургский Гуманитарный университет профсоюзов»
от « 22 » мая 2013 г. Протокол № 6.

© Плотникова Т.Г, 2013.
© АФ НОУ ВПО «СПбГУП», 2013

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
I. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАБОТ.....	6
II. ТЕМАТИЧЕСКОЕ СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	
1. Организация ввода и вывода на языке Pascal.....	10
2. Программирование линейных алгоритмов.....	14
3. Программирование разветвляющихся алгоритмов.....	17
4. Программирование циклических алгоритмов.....	21
5. Программирование с использованием массивов.....	26
6. Программирование задач обработки строковых данных.....	32
7. Программирование с использованием процедур и функций.....	37
8. Организация работы с файлами.....	42
III. ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ.....	48
IV. ПРИМЕРНЫЕ ТЕСТОВЫЕ ЗАДАНИЯ.....	52
V. ОСНОВНЫЕ ВОПРОСЫ ПО ИЗУЧАННОЙ ДИСЦИПЛИНЕ.....	62
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	64

ВВЕДЕНИЕ

Дисциплина «Информатика и программирование» относится к базовой части математического и естественно-научного цикла образовательной программы направления подготовки 230700 «Прикладная информатика».

Целью изучения дисциплины является расширение мировоззрения и формирование у студентов самостоятельного мышления в области информационных подходов к анализу процессов и явлений профессиональной сферы и информационных технологий.

Задача изучения дисциплины – получение систематических знаний об информационных процессах, системах, средствах и технологиях; приобретение специальных знаний и умений, необходимых для работы в фирмах и организациях профессионально работающих в информационной сфере или использующих новые информационные технологии.

Процесс изучения дисциплины направлен на формирование у студента следующих *компетенций*:

- способность использовать, обобщать и анализировать информацию, ставить цели и находить пути их достижения в условиях формирования и развития информационного общества (ОК-1);
- способность самостоятельно приобретать и использовать в практической деятельности новые знания и умения, стремиться к саморазвитию (ОК- 5);
- способность понимать сущность и проблемы развития современного информационного общества (ОК-7);
- способность использовать основные законы естественнонаучных дисциплин в профессиональной деятельности и эксплуатировать современное электронное оборудование и информационно-коммуникационные технологии в соответствии с целями образовательной программы бакалавра (ПК-3);
- способность ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий (ПК- 4);
- способность использовать технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств (ПК- 7);
- способность применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы (ПК-10);

- способность анализировать рынок программно-технических средств, информационных продуктов и услуг для решения прикладных задач и создания информационных систем (ПК-19).

В результате изучения дисциплины студенты должны:

- знать:

- основные положения информатики как научной дисциплины, изучающей процессы обработки информации, принципы организации и функционирования современных средств обработки информации;

- базовые принципы информационного анализа процессов и явлений предметной области;

- методы проектирования и разработки алгоритмов и программ.

- уметь:

- решать задачи обработки данных с помощью современных инструментальных средств конечного пользователя;

- проектировать, программировать, отлаживать и документировать программы на одном из языков высокого уровня с использованием современных технологий программирования.

- владеть навыками программирования в современных средах.

В данных методические указаниях приводятся рекомендации по выполнению практических работ по дисциплине «Информатика и программирование» (раздел «Программирование»). В качестве языка программирования при написании программ используется алгоритмический язык Pascal.

Pascal – язык программирования высокого уровня, широко применяется при обучении программированию в высшей школе, является базой для большого числа других языков.

I. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ РАЗЛИЧНЫХ ВИДОВ РАБОТ

1. ОБЩИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ РАБОТЫ

Перед выполнением практических работ студентам необходимо изучить теоретический материал по данной теме.

При решении задач по составлению алгоритмов и программ следует придерживаться следующих рекомендаций:

- вникнуть в постановку задачи;
- определить, какая информация является исходной для решения поставленной задачи;
- уяснить, что требуется получить в результате решения задачи, в какой форме должны быть представлены выходные данные;
- разработать алгоритм решения задачи (для сложных задач алгоритм можно составлять в несколько этапов: сначала в укрупнённом виде, а затем детализировать каждый из шагов);
- необходимо проверить каждый шаг алгоритма вручную, (на небольшом наборе исходных данных) и убедиться в его правильности. При этом следует буквально выполнять записанные действия, не заменяя их на желаемые;
- алгоритм можно записывать в виде псевдопрограммы или блок-схемы, но обязательно с использованием структурности, т.е. используя основные канонические структуры, соблюдая правила вложенности действий, добиваясь регулярности программ;
- после проверки правильности работы алгоритма на примере составить программу по разработанному алгоритму (закодировать алгоритм);
- обязательным шагом в разработке программ является тестирование – проверка работоспособности программы на контрольном примере с последующим анализом результатов;
- при описании алгоритма и программы необходимо делать пояснения к выполненным шагам, к введённым переменным.

Как общие, так и индивидуальные задания студент может выполнять самостоятельно.

Во время проведения практического занятия по определённой теме вначале предлагается разобрать готовое решение задачи, набрать и отладить одинаковую для всех студентов группы задачу; затем — самостоятельно решить несколько однотипных задач по индивидуальному заданию. Данный подход позволяет устранить зависимость сроков выполнения заданий от индивидуальных способностей студента к программированию.

Выполнив индивидуальную работу, студент оформляет отчет, в котором содержатся следующие разделы:

- содержательная постановка задачи (включает формулировку условия задачи и пример решения);
- формализованная постановка задачи (описывающая способ получения результатов по исходным данным, выбор и описание метода решения задачи, исходные данные и форму их представления, форму представления результатов выполнения программы, перечень исключительных ситуаций и требуемой реакции программы);
- алгоритмизация (представление алгоритма в виде блок-схемы или псевдопрограммы с необходимыми пояснениями);
- программирование и отладка (содержит тексты программ и подпрограмм);
- испытание программы (на различных наборах данных);
- заключение по работе с анализом полученных результатов.

При защите отчета необходимо отвечать на контрольные вопросы и уметь пояснять работу программы.

2. ОСНОВНЫЕ ПОНЯТИЯ ЯЗЫКА PASCAL

В алфавит языка Pascal входят: прописные и строчные буквы латинского алфавита; цифры от 0 до 9; символ подчёркивания « ».

Из этих символов конструируются идентификаторы – имена типов, переменных, констант, процедур, функций и модулей, а также меток перехода. Имя может состоять из разного количества символов, но обязательно должно начинаться с буквы, причём прописные и строчные не различаются.

Кроме перечисленных в алфавит входит символ « » (пробел), который является разделителем в языке Pascal.

При написании значений символьных и строчных констант могут участвовать управляющие символы с кодами ASCII.

При построении конструкция языка используются следующие специальные символы +, -, *, /, =, <, >, [,], (,), @, {, }, ;, :, ', # (номер), \$ (знак денежной единицы), ^ (тильда), пробел, точка и запятая, а также составные символы, воспринимаемые как один символ: <=, >=, :=, (*, *), (., .), .. Разделители между элементами составных символов недопустимы.

В языке Pascal имеется большое количество зарезервированных слов, которые нельзя использовать в качестве идентификаторов в программе. Зарезервированными являются следующие слова: *and, array, begin, case, const, div, do, downto, else, end, file, for, forward, function, goto, if, in, lable, mod,*

nil, not, of, or, packed, procedure, program, record, repeat, set, shl, shr, string, then, to, type, unit, until, uses, var, while, with, xor.

Общая структура программы на языке Pascal:

```
Program< имя программы >;
Uses< Список используемых модулей > ;
Label< Список меток в основном блоке программы > ;
Const< Определение констант >;
Type< Описание типов > ;
Var< Определение глобальных переменных программы > ;
Procedure< Определение процедур > ;
Function< Определение функций > ;
Begin
<Основной блок программы >
End.
```

Все данные (в зависимости от способа их хранения и обработки) можно разделить на две группы – константы и переменные.

Константами называются элементы данных, значения которых установлены в описательной части программы и в процессе выполнения программы не изменяются.

Константы могут быть целочисленные, вещественные, символьные и строковые.

Формат описания констант:

```
Const< Идентификатор>=значение;
```

Переменные, в отличие от констант, могут менять свои значения в процессе выполнения программы. Тип констант автоматически распознаётся компилятором без предварительного описания. Тип переменной должен быть описан перед тем, как с переменными будут выполняться какие-либо действия.

Формат описания переменных:

```
Var<Идентификатор>: тип;
```

Арифметические выражения строятся из имен переменных, констант, знаков операций и круглых скобок так, как это принято в математике. При вычислении их значений операции выполняются в порядке приоритета: *, /, DIV, MOD, а затем + и -. Операции одинакового старшинства выполняются слева направо.

Знаки арифметических операций следующие:

```
+ - сложение;
- - вычитание;
* - умножение;
/ - деление;
```


Div – целочисленное деление;

Mod – остаток от целочисленного деления.

Наряду с переменными и константами в арифметические выражения можно включать функции. При определении значения выражения, прежде всего, вычисляются значения входящих в него функций. Список стандартных математических функций языка Pascal приведён в таблице 1.

Таблица 1. Стандартные математические функции

Функция	Назначение
ABS (X)	Модуль числа x
ARCTAN (X)	Arctgx
COS (X)	cosx
EXP (X)	ex
FRAC (X)	Дробная часть x
INT (X)	Целая часть x
LN (X)	Ln x , $x > 0$
PI	Значение числа $\pi = 3.1415926535897932385$
ROUND (X)	Округление до ближайшего целого
SIN (X)	sin x
SQR (X)	Значение квадрата числа x
SQRT (X)	Квадратный корень из числа x, $x > 0$
TRUNC (X)	Целая часть значения x
RANDOM	Случайное число (0...1)
RANDOM (X)	Случайное число (0...X)

Аргумент функции обязательно заключается в скобки.

При записи выражений на языке Pascal для сохранения порядка действий используются круглые скобки.

II. ТЕМАТИЧЕСКОЕ СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

1. ОРГАНИЗАЦИЯ ВВОДА И ВЫВОДА НА ЯЗЫКЕ PASCAL

Цель работы: научиться осуществлять ввод и вывод данных различных типов, применяя различные виды расположения данных.

Методические рекомендации:

Для ввода с клавиатуры во время работы программы исходных данных (значений переменных) предназначена инструкция *read (readln)*.

Используя одну инструкцию *read (readln)*, можно ввести значения нескольких переменных. Необходимо помнить, что тип данных, вводимых во время работы программы, должен соответствовать типу переменной, указанной в инструкции *read (readln)*; в случае несоответствия типа введенных данных типу переменной, значение которой вводится с клавиатуры, программа завершает работу и на экран выводится сообщение об ошибке.

Для организации вывода на языке Pascal служит оператор *write (writeln)*. Одна инструкция *write (writeln)* может вывести на экран значения нескольких переменных и (или) несколько сообщений. Инструкция *writeln* без параметров переводит курсор в начало следующей строки экрана.

Для того, чтобы задать цвет фона и цвет символов, выводимых инструкциями *write* и *writeln*, можно использовать процедуры, содержащиеся в стандартной библиотеке *crt*.

