

Е. В. Савина

**ПРАКТИКУМ
ПО ПРОГРАММИРОВАНИЮ
на PascalABC.NET**

DirectMEDIA

Е. В. Савина

**ПРАКТИКУМ
ПО ПРОГРАММИРОВАНИЮ
на PascalABC.NET**



**Москва
Берлин
2021**

УДК 004.432(075)
ББК 32.973.22я723
С13

Рецензенты

Кобзаренко Дмитрий Николаевич, доктор технических наук,
ведущий научный сотрудник Института проблем геотермии и возобновляемых
энергоресурсов — Филиала Объединенного института высоких температур РАН;
Мустафаев Арслан Гасанович, доктор технических наук, доцент кафедры
информационных технологий и информационной безопасности
Дагестанского государственного университета народного хозяйства

Савина, Е. В.

С13 Практикум по программированию на PascalABC.NET / Е. В. Савина. —
Москва ; Берлин : Директ-Медиа, 2021. — 122 с.

ISBN 978-5-4499-1915-1

PascalABC.NET — современная версия языка программирования Паскаль, предназначенного для обучения программированию. Настоящий практикум включает краткую справку по языку и собственно задания и упражнения для лабораторных и практических занятий по таким дисциплинам, как «Информатика и программирование», «Языки программирования» и ряду других. Задания разделены по темам, и в каждом разделе приведены решения наиболее типичных задач.

В силу простоты и доступности, язык PascalABC.NET широко используется на начальных этапах обучения программированию. Настоящий практикум предназначен для обучения в системе СПО, он также может быть рекомендован преподавателям школ и вузов.

УДК 004.432(075)
ББК 32.973.22я723

ISBN 978-5-4499-1915-1

© Савина Е. В., текст, 2021
© Издательство «Директ-Медиа», оформление, 2021

ОГЛАВЛЕНИЕ

ОБ АВТОРЕ	4
Введение	5
I. Основные сведения о языке PascalABC.NET	6
II. Задачи и упражнения	13
1. Алфавит языка PascalABC.NET. Оператор присваивания. Ввод/вывод данных	13
2. Простые типы данных. Совместимость типов. Преобразование типов	17
3. Реализация линейных и разветвляющихся алгоритмов	23
4. Программирование циклических алгоритмов	28
5. Массивы	36
6. Записи, множества	44
7. Обработка символов и строк	49
8. Файлы	52
9. Процедуры и функции	57
10. Графика	60
III. Комплексные задания по разным темам	63
Ответы и комментарии	74
IV. Тестовые задания по темам	80
Ответы к тестовым заданиям	118
Список рекомендуемой литературы	121

ОБ АВТОРЕ



Савина Елена Владимировна, кандидат физико-математических наук, доцент кафедры «Информационные технологии и информационная безопасность» Дагестанского государственного университета народного хозяйства (г. Махачкала).

Введение

Язык Паскаль, созданный Н. Виртом в начале 70-х годов прошлого века для обучения студентов программированию, до сих пор не утратил своей актуальности. Его современная версия — PascalABC для платформы NET (в настоящем Практикуме используется версия 3.6.3 от 09.06.2020) предоставляет пользователю мощные возможности для создания программ, поддерживая императивный, объектно-ориентированный и функциональный стили программирования. При этом язык прост в изучении и использовании и наилучшим образом подходит для выполнения своей основной задачи — обучения принципам программирования, методам реализации базовых алгоритмов.

Структурно Практикум состоит из четырех частей. В первой части приведена краткая справочная информация о синтаксисе языка, во второй — задания для выполнения лабораторных и практических работ, разделенные по темам. В начале каждого раздела второй части приведены примеры решения наиболее типичных задач с комментариями.

В третьей части содержатся комплексные задания с элементами теста, для выполнения которых необходимы знания из различных разделов курса. Задания третьей части предназначены для выполнения без компьютера, они направлены на развитие навыка чтения программного кода, что является чрезвычайно важным в обучении программированию. К заданиям третьей части приведены ответы и необходимые пояснения.

Четвертая часть содержит порядка 200 тестовых заданий по разным разделам курса программирования на PascalABC.NET с ответами. Их удобно использовать для формирования тестов разного назначения (текущая проверка знаний, промежуточный и итоговый контроль, самоконтроль).

Для эффективной работы с Практикумом необходимо изучение литературы с подробным описанием языка PascalABC.NET. Список рекомендуемой литературы и Интернет-источников приведен в конце издания. Кроме того, нельзя не отметить, что PascalABC.NET располагает превосходной справочной системой, включающей, помимо, собственно, справочника по языку, несколько обучающих блоков (задачник, модули «Робот» и «Чертежник» и т. д.).

Практикум может быть использован для работы на практических и лабораторных занятиях, составления заданий для контрольных работ, карточек, самостоятельной подготовки студентов.

I. Основные сведения о языке PascalABC.NET

1. Алфавит языка PascalABC.NET (далее — PasABC) включает:

1) буквы латинского алфавита от a до z и от A до Z и знак подчеркивания;

2) цифры — арабские от 0 до 9;

3) специальные символы

+ — * / = , . ' : ; < > [] () { } ^ @ \$ #,

пары символов

<> <= >= := += -= *= /= ** (***) (.) и знак пробела;

4) зарезервированные слова и директивы.

Зарезервированные слова используются для описания операторов, данных, модулей, процедур и функций и т. д.

2. Идентификаторы в PascalABC.NET — имена переменных, констант, типов, меток, подпрограмм (процедур и функций), программ, модулей, объектов и полей в записях. Идентификатор должен начинаться с буквы или знака подчеркивания и включать только буквы, цифры и знак подчеркивания.

PascalABC.NET нечувствителен к регистру (прописные и строчные буквы эквивалентны).

3. В качестве констант могут использоваться целые и вещественные числа, логические константы (*false* или *true*), символы, строки символов, конструкторы множеств и неопределенный указатель *nil*.

4. Выражение представляет собой комбинацию переменных, констант и обращений к функциям (операндов), соединенных знаками операций.