Чтобы программе были доступны процедуры и функции библиотеки *crt*, первой инструкцией программы должна быть инструкция *uses crt*.

Цвет символов устанавливает процедура *TextBackGround*, цвет фона – процедура *Textcolor*.

При использовании процедур *TextBackGround* и *Textcolor* цвет можно задать с помощью именованной или целой константы.

С помощью процедуры *ClrScr* очищается текущее окно на экране.

Пример

Написать программу ввода двух чисел (x и y) целого типа и вывода частного от деления первого числа на второе (второе число должно быть отлично от нуля), в дробной части вывести три знака.

Программа на языке Pascal может быть следующей:

```
Program PR1;  
  Var x,y:integer;  
  Begin  
    Writeln ('Введите два числа');  
    Readln (x,y);
```

Writeln ('Частное от деления числа ', x, ' на число ', y, ' = ', x/y:8:3);
End.

Контрольные вопросы:

1. В чём различие инструкций вывода *write* и *writeln*?
2. В чём различие инструкций ввода *read* и *readln*?
3. Каким образом отделяются друг от друга подготовленные числовые данные?
4. Как указывается количество знаков в дробной части при выдаче результатов?
5. Каким образом будет выведено число вещественного типа без указания формата вывода?

ЗАДАЧИ

1. Написать программу, которая выводит на экран Ваши имя и фамилию.
2. Написать программу, которая выводит на экран на разных строках три сообщения 'Доброе утро', 'Добрый день' и 'Добрый вечер'.
3. Написать программу, которая выводит на экран четверостишие:
Унылая пора! Очей очарованье!
Приятна мне твоя прощальная краса –
Люблю я пышное природы увяданье,
В багрец и золото одетые леса.
А. С. Пушкин
4. Написать инструкцию вывода значения переменной *a* (тип *real*) с тремя цифрами в дробной части.
5. Определите, что делают инструкции: *write (a)*; *write ('a')*.
6. Написать инструкцию, которая выводит значения дробных переменных *a*, *b* и *d* в одной строке. Каждое число должно быть выведено с двумя знаками после десятичной точки.
7. Написать инструкции вывода значений переменных *a*, *d* и *c*. Значение каждой переменной должно быть выведено на отдельной строке.
8. Написать инструкции вывода значений дробных переменных *x1* и *x2*. На экране перед значением переменной должен быть выведен поясняющий

текст, представляющий собой имя переменной, за которым следует знак "равно".

9. Написать программу, которая выводит на синем фоне серыми буквами четверостишие:

Буря мглою небо кроет,
Вихри снежные крутя.
То как зверь она завоет,
То заплачет, как дитя.

А. С. Пушкин

10. Написать программу, которая выводит на экран фразу: «*Каждый охотник желает знать, где сидят фазаны*», позволяющую запомнить порядок следования цветов радуги (первая буква слова кодирует цвет: каждый — красный, охотник — оранжевый, желает — желтый, знает — зеленый, где — голубой, сидят — синий, фазаны — фиолетовый). Каждое слово фразы должно быть выведено наиболее подходящим цветом.

11. Написать инструкцию, обеспечивающую во время работы программы ввод значения переменной `radius`.

12. Написать инструкции, которые обеспечивают ввод значений переменных целого типа `u` и `r`. Предполагается, что во время работы программы пользователь будет после набора каждого числа нажимать клавишу `<Enter>`.

13. Написать инструкцию, которая обеспечивает ввод значений переменных целого типа `a` и `b`. Предполагается, что во время работы программы пользователь будет набирать числа в одной строке.

14. Объявите необходимые переменные и напишите фрагмент программы вычисления объема цилиндра, обеспечивающий ввод исходных данных.

15. Объявить необходимые переменные и написать инструкции ввода исходных данных для программы вычисления стоимости покупки нескольких тетрадей и карандашей. Предполагается, что во время работы программы пользователь будет вводить данные о каждой составляющей покупки в отдельной строке: сначала - цену, затем - количество.

16. Написать программу, в которой осуществляется ввод трёх вещественных чисел, подготовленных на одной строке, и вывод их с указанием четырёх знаков в дробной части.

17. Написать программу ввода двух чисел целого типа и одного – вещественного. На первой строке вывести целые числа, на второй – вещественное с двумя знаками в дробной части.

18. Ввести два числа вещественного типа. На первой строке выдать эти числа с одним знаком в дробной части, на второй строке – с двумя знаками в дробной части.

19. Ввести три числа целого типа, подготовленные на одной строке. Вывести эти числа в обратном порядке, по одному числу на строке.

20. Ввести два числа вещественного типа, подготовленные на разных строках. Вывести эти числа с тремя знаками в дробной части на одной строке.

21. Ввести два числа вещественного типа, два – целого типа. Числа подготовлены на одной строке. Выдать на первой строке целые числа, на второй строке – вещественные (с двумя знаками в дробной части).

22. Ввести три числа целого типа, подготовленные на разных строках. Вывести эти числа на одной строке, отделяя выводимые значения пробелом.

23. Ввести два числа целого типа, вывести их сумму и разность, предваряя значения пояснениями.

24. Ввести два числа вещественного типа. Вывести их сумму и произведение (с двумя знаками в дробной части), предваряя значения пояснениями.

25. Ввести три числа вещественного типа. Выдать на первой строке их сумму (с одним знаком в дробной части), на второй строке – их произведение (с тремя знаками в дробной части), предваряя значения пояснениями.

2. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

Цель работы: научиться создавать, выполнять и исправлять простейшие программы на языке Pascal в режиме диалога, познакомиться с диагностическими сообщениями компилятора об ошибках при выполнении программ, реализующих линейные алгоритмы.

Методические рекомендации:

Линейным называется алгоритм, в котором результат получается путем однократного выполнения заданной последовательности действий при любых значениях исходных данных. Операторы программы выполняются последовательно, один за другим, в соответствии с их расположением в программе.

Пример

Определить площадь поверхности куба, если известна длина ребра.

Математическая модель. Площадь поверхности куба (s) вычисляется по формуле $s=6*a*a$, где a —длина ребра куба.

Алгоритм в виде псевдопрограммы выглядит следующим образом:

1. Начало.
2. Ввод a
3. $s=6*a*a$
4. Вывод s
5. Конец

Программа на языке Pascal может быть следующей:

```
Program PR2;  
Var s,a:integer;  
Begin  
Writeln ('Введите значение ребра куба');  
Readln (a);  
S:=6*a*a;  
Writeln ('Площадь поверхности куба = ', s);
```

End.

Контрольные вопросы:

1. Каковы назначение и возможности системы программирования Borland Pascal?
2. Как запустить программу на трансляцию и выполнение?
3. Как записываются операторы начала и конца программы?
4. Из каких разделов состоит программа на языке Pascal?
5. В какой последовательности должны быть записаны разделы программы на языке Pascal?
6. Как записываются операторы вывода на экран в Pascal?

ЗАДАЧИ

1. Написать программу вычисления площади параллелограмма, если вводятся длина и ширина параллелограмма.

2. Написать программу вычисления объема параллелепипеда, если вводятся длина, ширина и высота параллелепипеда.

3. Написать программу вычисления площади поверхности параллелепипеда, если вводятся длина, ширина и высота параллелепипеда.

4. Написать программу вычисления объема куба, если осуществляется ввод длины ребра куба.

5. Написать программу вычисления объема цилиндра, если вводится радиус основания и высота цилиндра.

6. Написать программу вычисления стоимости покупки, состоящей из нескольких тетрадей и карандашей. Пользователь осуществляет ввод количества тетрадей и ручек, а также цену каждого из однотипных товаров.

7. Написать программу вычисления стоимости покупки, состоящей из нескольких тетрадей и такого же количества обложек к ним, если необходимо ввести цены тетради и обложки, а также количество комплектов.

8. Написать программу вычисления стоимости некоторого количества (по весу) яблок. Предусмотреть ввод цены 1 кг яблок, вес яблок.

9. Написать программу вычисления площади треугольника, если известна длина основания и высота.

10. Написать программу вычисления площади треугольника, если известны длины двух его сторон и величина угла между этими сторонами.

11. Написать программу вычисления стоимости поездки на автомобиле на дачу (туда и обратно). Исходными данными являются: расстояние до дачи (в километрах); количество бензина, которое потребляет автомобиль на 100 км пробега; цена одного литра бензина.

12. Написать программу пересчета расстояния изверст в километры (1 верста равняется 1066,8 м).

13. Написать программу пересчета веса из фунтов в килограммы (1 российский фунт равен 409,5 г).

14. Написать программу вычисления величины дохода по вкладу. Величина вклада, процентная ставка (в процентах годовых) и время хранения (в днях) задаются во время работы программы.

15. Написать программу пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах.

16. Написать программу, которая преобразует введенное с клавиатуры дробное число в денежный формат.

17. Написать программу пересчета из фунтов в килограммы (1 фунт равняется 409,5 г).

18. Написать программу, которая вычисляет площадь треугольника, если известны координаты его углов.

19. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.

20. Треугольник задан координатами своих вершин. Найти периметр треугольника и площадь треугольника.

21. Вычислить высоту треугольника, опущенную на сторону a , по известным значениям длин его сторон a , b , c .

22. Вычислить объем цилиндра с радиусом основания r и высотой h .

23. Вычислить площадь треугольника по формуле Герона, если заданы его стороны.

24. По данным сторонам прямоугольника вычислить его периметр, площадь и длину диагонали.

25. По заданным катетам прямоугольного треугольника найти гипотенузу и площадь треугольника.

3. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ

Цель работы: научиться правильно использовать условный оператор *if*; научиться составлять программы решения задач на разветвляющиеся алгоритмы.

Методические рекомендации:

Алгоритм называется разветвляющимся, если он содержит несколько ветвей, отличающихся друг от друга содержанием вычислений. Выход вычислительного процесса на ту или иную ветвь алгоритма определяется исходными данными задачи.

Перед выполнением работы необходимо ознакомиться с правилами записи логических выражений, операций сравнения, операторов *if, case*.

Приступая к решению задач этого раздела, следует вспомнить, что:

- инструкция *if* используется для выбора одного из двух направлений дальнейшего хода программы (последовательности инструкций, которые должны быть выполнены);
- выбор последовательности инструкций осуществляется во время выполнения программы в зависимости от выполнения условия;
- условие – это выражение логического типа, которое может принимать одно из двух значений: *true* (истина – условие выполняется) или *false* (ложь – условие не выполняется);
- при помощи логических операций *And* (логическое «И») и *Or* (логическое «ИЛИ») из простых условий можно строить сложные.

Пример

Даны два числа *a* и *b*. Получите число *max*, равное наибольшему из исходных. Если исходные числа равны, то *max* надо присвоить значение любого из них.

Решение

Для решения задачи нужно ввести значения переменных *a* и *b*, сравнить их значения и, в зависимости от результата сравнения, *max* положить равным *a* или *b*.

Следует обратить внимание на то, что вместе с результатом выводятся и исходные данные. Это позволит упростить тестирование и повысить наглядность вывода.

Псевдопрограмма может выглядеть следующим образом:

1. Начало
2. Ввод *a* и *b*
3. Если *a > b* то *max := a*
4. иначе *max := b*.
5. Вывод *a, b, max*

6. Конец

Программа на языке Pascal может быть следующей:

```
program PR3;  
{ Даны два числа a и b. Получите число max, равное наибольшему из  
исходных. }  
var  
  a,b, max:integer;  
begin  
  write ('Введите два числа через пробел ');  
  readln (a,b);  
  if a>=b then  max:=a  
  else  max:=b;  
  writeln (' a= ',a,' b= ',b,' max= ',max);  
end.
```

Контрольные вопросы:

1. Какие команды текстового редактора вы знаете?
2. Что такое блок текста и как его выделить?
3. Какие операторы используются для программирования разветвлений?
4. Как выполняются операторы перехода?
5. Какую из функций: Sin(x), Abs(x), Trunc(x) можно заменить условным оператором if $x < 0$ then $x := -x$?

ЗАДАЧИ

1. Написать программу вычисления площади кольца, где исходными данными являются радиусы кольца и отверстия. Предусмотреть в программе проверку правильности исходных данных (радиус отверстия не может быть больше радиуса кольца).

2. Написать программу решения квадратного уравнения. Программа должна проверить правильность исходных данных и в случае, когда коэффициент при второй степени неизвестного равен нулю, выводить соответствующее сообщение.

3. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 10% предоставляется, если сумма покупки больше 10000тенге.

4. Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется в том случае, если сумма покупки больше 500тенге, в 5% — если сумма больше 1000 тенге.

5. Написать программу, которая вычисляет частное от деления двух чисел. Программа должна проверять правильность введенных пользователем данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке.

6. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу (1, 3).

7. Даны действительные числа x , y . Если x , y отрицательны, то каждое значение заменить его модулем; если отрицательное только одно из них, то оба значения увеличить на 0.5; если оба значения не отрицательны и ни одно из них не принадлежит отрезку $[0.5, 2.0]$, то оба значения уменьшить в 10 раз; в остальных случаях x , y оставить без изменения.

8. Определить и вывести на печать номер квадранта, в котором расположена точка $M(x,y)$; x и y - заданные вещественные числа.

9. Из величин, определяемых выражениями $a=\sin x$, $b=\cos x$, $c=\ln|x|$ при заданном x , определить и вывести на экран дисплея минимальное значение.

10. Определить, какая из двух точек - $M_1(x_1,y_1)$ или $M_2(x_2,y_2)$ - расположена ближе к началу координат. Вывести на экран дисплея координаты этой точки.

11. Определить, какая из двух фигур (круг или квадрат) имеет наибольшую площадь. Известно, что сторона квадрата равна a , радиус круга — r . Вывести на экран название и значение площади большей фигуры.

12. Определить, попадает ли точка $M(x,y)$ в круг радиусом r с центром в точке (x_0,y_0) .

13. Написать программу, запрашивающую возраст пользователя. Если ему не менее 18 лет, то следует сообщить, что он имеет право голосовать, в противном случае надо вычислить, через сколько лет ему будет предоставлено это право.

14. Написать программу, которая спрашивала бы сокращенное имя, а печатала полное для пяти друзей. Ввод незнакомого имени должен провоцировать заявление типа «Я С ВАМИ НЕЗНАКОМА». Ответ, содержащий сокращенное имя, помещается в символьную переменную, а затем сравнивается последовательно с сокращенными именами пяти знакомых. Если введенное имя совпадает с каким-либо из использованных в программе, то вывести соответствующее полное имя.

15. Составить программу для определения подходящего возраста кандидатуры для вступления в брак, используя следующее соображение: возраст девушки равен половине возраста мужчины +7, возраст мужчины определяется соответственно как удвоенный возраст девушки -14. В начале выполнения программы на экране должен появиться вопрос МУЖЧИНА или ЖЕНЩИНА? ВВЕДИТЕ М, если Вы мужчина, или Ж, если женщина, и свой возраст. В зависимости от ответа выводятся соответствующие рекомендации.

16. Заданы координаты двух точек. Определить, расположены ли они на одной окружности с центром в начале координат. Результат присвоить символьной переменной. Две точки находятся на одной окружности, если длины радиус-векторов, соединяющих эти точки с началом координат, равны (проверку на равенство осуществить с точностью до $\epsilon=1.0E-3$).

17. Написать программу, которая по введённому номеру месяца в году определяет время года.

18. Написать программу, которая по номеру дня недели выводит название дня.

19. В программе вводятся радиус круга и сторона квадрата. Необходимо проверить, пройдет ли квадрат в круг.

20. Написать программу, которая по номеру месяца определяет название месяца и квартал года.

21. Даны три числа. Написать программу, определяющую, можно ли построить треугольник с заданными сторонами.

22. Даны длины сторон треугольника. Определить, является ли данный треугольник равнобедренным?

23. Даны длины сторон треугольника. Определить, является ли данный треугольник разносторонним?

24. Даны в градусах величины двух углов треугольника. Определить, является ли данный треугольник равнобедренным?

25. Треугольник задан координатами вершин. Определить, принадлежит ли точка с заданными координатами области треугольника.

4. ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ

Цель работы: закрепить практические навыки работы с системой Borland Pascal, научиться правильно использовать различные операторы циклов; научиться составлять программы решения задач с использованием циклических структур.

Методические рекомендации:

Алгоритм называется циклическим, если он содержит многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений этих операторов может быть задано в явной (цикл с известным заранее числом повторений) или неявной форме (цикл с неизвестным заранее числом повторений).

Перед выполнением работы необходимо изучить различные схемы организации циклов и операторы For, While, Repeat.

Особенности применения конструкции **For** заключаются в том, что:

- инструкция For используется для организации циклов с фиксированным, определяемым во время разработки программы, числом повторений;
- количество повторений цикла определяется начальным и конечным значениями переменной-счетчика;
- переменная-счетчик должна быть целого типа (integer).

При использовании конструкции оператора цикла «с предусловием» **While** необходимо учитывать то, что:

- число повторений инструкций цикла while определяется ходом выполнения программы;
- инструкции цикла while выполняются до тех пор, пока условие, записанное после слова while, истинно;
- после слова while надо записывать условие выполнения инструкций цикла;
- для завершения цикла while в теле цикла обязательно должны присутствовать инструкции, влияющие на условие выполнения инструкций цикла;

- цикл `while` — это цикл с предусловием, т. е. инструкции тела цикла вообще могут быть не выполнены;

- цикл `while`, как правило, используется для организации приближенных вычислений, задач поиска и обработки данных, вводимых с клавиатуры или из файла.

Применяя конструкцию оператора цикла «с постусловием» ***Repeat***, следует учесть, что:

- число повторений инструкций цикла `repeat` определяется ходом выполнения программы;

- инструкции цикла `repeat` выполняются до тех пор, пока условие, стоящее после слова `until`, ложно;

- после слова `until` надо записывать условие завершения цикла;

- для завершения цикла `repeat` в теле цикла обязательно должны быть инструкции, выполнение которых влияет на условие завершения цикла;

- цикл `repeat` — это цикл с постусловием, т. е. инструкции тела цикла будут выполнены хотя бы один раз;

- цикл `repeat`, как правило, используется для организации приближенных вычислений, задач поиска и обработки данных, вводимых с клавиатуры или из файла.

Пример

Составить программу, находящую сумму значений функции $\sin(x)$ для x , изменяющегося от 0 до 10π с шагом $\pi/3$.

Обозначим через S переменную, в которой хранится сумма значений функции $\sin(x)$.

Псевдопрограмму решения данной задачи можно представить следующим образом:

1. Начало.
2. $X:=0$
3. $S:=0$
4. Цикл пока $X \leq 10\pi$
5. $S:=S + \sin(X)$
6. $X:=X+\pi/3$
7. Конец цикла
8. Печать S
9. Конец

Программу для решения данной задачи можно написать, используя различные операторы циклов.

1 вариант программы предусматривает применение оператора цикла «с предусловием» *While*. В этом случае программа на языке Pascal имеет следующий вид:

```
program PR4;
var S,X: real;
Begin
  X:=0;
  S:=0;
  While X<=10*Pi do
    Begin
      S:=S+Sin(X);
      X:=X+Pi/3;
    End;
  Writeln (' Сумма элементов последовательности = ',S:6:2);
End.
```

2 вариант программы написан с использованием оператора цикла «с постусловием» *Repeat*.

```
program PR5;
var S,X: real;
Begin
  X:=0;
  S:=0;
  Repeat
    S:=S+Sin(X);
    X:=X+Pi/3;
  Until X>10*Pi;
Writeln (' Сумма элементов последовательности = ',S:6:2);
End.
```

Контрольные вопросы:

1. Как записывается и как работает оператор FOR?
2. Для организации каких циклов применим оператор FOR?
3. В чем отличие оператора WHILE от оператора REPEAT?
4. Как программируются циклические алгоритмы с явно заданным числом повторений цикла?
5. Как программируются циклические алгоритмы с заранее неизвестным числом повторений цикла?
6. Как записать оператор цикла, который не выполняется ни разу?
7. Как записать оператор цикла, который выполняется неограниченное число раз?

8. Как заменить оператор "Repeat A Until B" равносильным фрагментом программы с оператором While?

ЗАДАЧИ

1. Написать программу, которая выводит таблицу квадратов первых десяти целых положительных чисел.

2. Написать программу, которая выводит таблицу квадратов первых пяти целых положительных нечетных чисел.

3. Написать программу, которая вычисляет сумму первых n целых положительных целых чисел. Количество суммируемых чисел должно вводиться во время работы программы.

4. Написать программу, которая вычисляет сумму первых n членов ряда 1, 3, 5, 7, Количество суммируемых членов ряда задается во время работы программы.

5. Написать программу, которая выводит таблицу степеней двойки (от нулевой до десятой).

6. Написать программу, которая вычисляет факториал числа, введенного с клавиатуры. (Факториалом числа n называется произведение целых чисел от 1 до n).

7. Написать программу, которая выводит таблицу значений функции $y = -2,4x^2 + 5x - 3$ в диапазоне от -2 до 2 с шагом 0,5.

8. Написать программу, которая вводит с клавиатуры 5 дробных чисел и вычисляет их среднее арифметическое.

9. Написать программу, которая вычисляет среднее арифметическое вводимой с клавиатуры последовательности дробных чисел. Количество чисел должно задаваться во время работы программы.

10. Написать программу, которая вводит с клавиатуры последовательность из пяти дробных чисел и после ввода каждого числа выводит среднее арифметическое полученной части последовательности.

11. Написать программу, которая вычисляет среднее арифметическое последовательности дробных чисел, вводимых с клавиатуры. После того, как будет введено последнее число, программа должна вывести минимальное и максимальное число последовательности. Количество чисел должно задаваться во время работы программы.

12. Написать программу, которая генерирует 10 случайных чисел в диапазоне от 1 до 10, выводит эти числа на экран и вычисляет их среднее арифметическое.

13. Написать программу, которая генерирует три последовательности из десяти случайных чисел в диапазоне от 1 до 10, выводит каждую последовательность на экран и вычисляет среднее арифметическое каждой последовательности.

14. Написать программу, которая выводит на экран таблицу стоимости, например, яблок в диапазоне от 100 г до 1 кг с шагом 100.