5. В PascalABC.NET определены следующие операции:

унарные not, @;

бинарные:

1) мультипликативные *, /, div, mod, and;

2) аддитивные +, -, or, xor;

3) операции отношения =, <>, <, >, <=, >=, in.

Операции приведены в порядке убывания приоритета. Порядок выполнения операций можно изменять с помощью круглых скобок.

6. Структура программы в PascalABC.NET имеет следующий вид:

```
program <имя программы>; // строка заголовка
```

```
{ Раздел описаний }
```

```
begin
```

```
{ Раздел исполняемых операторов }
```

```
end.
```

Строка заголовка является необязательной. В разделе описаний должны быть описаны все константы, типы данных (если они определяются пользователем) и метки, используемые в программе. Раздел описания типов начинается с зарезервированного слова *type*, констант — *const*, меток — *label*. Разделы описания типов, констант, меток могут следовать в любом порядке и встречаться любое количество раз.

В разделе описаний также могут быть описаны глобальные переменные, которые должны быть доступны во всех частях программы. Однако описание переменной, которое начинается с зарезервированного слова *var*, в Pas-

calABC.NET рекомендуется располагать в разделе операторов непосредственно перед первым использованием этой переменной. PascalABC.NET допускает также определение типа переменной по инициализации.

Примеры:

```
var a: real;
```

```
var a := 1; // переменной a будет присвоен тип integer.
```

7. В таблице 1 приведены некоторые стандартные функции, определенные в PascalABC.NET. Заметим, что в предыдущих версиях языка Паскаль не было функции возведения в степень. В PascalABC.NET для этого предназначена операция, обозначаемая двойным символом **, а также функция двух переменных $\text{power}(x, y)$. Кроме того, в PascalABC.NET определена функция $\text{swap}(x, y)$, которая меняет местами значения переменных x и y .

Таблица 1

Функция (процедура)	Тип аргумента	Тип результата	Действие
$\text{abs}(x)$	вещественный, целый	совпадает с x	модуль x
$\text{ArcTan}(x)$	- * -	real	арктангенс (в радианах)
$\text{ArcSin}(x)$	- * -	real	
$\text{ArcCos}(x)$	- * -	real	
$\text{cos}(x)$	- * -	real	косинус, аргумент в радианах
$\text{sin}(x)$	- * -	real	синус, аргумент в радианах
$\text{tan}(x)$	- * -	real	
$\text{exp}(x)$	- * -	real	экспонента (e^x)
$\text{ln}(x)$	- * -	real	натуральный логарифм
$\text{sqr}(x)$	- * -	real	квадрат аргумента
$\text{sqrt}(x)$	- * -	real	квадратный корень
$\text{int}(x)$	вещественный	real	целая часть числа
$\text{frac}(x)$	вещественный	real	дробная часть числа
$\text{round}(x)$	вещественный	integer	ближайшее целое
$\text{trunc}(x)$	вещественный	integer	отсечение дробной части
$\text{dec}(x [,i])$	целый	совпадает с x	уменьшает значение x на i , если i отсутствует, то на 1
$\text{inc}(x [,i])$	целый	совпадает с x	увеличивает значение x на i , если i отсутствует, то на 1
$\text{Hi}(i)$		byte	возвращает старший байт аргумента
$\text{Lo}(i)$		byte	возвращает младший байт аргумента
$\text{odd}(x)$	целый	boolean	true, если аргумент — нечетное число, иначе false
$\text{random}(x)$	word	word	возвращает псевдослучайное число из диапазона $0..(x - 1)$

Продолжение табл. 1			
<i>Функция (процедура)</i>	<i>Тип аргумента</i>	<i>Тип результата</i>	<i>Действие</i>
chr(x)	byte	char	по коду символа x возвращает символ
upcase(x)	char	char	если x — строчная латинская буква, возвращает прописную, иначе — символ x
high(x)	диапазон	базовый порядковый	максимальное значение типа-диапазона
low(x)	диапазон	базовый порядковый	минимальное значение типа-диапазона
pred(x)	порядковый	совпадает с x	предыдущее значение x
succ(x)	порядковый	совпадает с x	следующее значение x
ord(x)	порядковый	целый	если x — целое, возвращает x, если символ — его код

9. Операторы

Операторы в языке PascalABC.NET делятся на простые и структурированные.

Простые операторы не содержат в себе других операторов. К простым операторам относятся:

- ✓ оператор безусловного перехода;
- ✓ оператор присваивания;
- ✓ оператор обращения к процедуре.

1) Оператор безусловного перехода goto<метка> позволяет изменять порядок выполнения операторов, использовать не рекомендуется.

2) Оператор присваивания

<имя_переменной> := <выражение>;

*a := 3.14; b := sqrt(a*a - 1); d := a/(a + b).*

PascalABC.NET допускает использование сокращенных операторов присваивания

*+= -= *= /=*

Запись *a += 2* эквивалентна *a := a + 2*, запись *b *= 3* эквивалентна *b := b*3* и т. д.

3) Процедуры ввода данных:

read(x1, x2, ..., xn);

readln(x1, x2, ..., xn);

x1, x2, ..., xn — список вводимых переменных, процедура readln после считывания последнего значения переводит курсор на новую строку. Процедура readln без параметров останавливает выполнение программы до нажатия клавиши Enter.

Кроме этого, в PascalABC.NET есть обширный набор функций ввода, которые позволяют одним оператором выполнять описание типа переменной и присваивание ей значения, вводимого с клавиатуры, например:

var a := ReadInteger;

a также множественный ввод данных, например:

var (a, b, c) := ReadReal3;

Функции ввода данных в PascalABC.NET: ReadReal, ReadInteger, ReadChar, ReadBoolean, ReadString, а также ReadReal2, ReadReal3, ..., ReadInteger2, ReadInteger3, ..., ReadString2 и т. д.

4) Процедуры вывода данных:

```
write(x1, x2, ..., xn);  
writeln(x1, x2, ..., xn);  
print(x1, x2, ..., xn);  
println(x1, x2, ..., xn);
```

x1, x2, ..., xn — список выводимых переменных, констант, выражений, процедуры writeln, println после вывода информации переводят курсор на новую строку. Процедуры writeln, println без параметров переводят курсор в начало новой строки.