15. Написать программу, которая выводит таблицу значений функции $y=|x|$. Диапазон изменения аргумента — от -4 до 4, шаг приращения аргумента — 0,5.

16. Написать программу, которая выводит таблицу значений функции $y=|x+2|$. Диапазон изменения аргумента — от -4 до 4, шаг приращения аргумента — 0,5.

17. Написать программу, которая выводит таблицу значений функции $y=|x-2|+|x+1|$. Диапазон изменения аргумента — от -4 до 4, шаг приращения аргумента — 0,5.

18. Написать программу, которая выводит на экран таблицу умножения на вводимое пользователем число.

19. Написать программу, которая выводит двоичное представление введенного с клавиатуры десятичного целого числа в диапазоне от 0 до 255.

20. Написать программу, которая выводит на экран квадрат Пифагора — таблицу умножения.

21. Напишите программу приближенного вычисления интеграла функции $f(x) = 5x^2 - x + 2$ методом прямоугольников.

22. Напишите программу приближенного вычисления интеграла методом трапеций.

23. Напишите программу, которая выводит на экран изображение шахматной доски. Черные клетки отображать "звездочкой", белые — пробелом.

24. Написать программу, которая проверяет, является ли целое число, введенное пользователем, простым.

25. Написать программу, которая вычисляет наибольший общий делитель двух целых чисел.

5. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ МАССИВОВ

Цель работы: научиться правильно описывать различные массивы, уметь инициализировать массивы, распечатывать содержимое массива; научиться решать задачи на использование массивов.

Методические рекомендации:

Массив - это структурированный тип данных, который используется для описания упорядоченной совокупности фиксированного числа элементов одного типа, имеющих общее имя. Для обозначения элементов массива используются имя переменной-массива и индекс.

Массив должен быть объявлен в разделе объявления переменных. При объявлении массива удобно использовать именованные константы. Доступ к элементу массива осуществляется путем указания индекса (номера) элемента, в качестве которого можно использовать выражение целого типа, например, целую константу или переменную типа `integer`.

Ввод и вывод элементов массива осуществляется поэлементно.

Для ввода, вывода и обработки массивов удобно применять инструкции циклов (`for`, `while`, `repeat`).

Перед выполнением работы необходимо изучить правила описания и использования переменных типа массив, типизированных констант типа массив.

Пример

Составить программу нахождения суммы элементов данного массива А, состоящего из n элементов.

Основная часть цикла – добавление очередного элемента к накапливаемой сумме. Для нахождения суммы всех элементов достаточно повторить цикл, изменяя номер элемента массива от 1 до n с шагом 1.

Алгоритм в виде псевдопрограммы имеет следующий вид:

1. Начало
2. Ввод элементов массива А
3. S:=0
4. I:=1
5. Цикл пока ($I \leq n$)
6. S:=S+A[i]
7. I:=I+1
8. Конец цикла
9. Вывод s
10. Конец

Программа на языке Pascal может быть следующей:

```
Program PR6;
Program PR6;
  Const  maxn=100; {Максимальное количество элементов массива}
  var  n,i:integer;
       s:real;
       A:array [1..maxn] of real;
begin
  write ('n= ');
  readln (n); {Ввод реального количества элементов массива}
  writeln ('Введите элементы массива A ');
  for i:=1 to n do
    begin
      write ('A [' ,i,']= ');
      readln (A [i]);
    end;
  S:=0;
  for i:=1 to n do
    S:=S+A[i];
  Writeln ('Исходный массив a, состоящий из ',n:4,' элементов');
  for i:=1 to n do
    write (A[i]:8:2);
  writeln;
```

```
writeln ('Сумма= ',S:8:2);  
end.
```

Контрольные вопросы:

1. Каким образом определяются переменные типа массив (одномерный и двумерный)?
2. Как осуществляется доступ к отдельному элементу одномерного и двумерного массива?
3. Каким образом выводятся элементы массива на экран?
4. Как осуществляется ввод элементов массива?
5. Каким образом можно осуществить вывод элементов двумерного массива на экран в виде матрицы?

ЗАДАЧИ

1. Написать программу, которая вводит с клавиатуры одномерный массив из N целых чисел и выводит количество ненулевых элементов. Перед вводом каждого элемента на экране должна появляться подсказка с его номером.

2. Написать программу, которая выводит минимальный элемент введенного с клавиатуры массива целых чисел.

3. Написать программу, которая вычисляет среднее арифметическое ненулевых элементов введенного с клавиатуры массива целых чисел.

4. Написать программу, которая вычисляет среднюю (за неделю) температуру воздуха. Исходные данные должны вводиться во время работы программы в одномерный массив.

5. Написать программу, которая проверяет, находится ли в массиве введенное с клавиатуры число. Массив должен вводиться во время работы программы.

6. Написать программу, которая проверяет, представляют ли элементы введенного с клавиатуры массива возрастающую последовательность.

7. Написать программу, которая проверяет, образуют ли элементы введенного с клавиатуры массива неубывающую последовательность.

8. Написать программу, которая вычисляет, сколько раз введенное с клавиатуры число встречается в массиве.

9. Написать программу, которая проверяет, есть ли во введенном с клавиатуры массиве элементы с одинаковым значением.

10. Написать программу, которая методом прямого выбора сортирует по убыванию введенный с клавиатуры одномерный массив.

11. Написать программу, которая методом обмена («пузырька») сортирует по убыванию введенный с клавиатуры одномерный массив.

12. Написать программу, которая объединяет два упорядоченных по возрастанию массива в один, также упорядоченный по возрастанию массив.

13. Написать программу, которая, используя метод бинарного поиска, выполняет поиск в упорядоченном по возрастанию массиве.

14. Написать программу, которая определяет количество студентов в группе, чей рост превышает средний. Данные о росте каждого студента занесены в одномерный массив.

15. Дан массив A , состоящий из n -элементов. Написать программу нахождения суммы элементов массива, стоящих на нечетных местах.

16. Дан массив A , состоящий из n -элементов. Написать программу нахождения суммы элементов массива, стоящих на четных местах.

17. Написать программу изменения расположения элементов данного массива в обратном порядке (первый элемент меняется с последним, второй - с предпоследним и т.д. до середины; если массив содержит нечетное количество элементов, то средний остается без изменения).

18. Написать программу, изменяющую в данном массиве порядок следования элементов таким образом, чтобы местами элементы, стоящие на нечетных местах, поменялись с элементами, стоящими на четных местах.

19. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить сумму модулей отрицательных элементов массива.

20. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить количество элементов массива, не принадлежащих интервалу (a, b) .

21. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить наименьший из элементов массива, принадлежащих отрезку $[a, b]$.

22. Составить программу, позволяющую в одномерном массиве, состоящем из N вещественных элементов, вычислить количество элементов массива, равных первому элементу.

23. Составить программу вычисления суммы тех элементов одномерного массива, номера которых совпадают со значениями элементов массива.

24. Составить программу, в которой умножаются на значение максимального элемента все числа, предшествующие первому по порядку элементу с наибольшим значением.

25. Составить программу замены наибольшего значения одномерного массива на число, вводимое с клавиатуры.

26. Написать программу, которая проверяет, является ли введенная с клавиатуры квадратная матрица магическим квадратом. (Магическим *квадратом* называется матрица, сумма элементов которой в каждой строке, в каждом столбце и по каждой диагонали одинакова.)

27. Написать программу, которая вводит с клавиатуры двумерный массив по строкам и вычисляет среднее арифметическое его элементов.

28. Написать программу, которая вводит по строкам с клавиатуры двумерный массив и вычисляет сумму его элементов по столбцам.

29. Написать программу, которая вводит по строкам с клавиатуры двумерный массив и вычисляет сумму его элементов по строкам.

30. Написать программу, которая вычисляет суммы элементов квадратной матрицы, находящихся на главной и побочной диагоналях.

31. Написать программу, которая вычисляет определитель квадратной матрицы второго порядка.

32. Написать программу нахождения количества нулевых элементов квадратной матрицы, стоящих выше главной диагонали.

33. Написать программу нахождения количества нулевых элементов квадратной матрицы, стоящих ниже главной диагонали.

34. Написать программу нахождения количества нулевых элементов квадратной матрицы, стоящих выше побочной диагонали.

35. Написать программу нахождения количества нулевых элементов квадратной матрицы, стоящих ниже побочной диагонали.

36. Дана вещественная матрица размерности $n \times m$. Написать программу формирования вектора b , в котором элементы вычисляются как произведение элементов соответствующих строк.

37. Дана вещественная матрица размерности $n \times m$. Написать программу формирования вектора b , в котором элементы вычисляются как среднее арифметическое соответствующих столбцов.

38. Дана вещественная матрица размерности $n \times m$. Написать программу формирования вектора b , в котором элементы вычисляются как разность наибольших и наименьших элементов соответствующих строк.

39. Дана вещественная матрица размерности $n \times m$. Написать программу формирования вектора b , в котором находятся значения первых отрицательных элементов в столбце.

40. Дана вещественная матрица размерности $n \times m$. Написать программу формирования вектора b , в котором находятся значения первых отрицательных элементов в строке.

41. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами в каждом столбце наибольший по модулю и последний положительный элементы.

42. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами в каждом столбце первый и последний отрицательные элементы.

43. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами в каждой строке наибольший и наименьший элементы.

44. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами в каждом столбце первый элемент, принадлежащий отрезку $[a, b]$, и первый положительный элемент.

45. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами строку, содержащую максимум из элементов матрицы, со строкой, содержащей её минимум.

46. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой поменять местами две строки, номера которых вводятся дополнительно.

47. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой заменить на вводимое число элементы строк, содержащих отрицательные элементы.

48. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой определить номера строк, элементы которых упорядочены по возрастанию.

49. Дана вещественная матрица размерности $n \times m$. Написать программу, в которой определить номера столбцов, элементы которых упорядочены по убыванию.

50. Дана вещественная матрица размерности $n \times m$. Написать программу, определить номера строк, содержащих нулевые элементы.

6. ПРОГРАММИРОВАНИЕ ЗАДАЧ ОБРАБОТКИ СТРОКОВЫХ ДАННЫХ

Цель работы: познакомить с понятием «строковые данные», выработать навыки работы с символьной информацией в языке

программирования Pascal, научиться использовать строки символов и множества при решении задач.

Методические рекомендации:

Для обработки текстов или строк в языке Pascal существует два типа данных: Char и String.

Типу данных Char соответствуют символьные константы и переменные. Символьная константа – это символ алфавита, взятый в апострофы.

Все символы алфавита образуют множество литер. Каждый символ имеет свой код в ASCII. Это позволяет использовать операции сравнения: =, <>, <, <=, >, >=.

Переменные типа String аналогичны массивам типа Char. Их отличием является то, что число символов (длина строки) может динамически меняться в интервале от единицы до заданного верхнего значения. Получить доступ к отдельному символу строки можно, указав его номер в квадратных скобках после имени переменной (строки).

При организации работы с текстовыми данными используются следующие функции:

copy (s,p,n) – выделение n символов из строки s, начиная с позиции p;

concat (s1,s2,...sn) – соединение (слияние) строк в одну строку;

length (строка) – определение длины строки;

pos (s1,s2) – определение номера позиции строки s1 внутри строки s2 (если строка s1 не найдена, то результат равен 0).