Отличие: процедуры writeln, write позволяют выполнить форматный вывод данных, т. е. задать ширину поля вывода, количество десятичных знаков после запятой; процедуры print, println автоматически добавляют пробелы между выводимыми данными.

К структурированным операторам относятся:

- ✓ составной оператор;
- ✓ операторы ветвления — условный оператор и оператор выбора;
- ✓ 5 операторов цикла: оператор цикла с параметром for, операторы цикла loop и foreach и 2 оператора цикла с условием (с предусловием и с постусловием).

5) Составной оператор — группа операторов, заключенных в операторные скобки begin ... end, которые рассматриваются как один оператор:

```
begin  
  a1 *= 4; // иначе: a1 := 4*a1;  
  a2 += a1 // иначе: a2 := a1 + a2  
end;
```

6) Условный оператор, полная форма:

```
if <условие> then <оператор_1> else <оператор_2>;
```

неполная форма:

```
if <условие> then <оператор>;
```

условие — выражение логического типа, оператор_1, оператор_2, оператор — любые операторы языка Паскаль (возможно, составные).

7). Оператор выбора:

```
case <ключ_выбора> of  
  <список_1>: <оператор_1>;  
  <список_2>: <оператор_2>;  
  ...  
  <список_n>: <оператор_n>  
else <оператор_n+1>  
end;
```

Ключ_выбора — параметр, по которому происходит выбор варианта продолжения программы, выражение любого порядкового типа, список_1...список_n — наборы значений ключа выбора; оператор_1...оператор_n+1 — любые операторы языка Паскаль; часть else <оператор_n+1> является необязательной.

8) Оператор цикла с параметром **for**:

for var <параметр_цикла>:= <начальное_значение> **to** <конечное_значение> **do** <тело_цикла>

Параметр_цикла — переменная целого типа, *тело_цикла* — любой оператор языка Паскаль (в том числе составной). Параметр цикла описывается внутри оператора цикла (например, так: **for var i := 1 to n do...**) и область его видимости ограничена только этим оператором.

9) Оператор цикла **loop**:

loop <выражение> **do** <оператор>;

Выражение должно иметь целый тип данных, оно указывает количество повторений тела цикла. Если значение выражения меньше или равно нулю, то тело цикла не выполняется ни разу. Цикл **loop** используется в простых ситуациях, когда тело цикла не зависит от номера итерации цикла.

10) Оператор цикла **foreach** имеет одну из следующих форм:

foreach <переменная> **in** <контейнер> **do** <оператор>

или

foreach <переменная>: <тип> **in** <контейнер> **do** <оператор>

или

foreach var <переменная> **in** <контейнер> **do** <оператор>

Контейнером может быть динамический или статический массив, строка, множество и т. д. Переменная цикла должна иметь тип, совпадающий с типом элементов контейнера. Переменная цикла пробегает все значения элементов контейнера и для каждого значения переменной цикла выполняется тело цикла. Изменение переменной цикла внутри тела цикла не меняет элементы контейнера, т. е. они доступны только для чтения.

11) Оператор цикла с предусловием:

while <условие> **do** <тело_цикла>;

Условие — выражение логического типа, *тело_цикла* — любой оператор языка Паскаль. *Тело_цикла* выполняется, пока *условие* имеет значение true. Если *тело_цикла* является составным оператором, его следует заключить в операторные скобки.

12) Оператор цикла с постусловием:

repeat <тело_цикла> **until** <условие>;

Тело_цикла выполняется, пока условие имеет значение false. Если *тело_цикла* является составным оператором, операторные скобки не нужны.

10. Типы данных

Тип данных определяет:

✓ множество допустимых значений, которое может принимать переменная или константа;

✓ набор допустимых операций, применимых к ней;

✓ формат внутреннего представления данных в памяти компьютера

Типы данных делятся на простые и структурированные.

Простые типы данных делятся на порядковые и вещественные.

Порядковые типы обладают свойствами упорядоченности и дискретности. К ним относятся: целые, логический, символьный, перечисляемый и тип-диапазон.

1) Целые типы `byte`, `shortint`, `smallint`, `word`, `longword` (`cardinal`), `integer` (`longint`), `int64`, `uint64`, `BigInteger`. Целые типы отличаются один от другого наличием или отсутствием знака и диапазоном значений, причем тип `BigInteger` имеет неограниченный диапазон (в предыдущих версиях Паскаля отсутствует).

2) Логический тип `boolean`, принимает два значения `true` (истина) и `false` (ложь). Обычно переменная логического типа является результатом проверки истинности некоторого условия.

3) Символьный тип `char`, значениями являются любые символы компьютера, закодированные целыми числами диапазона 0..255. К типу `char` применимы встроенные функции и операции отношения.

4) Перечисляемый тип задается перечислением значений, которые он может принимать.

5) Тип-диапазон — подмножество некоторого базового порядкового типа, задается границами значений внутри базового типа.

6) Вещественные типы: `single`, `real`, `double`, `decimal`. Все вещественные типы в `PascalABC.NET` являются знаковыми и отличаются диапазонами значений и точностью (количеством значащих цифр).

Структурированные типы данных

К структурированным типам относятся массив, запись, множество, файл, строка.

7) Массив — именованный упорядоченный набор элементов, принадлежащих к одному типу данных (простому или структурированному). В `PascalABC.NET` рассматриваются массивы двух видов — статические и динамические. Массив описывается либо как переменная с помощью зарезервированного слова `array` с указанием типа составляющих его элементов, например,

```
var mas: array[1..20] of real; // статический массив
var mas: array of real;      // динамический массив
либо через описание типа:
type mas = array[1..20] of real;
var m1: mas;
```

8) Запись — структура данных, состоящая из фиксированного количества компонентов, называемых полями записи. Поля могут принадлежать к разным типам данных. Запись объявляется в разделе описания типов, например:

```
type
  auto = record
    mark, color: string;
    number: integer
  end;
```

```
var ter1, ter2: auto;
```

Доступ к полям записи осуществляется по составному имени:

```
auto.mark:= Ford;
```

или с помощью оператора присоединения **with**:

```
with auto do
begin
```

```

number:= 1232;
color:= blue
end;

```

9) Множество представляет собой неупорядоченный набор элементов одного типа. Каждый элемент может входить во множество не более одного раза. Множество описывается с помощью зарезервированного слова **set**:

```
set of <базовый тип>;
```

Базовым типом может быть любой тип данных PascalABC.NET, кроме указателей.