Для изменения строковых данных применяются процедуры:

delete (s,p,n) – удаление n символов из строки s, начиная с позиции p;

insert (s1,s2,p) – вставка строки s1 в строку s2, начиная с позиции p.

Перед выполнением работы необходимо ознакомиться с правилами описания и использования строк, допустимых операций над ними, соответствующими стандартными процедурами и функциями.

Пример

Дан текст, слова в котором могут разделяться пробелами. Требуется напечатать все слова с удвоенной буквой «н».

Укрупнённая псевдопрограмма может быть организована следующим образом:

Начало

Ввод текста

Цикл пока (не конец текста)

Удаление пробелов до первого символа

Вычисление длины слова

Поиск буквы «н»

Подсчёт рядом стоящих букв «н»

Если число букв «н» =2, то

Печать слова

Удаление проанализированного слова

Конец цикла

Конец

Введём следующие переменные, обозначающие:

St – содержит введенный текст;

st1 – хранит первое слово текста;

i,j,k,n – вспомогательные переменные;

flag – указывает, что данное слово искомое.

Программа на языке Pascal может быть следующей:

```
program PR7;
```

```
var st, st1:string;
```

```
    i,j,k,n:integer;
```

```
    flag:boolean;
```

```
begin
```

```
    writeln ('Введитетекст');
```

```
    readln (st);
```

```
    repeat
```

```
        while st[1]= ' ' do delete (st,1,1);
```

```
        i:=1;
```

```
        while (not (st[i]=' ')) and (i<=length (st)) do i:=i+1;
```

```
        st1:=copy (st,1,i-1);
```

```
        flag:= false;
```

```
        while (pos ('н',st1)>0) and (not ( flag)) do
```

```
            begin
```

```
                j:=pos ('н',st1); n:=j; k:=0;
```

```
                while st1[n]='н' do begin n:=n+1; k:=k+1; end;
```

```
                if k=2 then flag:= true;
```

```
                delete (st1,j,k)
```

```
            end;
```

```
            if flag then writeln (copy (st,1,i-1));
```

```
            delete (st,1,i);
```

```
        until st="";
```

```
    readln;
```

```
end.
```

Контрольные вопросы:

1. Как описываются строковые переменные?
2. Какая максимальная длина строки допустима в Pascal?
3. Какие операции допустимы над строковыми данными?
4. В чем отличие строковой переменной от массива символов?
5. Какие существуют стандартные процедуры и функции для работы со строками?

ЗАДАЧИ

1. Написать программу, которая запрашивает имя пользователя и «здоровается» с ним.

2. Написать программу, которая удаляет начальные пробелы из введенной с клавиатуры строки.

3. Написать программу, которая проверяет, является ли введенная с клавиатуры строка целым числом.

4. Написать программу, которая проверяет, является ли введенная с клавиатуры строка двоичным числом.

5. Написать программу, которая проверяет, является ли введенная с клавиатуры строка шестнадцатеричным числом.

6. Написать программу, которая проверяет, является ли введенная с клавиатуры строка дробным числом.

7. Написать программу, которая исключает из вводимого текста символы, расположенные между скобками ‘(, ’). Сами скобки тоже должны быть исключены. Предполагается, что внутри каждой пары скобок нет других скобок.

8. Написать программу, которая преобразует текст по следующему правилу: если нет символа ‘*’, то оставить его без изменения, иначе заменить каждый символ, встречающийся после первого вхождения символа ‘*’, на символ ‘-’.

9. Написать программу, которая преобразует текст, заменяя все вхождения ‘abc’ на ‘def’.

10. Написать программу, которая изменяет вводимый текст, удаляя первое вхождение 'w', если такое есть.

11. Написать программу, которая удаляет из текста все вхождения символов 'th'.

12. Написать программу, которая в введённом тексте после каждой буквы 'g' добавляет букву 'u'.

13. Написать программу, которая находит количество цифр в введённом тексте.

14. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая преобразует текст, удаляя повторяющиеся слова.

15. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая определяет слова, встречающиеся только один раз.

16. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая определяет длину самого длинного слова.

17. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая находит слова, в которых первый и последний символ одинаковые.

18. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая находит симметричные слова.

19. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая находит слова, в которых первая буква слова встречается в нём ещё раз.

20. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать

программу, которая определяет слова, в которых буквы расположены по алфавиту.

21. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая преобразует текст, заменяя всякое вхождение слова 'это' на слово 'то'.

22. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая переставляет в словах первую и последнюю буквы.

23. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая удаляет из строки избыточные пробелы, чтобы между словами осталось только по одному пробелу.

24. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая определяет количество слов.

25. Дан текст. Группы символов, разделенные пробелами и не содержащие пробелы внутри себя, будем называть словами. Написать программу, которая находит самое короткое слово.

7. ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ ПРОЦЕДУР И ФУНКЦИЙ

Цель работы: познакомиться с понятиями «процедура» и «функция» в языке программирования Pascal, рассмотреть их сходства и различия, научиться применять метод последовательной детализации в практическом программировании; применять процедуры и функции при решении задач.

Методические рекомендации:

Часто в программе обнаруживаются однотипные участки, которые выполняют одни и те же вычисления, но с различными данными. Такие части программы целесообразно оформлять в виде подпрограмм. В языке Pascal существует два вида подпрограмм: процедуры и функции.

Для передачи данных в подпрограмму следует использовать только параметры (глобальные переменные, т. е. переменные, объявленные вне процедуры, применять не рекомендуется).

Среди выполняемых инструкций функции обязательно должна быть инструкция присваивания значения функции.

Тип каждого фактического параметра (константы или переменной) в инструкции вызова подпрограммы (процедуры или функции) должен быть таким же, как тип соответствующего формального параметра, указанного при объявлении подпрограммы.

Если в инструкции объявления подпрограммы (процедуры или функции) перед именем формального параметра не стоит слово `var`, то при вызове подпрограммы можно использовать в качестве формального параметра константу или переменную соответствующего типа. Если слово `var` указано, то формальным параметром может быть только переменная.

Если аргумент процедуры применяется для возврата результата в программу, вызвавшую эту процедуру, то перед именем аргумента нужно поставить слово `var`.

Перед выполнением данной работы необходимо изучить правила описания процедур и функций, механизм передачи параметров, ознакомиться с понятием локальной и глобальной переменной.

Пример

Ввести элементы квадратной матрицы (представленной в виде двумерного массива). Написать программу нахождения количества положительных элементов каждого столбца матрицы.

При разработке алгоритма следует учесть, что просмотр и нахождение количества положительных элементов столбца матрицы повторяется многократно и зависит от количества столбцов. Эти действия можно оформить в виде функции, а ввод и вывод элементов матрицы организовать в виде подпрограмм.

Укрупнённая псевдопрограмма решения данной задачи может выглядеть следующим образом:

Начало.

Вызов процедуры ввода элементов матрицы.

Цикл пока (не закончились столбцы)

Нахождение и вывод количества положительных элементов текущего столбца

Конец цикла

Конец

Введём следующие обозначения:

A – имя матрицы;

Mas – имя типа для двумерного массива;

Vvod – процедура ввода элементов матрицы;

Kolp – функция подсчёта количества положительных элементов столбца;
 N – размерность матрицы;
 I, j, M – локальные переменные;
 D – формальный (локальный) параметр, определяющий матрицу;
 K – переменная, определяющая текущий номер столбца матрицы;
 L – рабочая переменная для организации цикла по столбцам;
 kp – переменная для подсчёта количества положительных элементов.

Программа на языке Pascal может быть следующей:

```

program PR8;
const mr=100;
type  Mas = array [1..mr,1..mr] of integer;
Var A: Mas;
      N, L, kp: integer;
Procedure Vvod (var D: Mas; var M:integer);
  Vari,j:integer;
Begin
  Writeln ('Введите размерность матрицы');
  Read (M);
  Writeln ('Введите элементы матрицы размерностью ',M,' * ', M);
  For i:= 1 to M do
    For j:=1 to M do read (D[I,J]);
    Writeln;
  End;
Function Kolp (D: Mas; M:integer;J:integer):integer;
  Var i,s:integer;
  Begin
  S:=0;
  For I:= 1 to M do if D[I,J]>0 then s:= s+1;
    Kolp:=s;
  End;
begin
  Vvod (A,n);
  For L:=1 to n do
    Begin
    Kp:= Kolp (A,N,L);
    Writeln ('количество элементов >0 в строке ',L,' = ',Kp);
  end;
end.

```

Контрольные вопросы:

1. Для чего нужны в программе процедуры и функции?
2. В чем отличие между процедурой и функцией?
3. Чем отличаются формальные и фактические параметры?
4. Чем отличаются параметры-значения и параметры-переменные?
5. Как объявляются глобальные и локальные переменные? Каково правило видимости этих переменных?
6. Почему при обращении к процедуре, аргумент, передаваемый параметру-переменной, может быть только переменной, а не константой или выражением?

ЗАДАЧИ

1. Написать функцию, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота цилиндра.
2. Написать функцию, которая возвращает максимальное из двух целых чисел, полученных в качестве аргумента.
3. Написать функцию, которая сравнивает два целых числа и возвращает результат сравнения в виде одного из знаков: $>$, $<$ или $=$.
4. Написать функцию `Dohod`, которая вычисляет доход по вкладу. Исходными данными для функции являются: величина вклада, процентная ставка (ставка годовых) и срок вклада (количество дней).
5. Написать функцию `LTrim`, которая удаляет начальные пробелы из строки, полученной в качестве аргумента.
6. Написать функцию вычисления факториала (факториалом целого числа N называется число, равное произведению целых чисел от 1 до N).
7. Написать функцию для определения количества корней квадратного уравнения. Значение функции равно 2, если имеются два разных корня, 1 – если корни одинаковые, 0 – уравнение не имеет решения.
8. Даны отрезки a, b, c и d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь данного треугольника. Определить процедуру `Plo(x, y, z)`, печатающую площадь треугольника со сторонами x, y и z , если такой треугольник существует.

9. Дано натуральное число n . Выяснить, является ли оно полным квадратом. Определить функцию, позволяющую распознавать полные квадраты.

10. Дано четное число $n > 2$. Проверить для этого числа гипотезу Гольдбаха. Эта гипотеза (по сегодняшний день не опровергнутая и полностью не доказанная) заключается в том, что каждое четное n , большее двух, представляется в виде суммы двух простых чисел. Воспользоваться функцией распознавания простых чисел.

11. Написать процедуру, которая выводит на экран строку, состоящую из звездочек. Длина строки (количество звездочек) является параметром функции.

12. Написать процедуру, которая выводит строку, состоящую из одинаковых символов. Длина строки и символ являются параметрами процедуры.

13. Написать процедуру, которая вычисляет объем и площадь поверхности параллелепипеда.

14. На товар дважды была сделана скидка – на p_1 , а затем на p_2 процентов. Первоначальная стоимость товара составляла S тенге. Используя функцию вычисления стоимости товара с учетом скидки на P процентов, найти стоимость товара после двойной скидки.

15. Тариф предусматривает оплату телефонных разговоров следующим образом: при продолжительности разговора меньше P минут стоимость одной минуты составляет S_1 копеек, в противном случае – S_2 тенге/мин (S_1 , S_2 , P - константы). Используя функцию вычисления стоимости одного разговора, найти суммарную стоимость трех звонков известной продолжительности.

16. Вычислить значения функции $f(x)=2\cos(x)+3$ (количество аргументов x и их значения вводятся дополнительно). Вывести результаты в два столбца: в первом - значения x , во втором - значения $f(x)$. Вычисления провести двумя способами: с помощью функции и процедуры.

17. Для чисел a , b , c найти значение выражения $\min(a,b)+\min(a,c)+1$ с использованием функции вычисления минимального из двух чисел.

18. Дан квадрат со стороной a , диагональ этого квадрата является стороной второго квадрата, диагональ второго квадрата – стороной третьего. Найти длину стороны третьего квадрата, используя функцию вычисления длины диагонали квадрата по его стороне.

19. Даны отрезки a, b, c и d . Для каждой тройки этих отрезков, из которых можно построить треугольник, найти площадь данного треугольника. Определить процедуру нахождения площади треугольника, если такой треугольник существует и известны длины его сторон.

20. Даны три числа. Написать функцию, проверяющую, можно ли построить треугольник с такими сторонами.

21. Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Найти периметр десятиугольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. При выполнении задания написать процедуру вычисления расстояния между двумя точками, заданными своими координатами.

22. Написать функцию $\text{Stepen}(x, n)$, значение которой зависит от вещественного x и целого n , вычисляющую (посредством умножения) величину x^n . Применить функцию для вычисления значений функции для различных аргументов, определяемых дополнительно.

23. Написать процедуру сортировки элементов одномерного массива по убыванию элементов. Выполнить процедуру на различных наборах данных.

24. Написать процедуру замены отрицательных элементов матрицы на их квадраты. Выполнить процедуру на различных наборах данных.

25. Написать функцию нахождения следа квадратной матрицы (суммы элементов, лежащих на главной диагонали). Из двух введённых матриц выдать на экран ту, у которой след меньше.

8. ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛАМИ

Цель работы: познакомиться с понятием файлового типа данных (типизированные, текстовые и нетипизированные файлы); выработать навыки работы с файловым типом данных в языке программирования Pascal;

научиться считывать информацию из файлов, записывать информацию в файл; научиться решать задачи с использованием файлов.

Методические рекомендации:

Файл представляет собой структурированный тип данных, содержащий последовательность компонентов одного типа и одной длины. Число элементов в файле (длина файла) не фиксировано. Это является основным отличием файла от массива.

Файл можно представить как ленту, у которой есть начало, а конец не фиксирован. Элементы файла записываются на эту ленту последовательно, друг за другом с помощью некоторого устройства - указателя файла. При чтении или записи этот указатель перемещается к следующему элементу и делает его доступным для обработки. В каждый момент доступен для чтения или записи только тот элемент файла, на который установлен указатель.

При организации работы с файлами необходимо учитывать, что:

- в программе, которая выводит результаты в файл или читает исходные данные из файла, должна быть объявлена файловая переменная типа `text`;
- для доступа к конкретному файлу файловую переменную нужно связать с этим файлом (делается это при помощи инструкции `assign`);
- для того, чтобы файл был доступен, его надо открыть (для чтения с помощью инструкции `reset`, для записи — `rewrite`, для добавления — `append`);
- при работе с файлами возможны ошибки (например, из-за того, что программа пытается открыть файл, которого нет), поэтому после каждой инструкции, которая может привести к возникновению ошибки, желательно, используя функцию `IOResult`, проверять код завершения операции с файлом: чтобы программа могла контролировать результат выполнения операции с файлом, в ее текст надо поместить директиву `{SI-}`;
- запись в файл выполняют инструкции `write` и `writeln`, чтение — `read` и `readln`, причем в качестве первого параметра этих инструкций следует указывать файловую переменную;
- по завершении работы с файлом его нужно обязательно закрыть инструкцией `close`;
- файл, созданный программой, в которой тип файловой переменной объявлен как `text`, можно просмотреть при помощи редактора текста.

Пример

Переписать из текстового файла `f` в файл `g` строки в «перевернутом» виде, причём порядок строк должен быть обратным.

Для решения данной задачи необходимо считывать строки исходного файла, «переворачивая» их и помещая в массив строк (полагая, что

количество записей в файле не более 100). После ввода всех записей создадим новый файл, помещая в него записи из массива в обратном порядке.

Псевдопрограмма реализации данного задания может быть следующей:

Начало

Цикл пока (не конец файла)

Считывание текущей записи файла

«Перевоорачивание» записи

Формирование очередного элемента массива записей

Конец цикла

Формирование нового файла из элементов массива записей

Конец

Программа на языке Pascal может быть следующей:

```
program PR9;
  var f,g:text;
      m:array[1..100] of string;
      k,i:integer;
  begin
    assign(f,'input.dat');
    reset(f);
    k:=0;
    while not eof(f) do
      begin
        readln(f,st);
        k:=k+1;
        st1:= "";
        for i:= 1 to length(st) do st1:= st[i]+st1;
        m[k]:=st1;
      end;
    assign(g,'output.dat');
    rewrite(g);
    for i:= 1 to k do writeln(g,m[i]);
  end.
```

Контрольные вопросы:

1. Что такое файл? Какие существуют виды файлов?
2. Какими стандартными процедурами и функциями располагает Pascal для работы с файлами?
3. Каково должно быть содержание программы по созданию файла и его корректировки (замена элементов, добавление элементов, удаление элементов)?

4. Каковы особенности работы с текстовыми файлами?
5. Каковы особенности работы с типизированными файлами?
6. Как подсчитать число строк в текстовом файле?

ЗАДАЧИ

1. Написать программу, которая позволяет просматривать текстовые файлы (выводит на экран содержимое файла), например, файлы исходных программ Turbo Pascal. Имя просматриваемого файла должно вводиться во время работы программы.

2. Написать программу, которая выводит на экран содержимое файла C:\numbers.txt.

3. Написать программу, которая вычисляет среднее арифметическое чисел, находящихся в файле C:\numbers.txt.

4. Написать программу, которая дописывает в файл phone.txt, находящийся на диске C, фамилию и номер телефона сотрудников фирмы. Если файла на диске нет, то программа должна создать его. В файле каждый элемент данных (имя, фамилия, телефон) должен находиться в отдельной строке.

5. Написать программу, используя которую можно за один сеанс работы добавить информацию о нескольких людях в файл C:\phone.txt.

6. Написать программу, которая позволяет найти нужные сведения в телефонном справочнике (C:\phone.txt). Программа должна запрашивать фамилию человека и выводить его телефон. Если в справочнике есть одинаковые фамилии, то программа должна вывести список всех людей, имеющих эти фамилии.

7. Даны текстовые файлы f1 и f2. Переписать с сохранением порядка следования компоненты файла f1 в файл f2, а компоненты файла f2 в файл f1. Использовать вспомогательный файл h.

8. Дан текстовый файл f. Записать в файл g компоненты файла f в обратном порядке.

9. Даны текстовые файлы f и g . Записать в файл h сначала компоненты файла f , затем - компоненты файла g с сохранением порядка.

10. Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла f , являющиеся четными числами.

11. Дан файл f , компоненты которого являются целыми числами. Получить файл g , образованный из файла f исключением повторных вхождений одного и того же числа.

12. Дан файл f , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Используя вспомогательный файл h , переписать компоненты файла f в файл g так, чтобы в файле g вначале шли положительные, затем - отрицательные числа.

13. Написать программу, переписывающую содержимое текстового файла $t1$ в текстовый файл $t2$, но без пустых строк.

14. Дан файл f , компоненты которого являются целыми числами. Записать в файл g наибольшее значение первых пяти компонент файла f , затем - следующих пяти компонент и т.д. Если в последней группе окажется менее пяти компонент, то последняя компонента файла g должна быть равна наибольшей из компонент файла f , образующих последнюю (неполную) группу.

15. Дан символьный файл f . Подсчитать число вхождений в файл сочетаний 'ab'.

16. Дан символьный файл f . Определить, входит ли в файл сочетание 'abcdefgh'.

17. Дан символьный файл f . Определить число вхождений в файл каждой из букв: 'a', 'b', 'c'.

18. Дан символьный файл f . Группы символов, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Удалить из файла все однобуквенные слова и лишние пробелы. Результат записать в файл g .

19. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *f*, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле - целого типа (количество вещей) и одно - действительное (вес в килограммах). Найти число пассажиров, имеющих более двух вещей, и число пассажиров, количество вещей которых превосходит среднее число вещей.

20. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Дан файл *f*, содержащий сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле - целого типа (количество вещей) и одно - действительное (вес в килограммах). Выяснить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей, и по весу.

21. Создать файл, содержащий сведения о студентах (фамилия, имя, отчество, код группы, год поступления). Определить, имеются ли студенты-однофамильцы.

22. Создать файл *f*, содержащий сведения об автомобиле (фамилия автовладельца, марка машины, год выпуска и государственный номер автомобиля). Определить фамилии владельцев и номера автомобилей определенной марки.

23. Дан файл *f*, содержащий различные даты. Каждая дата – это число, месяц и год. Найти: а) год с наименьшим номером; б) все весенние даты; в) самую позднюю дату.

24. Создать файл *f*, содержащий сведения о книгах (фамилия автора, название и год издания). Найти названия книг данного автора, изданных после 2010г.

25. Создать файл *f*, содержащий сведения о кубиках: размер каждого кубика (длина ребра в сантиметрах), его цвет (красный, зеленый, желтый или синий) и материал (деревянный, металлический, картонный). Найти количество кубиков каждого из перечисленных цветов и их суммарный объем.

III. ЗАДАНИЯ ПОВЫШЕННОЙ СЛОЖНОСТИ

1. Две точки заданы на плоскости своими координатами, которые могут быть как декартовыми, так и полярными. Требуется вычислить расстояние между этими двумя точками.

2. Даны действительные числа a , b , c , x , y . Выяснить, пройдет ли кирпич с ребрами a, b, c в прямоугольное отверстие со сторонами x и y . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из его ребер было параллельно или перпендикулярно каждой из сторон отверстия.

3. Сможет ли шар радиуса R пройти в ромбообразное отверстие со стороной P и острым углом Q ?

4. Написать программу, которая печатает True или False в зависимости от того, выполняются или нет заданные условия:

- квадрат заданного трехзначного числа равен кубу суммы цифр этого числа;
- сумма двух первых цифр заданного четырехзначного числа равна сумме двух его последних цифр;
- среди цифр заданного трехзначного числа есть одинаковые;
- среди первых трех цифр из дробной части заданного положительного вещественного числа есть цифра 0.

5. Проверить, можно ли из четырех данных отрезков составить параллелограмм. Написать программу, определяющую попадает ли точка с координатами (x, y) в заштрихованную область.

6. На промежутке от 1 до M найти все числа Армстронга. (Натуральное число из n цифр называется числом Армстронга, если сумма его цифр, возведенных в n -ю степень, равна самому числу.)