10) Файл — именованная область внешней памяти компьютера или логическое устройство. Существует три вида переменных файлового типа:

- типизированный файл, определяется конструкцией `file of <тип>`, `тип` — любой тип языка, кроме типа `file`;
- нетипизированный (бестиповой) файл, определяется словом `file`;
- текстовый файл, определяется словом `text`.

Примеры: `ab1: file of string`; `ab2: file`; `ab3: text`.

Физический файл перед использованием в программе необходимо связать с переменной файлового типа с помощью процедуры `assign`:

`assign(<ф.п.>, <имя_файла>)`, где `ф.п.` — имя переменной файлового типа, `имя_файла` — имя физического файла (с указанием пути к нему) или логического устройства.

11) Строка — набор символов, вообще говоря, произвольной длины, задается с помощью зарезервированного слова `string`. Максимальная длина строки может быть указана в квадратных скобках после слова `string`:

```

var
  s1: string;
  s2: string[30];

```

12) Кортеж

Кортеж — это тип данных, аналогичный записи. Как и запись, кортеж имеет поля разных типов, но он проще в описании. Поля кортежа имеют определенные имена и являются неизменными: поменять поля кортежа после его создания невозможно.

Описать кортеж можно, например, так: `var a: (string, integer)`.

Можно также определить кортеж, используя автоопределение типа:

```
var g := ('Лето', 2020);
```

Кортеж позволяет поменять местами значения двух или более переменных без использования функций или дополнительных переменных:

```

(a, b) := (b, a);           // b принимает значение a и наоборот
(a, b, c) := (c, a, b);    // a принимает значение c, b — значение a,
                           // c — значение b

```

II. Задачи и упражнения

1. Алфавит языка PascalABC.NET. Оператор присваивания.

Ввод/вывод данных

Пример 1.1. Записать на языке PascalABC.NET следующие выражения:

$$1) \frac{\cos^2 x}{\sqrt[5]{(x-1)}} \quad 2) \sin 30^\circ \cdot \cos 45^\circ \quad 3) \sqrt[10]{a^9 + 4}$$

Решение. 1) `sqr(cos(x))/(x — 1)**(1/5);`

2) `sin(Pi/6)*cos(Pi/4);`

3) `(power(a, 9) + 4)**0.1.`

Пример 1.2. Присвоить переменной m значение первой цифры заданного трехзначного натурального числа k .

Решение. Для определения первой цифры трехзначного числа можно использовать операцию целочисленного деления `div`:

`m := k div 100.`

Пример 1.3. Пусть m и k — натуральные числа, $k = 23\,451$. Определить значение переменной m после выполнения следующих операторов присваивания:

1) `m := k div 10 mod 10`

2) `m := k div 1000 + k mod 100 div 6`

Решение. 1) $23\,451 \text{ div } 10 = 2\,345$; $2\,345 \text{ mod } 10 = 5$; $m = 5$.

2) $23\,451 \text{ div } 1000 = 23$; $23\,451 \text{ mod } 100 = 51$; $51 \text{ div } 6 = 8$; $m = 23 + 8 = 31$.

Пример 1.4. Вычислить значения выражений

1) `trunc(-3,8) + round(-2,7) + abs(-2)`

2) `5*14 div 7 mod 7 — trunc(sin(2))`

3) `succ(round(17/3)) — pred (-2)`

Решение. 1) `trunc(-3,8) = -3`; `round(-2,7) = -3`; `abs(-2) = 2`; **ответ: -4**;

2) $5*14 = 70$; $70 \text{ div } 7 = 10$; $10 \text{ mod } 7 = 3$; $0 < \sin(2) < 1 \Rightarrow$

`trunc(sin(2)) = 0`; **ответ: 3**;

3) `round(17/3) = 6`; `succ(6) = 7`; `pred(-2) = -3`; $7 - (-3) = 10$; **ответ: 10**.

Пример 1.5. Записать оператор присваивания в виде обычной математической формулы:

1) `y := 5**(x — 4) + tan(x*2);`

$$2) r := \text{sqr}(\ln(x + x^{**3})) / (x + \text{sqr}(x*x - 1)).$$

Решение. 1) $y = 5^{x-4} + \text{tg } 2x;$ 2) $r = \frac{\ln^2(x+x^3)}{x+\sqrt{x^2-1}}$

1.1. Записать выражения на языке Паскаль

1) $\frac{2^x-x}{\sqrt{4x+1}}$ 2) x^{115} 3) $x^{22} + 22^x$

4) $\sqrt[12]{a^5 + 4a}$ 5) $\sqrt[a]{x^a + a^x}$ 6) $\frac{-b+\sqrt{b^2-4ac}}{2a}$

7) $\frac{1}{3} \left(\left(\frac{\sin^2 x - \cos^2 x}{\sin \frac{x+y}{2}} - e^{|\cos x| + \sin x} \right) \ln x - \sqrt{x-1} \right)$

8) $x^2 \left(1 + \frac{x}{2!} + \frac{y}{3!} + \frac{z}{4!} \right)$ 9) $\frac{1+\sin^2(x+y)}{2+\left|\frac{x-2}{1+x^2y^2}\right|} + x$