7. Определить, является ли заданное число совершенным, т.е. равным сумме всех своих (положительных) делителей, кроме самого этого числа (например, число 6 совершенно: $6=1+2+3$).

8. Дано натуральное k . Напечатать k -ю цифру последовательности 1234567891011121314..., в которой выписаны подряд все натуральные числа.

9. Дано натуральное k . Напечатать k -ю цифру последовательности 149162536..., в которой выписаны подряд квадраты всех натуральных чисел.

10. Дано натуральное k . Напечатать k -ю цифру последовательности 1123581321..., в которой выписаны подряд все числа Фибоначчи.

11. Дан двумерный массив. В каждой строке все его элементы, не равные нулю, переписать (сохраняя порядок) в начало строки, а нулевые элементы - в конец массива. Новый массив не заводить.

12. В массиве $A[1..N,1..N]$ определить номера строки и столбца какой-нибудь седловой точки. Некоторый элемент массива называется седловой точкой, если он является одновременно наименьшим в своей строке и наибольшим в своем столбце.

13. Массив $A[1..5,1..7]$ содержит вещественные числа. Требуется ввести целое число K и вычислить сумму элементов $A[I,J]$, для которых $I+J=K$. Однако, прежде следует убедиться, что значение K позволяет найти решение, в противном случае нужно напечатать сообщение об ошибке.

14. Дан массив $A[1..N,1..N]$. Составить программу, которая прибавила бы каждому элементу данной строки элемент, принадлежащий этой строке и главной диагонали.

15. Дана матрица $N \times M$. Переставляя ее строки и столбцы, переместить наибольший элемент в верхний левый угол. Определить можно ли таким же образом поместить минимальный элемент в нижний правый угол.

16. Элемент двумерного массива называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером $N \times N$, найти максимум среди всех локальных минимумов.

17. Зашифровать введенную с клавиатуры строку, поменяв местами первый символ со вторым, третий с четвертым и т. д. Затем провести дополнительную шифровку результата смещением кода. Провести дешифровку.

18. Составить процедуру создания текстового окна, окаймленного рамкой из псевдографических символов. В параметры процедуры ввести координаты левого верхнего угла, размеры и цвет окна, а также цвет рамки.

19. Написать программу вычисления длин сторон, углов, площади, радиусов вписанной и описанной окружностей для треугольника, заданного координатами вершин на плоскости в декартовых или полярных координатах. Использовать структуру данных типа «запись» с вариантами.

20. Дана дата (число, месяц, год и день недели). Необходимо определить:

- правильность заданной даты;
- последний день данного месяца;
- сколько дней прошло от 1-го января 1-го года нашей эры;
- день недели, приходящийся на заданную дату, если 1-е января 1-го года нашей эры было понедельником.

21. Разработать программу вычисления длин сторон, углов, площади, радиусов вписанной и описанной окружностей для треугольника, заданного координатами вершин: на плоскости в декартовых или полярных координатах.

22. Написать программу, осуществляющую приведение подобных членов многочлена с использованием структуры данных типа «запись».

23. Рассмотрим произвольное натуральное число и найдем сумму его цифр, затем сумму цифр полученного числа и так далее, пока не получим однозначное число. Назовем это число цифровым корнем. Требуется написать программу, которая для заданного N ($N < 10100$) находит его цифровой корень.

24. Требуется написать программу, которая выводит в порядке возрастания все правильные несократимые дроби, знаменатели которых не превосходят N ($2 \leq N \leq 500$).

25. На экране компьютера, работающего в операционной системе Windows, было открыто N ($N \leq 20$) окон, положение каждого из которых однозначно определяется четверкой натуральных чисел - X_1, Y_1, X_2, Y_2 - координатами левого верхнего и правого нижнего угла окна. Очевидно, что окна, открытые позже, могут частично или полностью перекрывать открытые

ранее. Окно считается видимым, если виден хотя бы один образующий его пиксел. Определить количество видимых окон.

IV. ПРИМЕРНЫЕ ТЕСТОВЫЕ ЗАДАНИЯ

1. Инструкции в линейной программе ...

- A. выполняются в определенном порядке, в зависимости от некоторого условия
- B. выполняются строго последовательно друг за другом
- C. не выполняются ни разу
- D. выполняться несколько раз

2. Для запуска программы на выполнение в среде TurboPascal используется комбинация клавиш

- A. {Alt + F5}
- B. {Alt + F9}
- C. {Ctrl + F9}
- D. {Ctrl + F5}

3. Чтобы окно пользователя по завершении программы не исчезало с экрана, в конец программы нужно добавить инструкцию:

- A. writeln
- B. readln
- C. repeat
- D. read

4. Переменные, используемые в программе, объявляются:

- A. в начале программы после слова *Program*
- B. в любом месте программы между *begin* и *end*
- C. в начале программы после слова *var*
- D. в любом месте программы после оператора *for*

5. Какое значение будет иметь переменная *x* после выполнения следующих строк программы:

```
x:=2;  
y:=x+1;  
x:=y*x;  
y:=y+x;  
x:=x - y;
```

- A. -1
- B. -4
- C. -3
- D. 3

6. Выберите правильный вариант вывода на экран значения переменной *z*:

- A. writeln z
- B. writeln (z)
- C. writeln 'z'
- D. writeln ('z')

7. **Оператор выбора** начинается словом ...

- A.case
- B.if
- C.else
- D.until

8. Из следующих утверждений выберите истинное:

- A. переменная, используемая в операторе *For*, может быть типа *real*
- B. в операторе *While* проверка условия осуществляется после тела цикла
- C. перед *Else* всегда ставится символ «;»
- D. оператором *Write* производится вывод данных на экран

9. В результате выполнения следующих инструкций

```
Fori:=1 to 5 do  
  write('*');
```

на экран будут выведены

- A. пять звездочек в одной строке
- B. цифры от 1 до 5 в одной строке
- C. пять звездочек столбиком
- D. цифры от 1 до 5 столбиком

10. После выполнения следующих инструкций

```
n:=0;  
while n<=5 do  
  n:=n+1;
```

значение переменной n будет равно

- A.1
- B.0
- C.6
- D.5

11. Для обозначения числа 3.14 нужно использовать переменную типа **real**

- A.integer
- B.string
- C.char

12. Цикл **ПОКА** имеет формат

- A.if – then – else
- B.for – to – do
- C.while – do
- D.repeat – until

13. Условный оператор применяется для программирования:

- A. сложных алгоритмов
- B. линейных алгоритмов

- С.ветвящихся алгоритмов
- Д.циклических алгоритмов

14. Ветвящийся алгоритм подразумевает:

- А.неоднократное повторение отдельных частей программы
- В.последовательное выполнение всех элементов программы
- С.выполнение лишь нескольких, удовлетворяющих заданному условию частей программы
- Д.безусловный переход в любую часть программы

15. Оператор CASE служит для создания:

- А.линейных алгоритмов
- В.ветвящихся алгоритмов
- С.циклических алгоритмов
- Д.перехода в любую часть программы

16. В результате выполнения фрагмента программы

```
a:=7;  
b:=8;  
k:=5+a;  
if a>b then k:=k*3  
else k:=k-3;  
writeln (k);
```

...

переменная k...примет следующее значение:

- А.7
- В.8
- С.9
- Д.10

17. Алгоритм включает в себя ветвление, если:

- А.он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий
- В.ход его выполнения зависит от истинности тех или иных условий
- С.его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
- Д.его команды не выполняются никогда

18. Определите значение целочисленных переменных a и b после выполнения фрагмента программы(div и mod – операции, вычисляющие результат деления нацело первого аргумента на второй и остаток от деления соответственно):

```
a:= 3 + 8*4;  
b:= (a div 10) + 14;  
a:= (b mod 10) + 2;
```

19. Укажите приоритет операций в языке Паскаль:

- A. Операции типа сложения: +, -, or, xor
- B. Операции отношения: =, <>, <, >, <=, >=, in
- C. Унарная операция not, унарный минус -, взятие адреса @
- D. Операции типа умножения: *, /, div, mod, and, shl, shr

20. Укажите, в каком из условных операторов допущена синтаксическая ошибка:

- A. If B=0 then writeln ('Деление на нуль невозможно')
- B. If a>b then max := a else max:=b
- C. If (a>b) and (b>0) then c:=a+b
- D. If a<b then min:=a; else min:=b

21. Вывод данных на экран осуществляется с помощью оператора:

- A. WRITE, WRITELN
- B. READ, READLN
- C. PROGRAM
- D. PROCEDURE

22. Определите, чему будет равно значение переменной M в результате выполнения серии операторов

N:=5; Y:= 3 mod 2

If Y>0 then

M:=N/5+3*Y*N

Else M:=888;...

23. Тип переменных INTEGER – это:

- A. целочисленный тип
- B. логический тип
- C. натуральный тип
- D. перечисляемый тип

24. Выберите цикл с предусловием:

- A. For
- B. Repeat
- C. Case
- D. While

25. Выберите оператор выбора:

- A. While
- B. For
- C. Repeat
- D. Case

26. Выберите цикл с параметром:

- A. While
- B. For

C.Repeat

D.Case

27. Определите значение переменной S после выполнения операторов:

```
i:=0; S:=0  
While i<5 do  
Begin  
i:=i+2;  
S:=S+i*i  
End;
```

28. Определите, сколько строк напечатает программа:

```
var k,l:integer;  
begin  
FOR k:=1 TO 10 DO  
FOR l:=1 TO 5 DO  
Writeln ('я программирую на языке Pascal');  
end.
```

A.10

B.5

C.50

D.15

29. Определите, что будет выведено на экран в результате выполнения фрагмента программы:

```
s:=0;  
FOR j:=1 TO 3 DO  
s:=s+2*j;  
write (' j=', j, ' s=', s)
```

30. Определите, что будет выведено на экран в результате выполнения фрагмента программы:

```
p:=1;  
FOR k:=1 TO 5 DO  
begin  
p:=p*k;  
end;  
write (' k=', k, ' p=', p)
```

31. Укажите, какой из операторов цикла имеет два вида записи:

A.For

B.While

C.Repeat

D.Case

32. Операторными скобками называются следующие конструкции:

- A. ()
- B. <>
- C. begin ... end
- D. { }

33. Определите, в каком из операторов допущена ошибка:

- A. For i:=1 to 20 do p:=p+1
- B. While s<3 do s:=s-3
- C. Repeat k<1 until k:=k+1
- D. For I:=10 downto 5 do p:=p+1

34. Тело цикла While для вычисления значений функций $f(x)=x^4$, на отрезке [-50;50] с шагом 10 будет иметь вид:

- A. f:= x+x+x+x; x:=x+10;writeln (f,x)
- B. f:=x⁴; x:=x+10;writeln (f,x)
- C. f:= x*x*x*x; x:=x+0.10;writeln (f,x)
- D. f:= x*x*x*x; x:=x+10; writeln (f,x)

35. Оператором цикла не является оператор

- A. While
- B. For
- C. Readln
- D. Repeat

36. Оператор присваивания имеет вид:

- A. =
- B. :=
- C. =:
- D. ==

37. Команда CLRSCR служит для

- A. ожидания нажатия клавиши
- B. ввода данных с экрана
- C. очистки экрана
- D. вывода данных на экран

38. Раздел VAR служит для описания

- A. используемых переменных
- B. величин
- C. выражений
- D. типов

39. Правильная строковая запись выражения

$x - \frac{x^3}{|x|} + \sin x$ имеет следующий вид:

- A. x-x***3/abs x + sin (x)

- B. $x - x * x * x / \text{abs}(x) + \sin(x)$
- C. $x - (x * x * x) / \text{abs}(x) + \sin(x)$
- D. $x - (x ** 3) / \text{abs}(x) + \sin(x)$

40. Целочисленное деление можно выразить следующей функцией:

- A. $A \text{ mod } B$
- B. $A \text{ div } B$
- C. $\text{Abs}(A) * B$
- D. A / B

41. Определите значение переменной у после выполнения операторов

```
y:=1;i:=2;
for i:=1 to 4 do y:=1/i+5; y:=y/2;
```

42. Определите значение переменной у после выполнения операторов

```
y:=20; i:=6;
while i>4 do i:=i-1; y:=y/i;
```

43. Определите значение переменной у после выполнения операторов

```
y:=0; i:=1;
repeat y:=y+i; i:=i+2;
until i<5;
```

44. Определите, что будет выведено на экран в результате выполнения фрагмента программы:

```
s:=0;
for j:=10 to 15 do
begin
    s:=s+2*j;
    write(' j=', j:2, ' s=', s:4)
end;
```

45. Определите, что будет выведено на экран в результате выполнения фрагмента программы:

```
p:=1;
for k:=5 downto 1 do
begin
    p:=p*k;
end;
write(' k=', k:2, ' p=', p:3)
```

46. Служебное слово VAR в программе на языке Pascal фиксирует начало раздела программы, содержащего:

- A. список меток

- В. описание сложных типов данных
- С. перечень констант
- Д. описание переменных

47. В алфавит языка Pascal не входит служебное слово:

- A. THEN
- B. END
- C. STEP
- D. IF

48. Числа в языке Pascal различаются:

- A. на натуральные и целые
- B. на целые и вещественные
- C. на натуральные и вещественные
- D. на целые и рациональные

49. Формальные параметры процедуры:

- A. описываются в ее заголовке
- B. перечисляются при вызове процедуры
- C. указываются при описании данных в программе
- D. присваиваются в процессе выполнения программы

50. В системе Turbo Pascal начало раздела программы, содержащего список меток, фиксируется служебным словом:

- A. CONST
- B. TYPE
- C. VAR
- D. LABEL

51. Оператор организации ввода данных с клавиатуры в системе программирования Turbo Pascal записывается с использованием служебного слова:

- A. WRITE
- B. INPUT
- C. READ
- D. RESET

52. Комментарий к тексту программы на языке Pascal заключается:

- A. в фигурные скобки
- B. в круглые скобки
- C. в квадратные скобки
- D. в апострофы

53. Служебное слово CONST в программе на языке Pascal фиксирует начало раздела программы, содержащего:

- A. перечень констант
- B. список меток

- C. описание переменных
- D. описание сложных типов данных

54. В качестве имени в языке Pascal нельзя использовать:

- A. BR
- B. OR
- C. WR
- D. AR

55. При записи оператора вывода на языке Pascal используется служебное слово:

- A. WRITE
- B. READ
- C. BEGIN
- D. WHILE

56. Операторы в программе на языке Pascal отделяются друг от друга:

- A. двоеточием
- B. пробелом
- C. запятой
- D. точкой с запятой

57. Логическое выражение $(X \geq A) \text{ AND } (X \leq B)$ истинно при:

- A. x принадлежит отрезку $[A, B]$
- B. x принадлежит объединению интервалов $[A, \infty]$ и $[B, \infty]$
- C. x принадлежит объединению интервалов $[-\infty, A]$ и $[B, +\infty]$
- D. x принадлежит объединению интервалов $[A, +\infty]$ и $[-\infty, B]$

58. Фактические параметры процедуры:

- A. описываются в ее заголовке
- B. указываются при описании данных в программе
- C. перечисляются при ее вызове
- D. указываются при описании промежуточных переменных процедуры

59. В программе вычисления суммы элементов арифметической прогрессии (при известном ее первом члене, разности и количестве ее членов):

```
Program Prim1;  
  Var a, d, s: real;  
      n: integer;  
  Begin  
    readln(a, d, n);  
    s:= ;  
    writeln(s);  
  End.
```

в операторе присваивания не указано арифметическое выражение. Оно может быть записано как:

A. $a*n/2+d*(n-1)*n/2$

B. $a*(n+d*(n-1))*n/2$;

C. $a+d*(n-1)*n/2$

D. $a*n+d*(n-1)*n/2$

60. Дана программа:

```
Program Prim2;
```

```
Var X: Integer;
```

```
Begin
```

```
Readln(X);
```

```
If X MOD 2=0 Then Writeln('ДА')
```

```
Else Writeln('НЕТ');
```

```
End.
```

При каком значении X будет получен ответ 'ДА'?

A.7

B.4

C.15

D.21

V. ОСНОВНЫЕ ВОПРОСЫ ПО ИЗУЧАЕМОЙ ДИСЦИПЛИНЕ

1. Основные стадии компьютерного решения прикладной задачи.
2. Общее понятие алгоритма.
3. Свойства компьютерных алгоритмов.
4. Способы представления алгоритмов.
5. Понятие блок-схемы.
6. Основные алгоритмические конструкции: следование, цикл, ветвление.
7. Стандартные алгоритмы.
8. Язык программирования как средство представления алгоритма.
9. Классификация языков программирования.
10. Подходы к выбору языка программирования.
11. Содержание и порядок выполнения отдельных этапов создания программы.
12. Компоненты и возможности системы программирования.
13. Модули системы программирования.
14. Понятие типов данных в языке программирования.
15. Основные компоненты языка программирования Pascal.
16. Структура программы на языке Pascal.
17. Назначение и оформление раздела описания данных.
18. Описание одномерных массивов.
19. Описание двумерных массивов.
20. Размещение одномерных массивов в памяти.
21. Размещение двумерных массивов в памяти.
22. Назначение процедур.
23. Правила описания процедур-подпрограмм.
24. Правила обращения к процедурам-подпрограммам.
25. Правила описания процедур-функций.
26. Правила обращения к процедурам-функциям.
27. Присваивание значений переменным.
28. Структура оператора присваивания.
29. Понятие совместимости типов.
30. Операторы перехода и ветвления.
31. Структура и синтаксис полного условного оператора.
32. Структура и синтаксис неполного условного оператора.
33. Понятие заголовка и тела цикла.
34. Синтаксис оператора цикла While.
35. Синтаксис оператора цикла Repeat.
36. Синтаксис оператора цикла Do.
37. Синтаксис вложенных циклов.

38. Организация работы с одномерными массивами.
39. Организация работы с двумерными массивами.
40. Организация работы с многомерными массивами.
41. Варианты оператора ввода информации на языке Pascal.
42. Варианты оператора вывода информации на языке Pascal.
43. Описание строковых переменных.
44. Операции над строковыми переменными.
45. Понятие файла. Виды файлов.
46. Стандартные процедуры языка Pascal для организации работы с файлами.
47. Структура программы по созданию файла.
48. Структура программы по корректировке файла.
49. Особенности организации работы с текстовыми файлами.
50. Особенности организации работы с типизированными файлами.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основная литература

1. Информатика: учебник: В 2-х ч, Ч. 2/ Ред. С.Н. Назаров. - М.: ИНТУИТ: БИНОМ. Лаборатория знаний, 2012. - 527 с.
2. Информатика и программирование: Компьютерный практикум: учеб. пособие/ А.Н. Гуда; Ред. В.И. Колесников. - М.: Дашков и К, 2009. - 236с.
3. Информатика и программирование: Компьютерный практикум: учеб. пособие/ А.Н. Гуда; Ред. В.И. Колесников. - М.: Дашков и К, 2009. - 236с.
4. Немнюгин С.А. Turbo Pascal. Программирование на языке высокого уровня : учебник/ С.А. Немнюгин. - СПб.: Питер, 2008. - 543 с.
5. Фаронов В.В. Turbo Pascal: учеб. пособие/ В.В. Фаронов. - СПб.: Питер, 2007.- 366 с.
6. Фаронов В.В. Turbo Pascal 7.0. Учебный курс: учеб. пособие/ В.В. Фаронов. - М.: КНОРУС, 2013. - 363 с.

Дополнительная литература

1. Васильев П.П. Турбо Паскаль в примерах и задачах. Освой самостоятельно: Учеб. пособие/ П.П. Васильев. - М.: Финансы и статистика, 2002. - 495 с.
2. Глушаков С.В. Turbo Pascal 7.0/ С.В. Глушаков, В.Н. Зорянский, С.Н. Хоменко. - Харьков: Фолио, 2002. - 501 с.
3. Голицына О.Л. Основы алгоритмизации и программирования: учеб. пособие/ О.Л. Голицына, И.И. Попов. - М.: Форум: ИНФРА-М, 2005. - 429 с.
4. Голицына О.Л. Языки программирования: учеб. пособие/ О.Л. Голицына, Т.Л. Партыка, И.И. Попов. - М.: Форум: ИНФРА-М, 2008. - 397 с.
5. Даулеткулов А.Б. Основы программирования на языке «Паскаль» (Алгоритмизация и программирование): Учеб.-метод. пособие/ А.Б. Даулеткулов. - Алматы: ИНТ, 2004. - 166 с.
6. Меняев М.Ф. Информатика и основы программирования: учеб. пособие/ М.Ф. Меняев. - М.: Омега-Л, 2005. - 463 с.
7. Милов А.В. Основы программирования в задачах и примерах/ А.В. Милов. - М.: АСТ; Харьков: Фолио, 2003. - 397 с.
8. Моргун А.Н. Решение задач средствами языка Turbo Pascal 7.0: Учеб. пособие/ А.Н. Моргун. - Киев: Юниор, 2002. - 213 с.
9. Немнюгин С.А. Turbo Pascal: Практикум/ С.А. Немнюгин. - СПб.: Питер, 2003. - 267 с.

10. Немнюгин С.А. Turbo Pascal: Учеб. пособие для вузов/ С.А. Немнюгин. - СПб.: Питер, 2003.- 491 с.

11. Немнюгин С.А. Изучаем Turbo Pascal/ С. Немнюгин, Л. Перколаб. - СПб.: Питер, 2003. - 309 с.

12. Павловская Т.А. Паскаль. Программирование на языке высокого уровня: Учеб. для вузов/ Т.А. Павловская. - СПб.: Питер, 2004. - 392 с.

13. Рудаков П.И. Основы языка Pascal/ П.И. Рудаков, М.А. Федотов. - М.: Горячая линия - Телеком, 2002. - 260 с.

Для заметок

Для заметок

Подписано в печать 14.06.2013 г. Тираж 50 экз.

Формат изд. 60x84/16. Объем 4,25 усл. печ. л.

Отпечатано в типографии “ИП Волков А.И.”

Райымбека 212/1, оф. 319. Тел.: 330-03-12, 330-03-13