10) $\sqrt{p(p-a)(p-b)(p-c)}$ 11) $\sqrt{p(p-a)^2(p-b)(p-c)^3}$

12) $\frac{1}{a+\frac{1}{b+c+\frac{1}{a}}}$ 13) 4^{x+1} 14) $|\sin x|^x$

15) $(b+d)^{4(f+x)}$ 16) $\text{tg } \frac{x}{y} \cdot \text{ctg } \frac{(x+1)}{4}$ 17) $\cos(a\pi) \cdot \text{tg } \frac{3\pi x}{(a+1)}$

18) $\frac{x+y^2}{|\text{tg}x+y|}$ 19) $\sqrt[4]{\sin^2 x + \cos x^2}$ 20) $(a^2 + 2b)^{12} - 12^{a+b}$

21) $\frac{\sqrt[4]{4 \ln(\sin x + 2)}}{|\cos x|}$ 22) $\frac{\ln x \cdot |z^2 - x^{12}|}{\text{tg}(x+4)}$ 23) $\frac{a \cdot b^3 - \sqrt{ac}}{ab - bc^6}$

24) $\text{tg}|x-t| \cdot 2^{-xt}$ 25) $\sqrt[7]{x^{33} + y^3}$ 26) $\frac{|\ln x| - x}{3x^4 + 4x}$

27) $\frac{4^{y+z} \cdot \sqrt{y+z}}{\sin(y+z)^2}$ 28) $(a+x)^{\cos|a-x|}$

1.2. Записать оператор присваивания в виде обычной математической формулы.

1) $y := \sin(\text{abs}(\text{sqr}(\ln(x)))) / \text{sqr}(x*x + x - 1)$

2) $y := \sin(\exp(6.5*\ln(x))) / \cos(\exp(6.5*\ln(x)))$

3) $y := x**\sin(x*x - 1)$

4) $y := (x + \sin(x))**x / \text{sqr}(x*x*x - 1)$

5) $y := \cos(x**(x+1)) - \sin(4*\ln(x))$

6) $z := \text{abs}(t*t*t - \exp(\sin(x))) - \text{sqr}(t/x)$

7) $z := \cos((t*t+1)**(-1/3))$

8) $z := \arctan(t / \text{sqr}(\text{abs}(1 - x*x)))$

9) $z := (\text{abs}(\text{sqr}(x) - \text{sqr}(t)))**(1/5)$

10) $f := (\ln((x+z)/(x*x+1)) + x) / (x*x - 1)$

$$11) f := \ln((x*x - z) / (\sqrt{x*x + 1} + x))$$

$$12) z := (\text{abs}(1 - t*t))^{1/6} * \sqrt{1 + t*t}$$

$$13) z := \text{sqr}(\arctan(t / \sqrt{1 + x*x}))$$

1.3. Пусть k и m — натуральные числа, $k = 742\,298$. Определить значение переменной m после выполнения следующих операторов присваивания:

1) $m := k \text{ div } 100 \text{ mod } 10$

2) $m := k \text{ div } 100 \text{ div } 10$

3) $m := k \text{ mod } (100 \text{ div } 10)$

4) $m := k \text{ mod } 1000 \text{ mod } 10$

5) $m := k \text{ mod } 1000 \text{ div } 10$

6) $m := k \text{ div } 10 \text{ mod } 100$

7) $m := k \text{ mod } 1000 \text{ div } 1000$

8) $m := k \text{ div } 100 \text{ mod } 100$

9) $m := k \text{ mod } (1000 \text{ div } 10)$

10) $m := k \text{ div } (1000 \text{ div } 10)$

1.4. Записать оператор присваивания для следующих действий:

1) переменной x присвоить значение полуразности a и b ;

2) удвоить значение переменной x ;

3) поменять знак переменной x на противоположный;

4) уменьшить значение переменной целого типа x на 1;

5) заменить значение вещественной переменной t его абсолютной величиной;

6) присвоить переменной логического типа q значение «ложь»;

7) значением переменной символьного типа сделать цифру 1;

8) сделать значением переменной логического типа nl значение выражения $x \geq 0$;

9) присвоить переменной a значение переменной b , а переменной b — значение переменной a ;

10) заменить значение переменной gr , задающей величину угла в градусах, величиной того же угла в радианах.

1.5. Присвоить переменной m значение последней цифры натурального числа k .

1.6. Присвоить переменной m значение остатка от деления двух натуральных чисел k и p .

1.7. Ввести 3 действительных числа и вывести их на экран в одной строке через запятую, каждое число должно иметь 2 десятичных знака.

1.8. Написать фрагмент программы для вывода на экран 3 действительных чисел. Каждое число должно быть выведено на отдельной строке с тремя десятичными знаками, после первых двух чисел должна стоять точка с запятой, после третьего — точка.

1.9. Ввести величины двух катетов прямоугольного треугольника. Вывести на экран значения:

- 1) длины гипотенузы;
- 2) площади треугольника;
- 3) меньшего из острых углов треугольника.

1.10. Ввести значения гипотенузы c и острого угла α прямоугольного треугольника. Вычислить и вывести на экран значение площади треугольника.

1.11. Диагональ квадрата равна d . Вычислить и вывести на экран его площадь и периметр.

1.12. Известна диагональ прямоугольника и угол между диагональю и большей стороной. Вычислить площадь прямоугольника.

1.13. Длины сторон треугольника равны a , b , c . Найти больший из углов треугольника.

1.14. Высота прямоугольного параллелепипеда равна h . Диагональ прямоугольника, лежащего в основании, равна d . Известно, что диагонали основания пересекаются под углом α . Найти объем параллелепипеда и площадь его поверхности.

1.15. В прямоугольном треугольнике известен катет a и площадь S . Найти величину гипотенузы c , второго катета b и углов α и β .

1.16. Площадь квадрата равна S . Вычислить сторону квадрата a , диагональ d и площадь S_1 круга, описанного около этого квадрата.

1.17. В равнобедренном треугольнике известно основание c и угол при нем α . Найти площадь треугольника S и величину боковой стороны a .

1.18. Даны три пары действительных чисел, которые являются координатами точек плоскости. Вычислить площадь треугольника с вершинами в этих точках.

1.19. Ввести двухзначное число и вывести на экран

- 1) его цифры в одной строке через запятую;
- 2) его цифры в обратном порядке через пробел;
- 3) сумму цифр.

1.20. Ввести трехзначное число и вывести на экран

- 1) первую и последнюю цифры;
- 2) сумму цифр;
- 3) число, которое получится, если первую и последнюю цифры поменять местами;
- 4) цифры через запятую в обратном порядке;
- 5) число, которое получится, если среднюю цифру заменить нулем.

1.21. Ввести четырехзначное число и вывести на экран

- 1) произведение первой и последней цифр;
- 2) вторую цифру;
- 3) сумму первой и третьей цифр;
- 4) все цифры в обратном порядке через запятую;
- 5) двухзначное число, которое получится, если отбросить первую и последнюю цифры.

1.22. Ввести действительное число и вывести на экран

- 1) его целую и дробную части в разных строках;
- 2) первую цифру его дробной части;
- 3) последнюю цифру его целой части;
- 4) двухзначное число, которое образуют две первые цифры дробной части.

1.23. Ввести действительные числа a , b , β . Вычислить площадь треугольника, стороны которого равны a и b , а угол между ними равен β . Считайте, что угол β измеряется

- 1) в радианах;
- 2) в градусах.

1.24. Ввести 2 действительных числа. Вывести на экран коэффициенты приведенного квадратного уравнения, корнями которого эти числа являются.

2. Простые типы данных. Совместимость типов.

Преобразование типов

Пример 2.1. Вычислить значение выражения $\text{int}(\sin(1)) + \text{round}(3.8) - \text{trunc}(-3.8)$.

Решение. $0 < \sin 1 < 1 \Rightarrow \text{int}(\sin 1) = 0$; $\text{round}(3.8) = 4$; $\text{trunc}(-3.8) = -3$;

ответ: 7.

Пример 2.2. Найти значение логического выражения $(\text{not } a \text{ or } b)$ and c , если $a = \text{true}$; $c = \text{true}$; $b = \text{false}$.

Решение. $\text{not } a = \text{false}$, $b = \text{false} \Rightarrow (\text{not } a \text{ or } b) = \text{false} \Rightarrow (\text{not } a \text{ or } b) \text{ and } c = \text{false}$;

ответ: false.

Пример 2.3. Пусть x — переменная вещественного типа. Вычислить значение логического выражения $(\text{abs}(\sin(x)) > 1)$ or $(\exp(x) > 0)$ and $(x*x > -1)$.

Решение. $\exp(x) > 0 = \text{true}$, $x*x > -1 = \text{true} \Rightarrow (\exp(x) > 0) \text{ and } (x*x > -1) = \text{true} \Rightarrow (\text{abs}(\sin(x)) > 1) \text{ or } (\exp(x) > 0) \text{ and } (x*x > -1) = \text{true}$; **ответ:** true.

2.1. Вычислить значения выражений

- 1) $\sin(\text{sqr}(x) - 1) + 2 * \text{abs}(y) / \cos(2 + y)$ при $x = 1$, $y = -2$
- 2) $\text{trunc}(6,9) - \text{trunc}(6,1) - 1$
- 3) $\text{round}(1,3) + \text{round}(1,6) + 2$
- 4) $\text{trunc}(-1,8) + \text{round}(-1,8) + 1$
- 5) $\text{trunc}(-2,7) + \text{round}(-1,6) + \text{abs}(-5)$
- 6) $\text{round}(0,5) + \text{round}(-0,5) + \text{abs}(0,5)$
- 7) $6,45 - \text{trunc}(6,45)$
- 8) $2,48 + \text{trunc}(-2,48)$
- 9) $2 * \text{frac}(12,5) - \text{round}(-1,6)$
- 10) $30 \text{ div } 6 - 30 \text{ mod } 5 + \text{succ}(2)$
- 11) $32 \text{ div } 5 + 32 \text{ mod } 6 + \text{pred}(4)$

- 12) $3 * 7 \text{ div } 2 \text{ mod } 7 / 3$
- 13) $\text{trunc}(\sin(1))$
- 14) $2 * 23 \text{ div } 3 \text{ mod } 3 - \text{trunc}(\sin(3))$
- 15) $\text{succ}(\text{round}(19/4)) - \text{pred}(5)$
- 16) $4 * \arctan(0) + \text{abs}(-3)$

2.2. Пусть m — целое число, $z = 327,12$. Определить значение переменной m после выполнения следующих операторов присваивания:

- 1) $m := \text{trunc}(z * 10) \text{ div } 100$
- 2) $m := \text{trunc}(z * 100) \text{ mod } 10$
- 3) $m := \text{frac}(z) * 100$
- 4) $m := \text{frac}(z) * 100 \text{ mod } 10$
- 5) $m := \text{round}(z) \text{ mod } 100$
- 6) $m := \text{round}(z) \text{ div } 10$
- 7) $m := \text{round}(z * 10) \text{ mod } 1000$
- 8) $m := \text{frac}(z / 10) * 1000 \text{ mod } 100$
- 9) $m := \text{trunc}(z / 10) \text{ div } 10$
- 10) $m := \text{trunc}(\text{trunc}(z * 10) / 100)$

2.3. Пусть m — целое число, $z = -213,64$. Определить значение переменной m после выполнения следующих операторов присваивания:

- 1) $m := \text{int}(z) - \text{round}(z)$
- 2) $m := \text{int}(z) - \text{trunc}(z)$
- 3) $m := \text{frac}(z) * 10$
- 4) $m := \text{frac}(z) * 100 \text{ div } 16$
- 5) $m := \text{trunc}(\text{int}(z) / 10)$
- 6) $m := \text{int}(\text{round}(z) / 100)$

2.4. Пусть a и b — переменные логического типа. Доказать тождества

- | | |
|---|---|
| 1) $\text{true or } a \equiv \text{true}$ | 2) $a \text{ and false} \equiv \text{false}$ |
| 3) $a \text{ or } (\text{not } a) \equiv \text{true}$ | 4) $a \text{ and } (\text{not } a) \equiv \text{false}$ |
| 5) $a \text{ and } (a \text{ or } b) \equiv a$ | 6) $a \text{ or } (a \text{ and } b) \equiv a$ |

2.5. Найти значение логического выражения $(a \text{ and not } b) \text{ or } c$, если

- 1) $a, b, c = \text{true}$
- 2) $a, b, c = \text{false}$
- 3) $a, b = \text{true}; c = \text{false}$
- 4) $a, c = \text{true}; b = \text{false}$,
- 5) $a, c = \text{false}; b = \text{true}$,

2.6. Пусть x — переменная вещественного типа, a — переменная логического типа. Вычислить значение логических выражений

- 1) $(x > 7) \text{ and } (x \leq 3)$
- 2) $(\text{abs}(\sin(x)) \leq 1) \text{ or } (\exp(x) < 0)$
- 3) $\text{not } (x * x + \text{abs}(x) < 0) \text{ and true}$
- 4) $(1/x > x) \text{ and not } (1 + x * x > 0) \text{ or } (132 < 13.2 * 10) \text{ or } (x * x - 2 * x + 1) < 0)$
- 5) $\text{not } (12.5 > 25/2) \text{ and } (x < x * x) \text{ and not } (-5 * 6 = (7.5 * (-4)))$
- 6) $(a \text{ or } (\text{not } a)) \text{ and odd}(345) \text{ or } (\text{sqr}(x) + 3 \leq 4)$

2.7. Нарисовать на координатной плоскости область, для которой истинными являются выражения

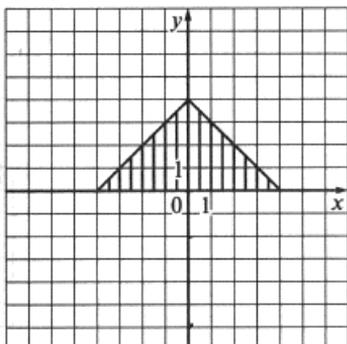
- 1) $x \geq 0$;
- 2) $(x > 0) \text{ or } (y < 0)$;
- 3) $(x \leq 0) \text{ and } (y \geq 0)$;
- 4) $(x \geq 1) \text{ and } (y \leq 1)$;
- 5) $(y > -2) \text{ and } (y < 2)$;
- 6) $(x < 0) \text{ or } (x \geq 5)$;
- 7) $(y \geq x) \text{ and } (y + x \geq 0) \text{ and } (y \leq 1)$;
- 8) $(\text{abs}(x) \leq 1) \text{ or } (\text{abs}(y) \geq 1)$;
- 9) $(y \geq -x - 1) \text{ and } (x \leq 0) \text{ and } (y \leq 0) \text{ or } ((x = y) \text{ and } (x \geq 0))$;
- 10) $(y = 0) \text{ and } ((x \geq -2) \text{ and } (x \leq -1) \text{ or } (x \geq 1) \text{ and } (x \leq 2) \text{ or } (x = 0) \text{ and } (y \geq 2))$;
- 11) $(x^2 + y^2 < 4) \text{ and } (y \geq -1) \text{ and } (y \leq 1)$;
- 12) $(\text{abs}(x) + \text{abs}(y) \leq 2) \text{ or } (x > 2) \text{ or } (x < -2)$;
- 13) $(x^2 + y^2 \leq 4) \text{ and } (x^2 + y^2 \geq 1)$;
- 14) $(x = 0) \text{ and } (y < 0) \text{ or } (y \geq 4) \text{ and } (y \leq 7)$;
- 15) $(x^2 + y^2 \leq 4) \text{ or } (y < 0)$;
- 16) $(x + y \geq 0) \text{ and } (x + y \leq 5) \text{ or } (x = 0) \text{ or } (y = 0)$.

2.8. Записать следующие условия в виде логических выражений:

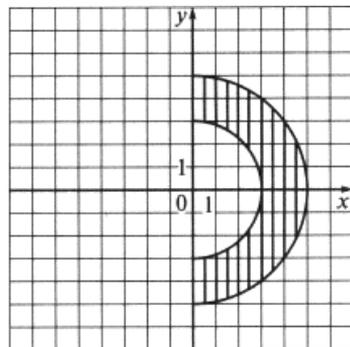
- 1) числа a и b имеют одинаковый знак;
- 2) хотя бы одно из чисел a, b, c отрицательно;
- 3) все числа a, b, c неотрицательны;
- 4) целые числа m и n имеют одинаковый знак и оба четные;
- 5) по крайней мере одна из двух заданных логических переменных имеет значение true;
- 6) две логические переменные имеют одинаковое значение;
- 7) целое число x неотрицательно и кратно 7;
- 8) цифры в двухзначном числе различны;
- 9) в двухзначном числе обе цифры нечетные;
- 11) десятичная запись трехзначного числа содержит ноль;
- 12) заданное натуральное число является трехзначным;
- 13) точка с координатами $(x; y)$ находится вне квадрата, образованного отрезками прямых $x = 1, x = -1, y = 1, y = -1$;
- 14) точка с координатами $(x; y)$ находится между ветвями параболы $y = x^2$.

2.9. Описать заштрихованные области с помощью логических выражений.

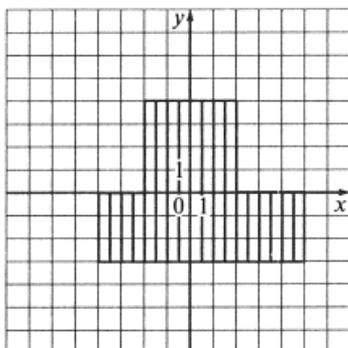
1)



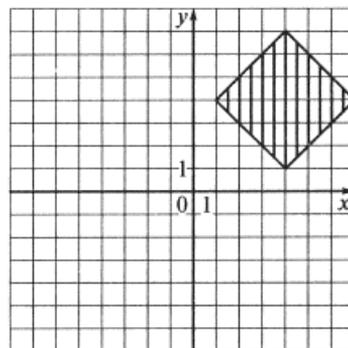
2)



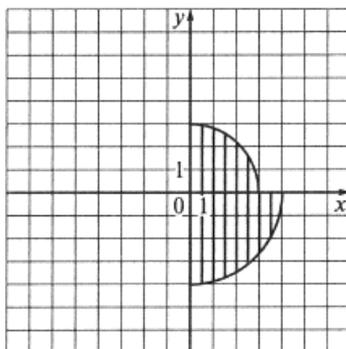
3)



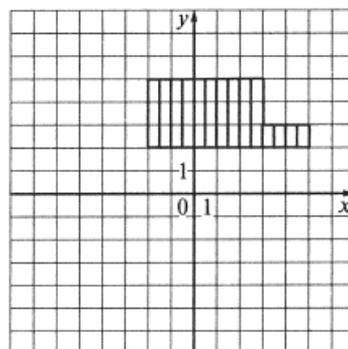
4)



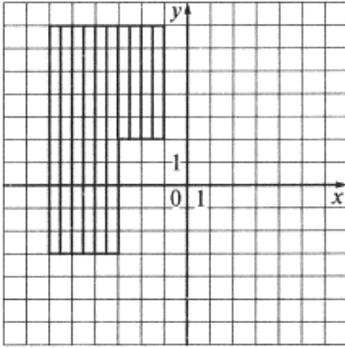
5)



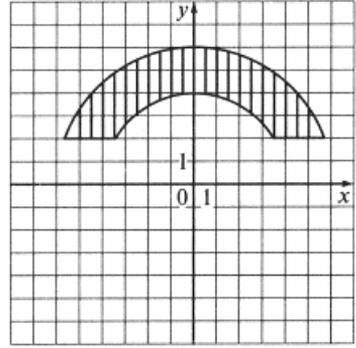
6)



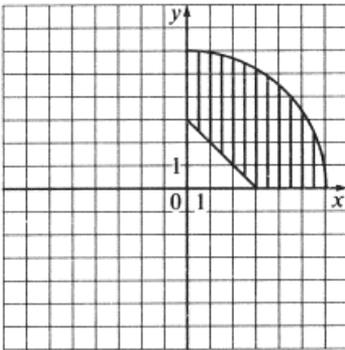
7)



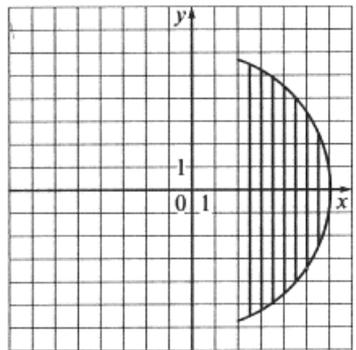
8)



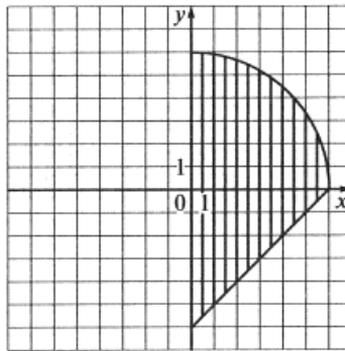
9)



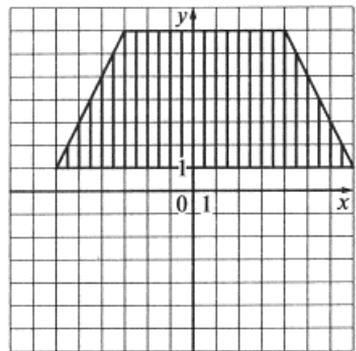
10)



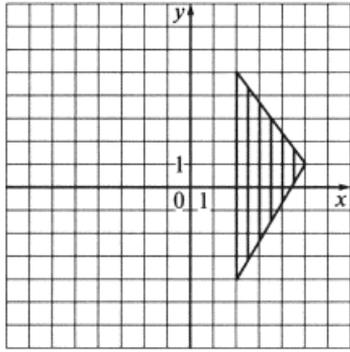
11)



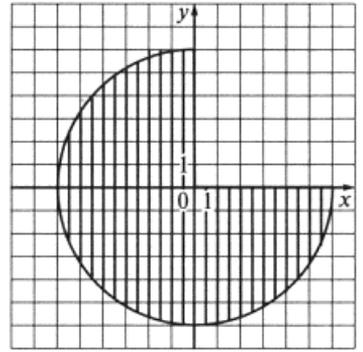
12)



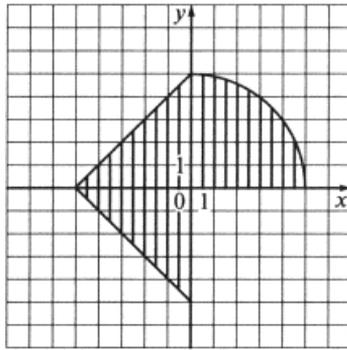
13)



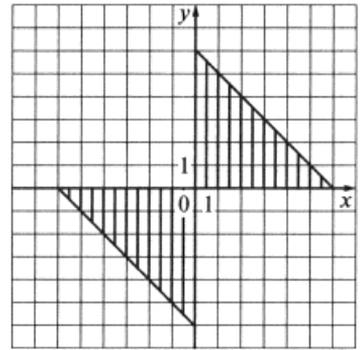
14)



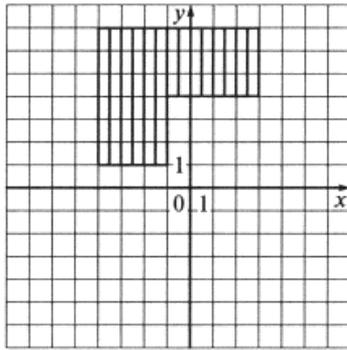
15)



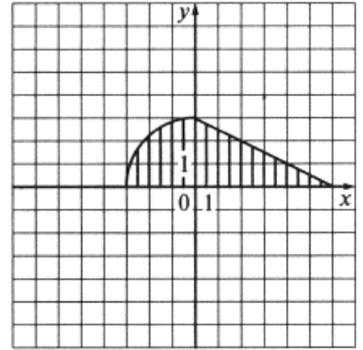
16)



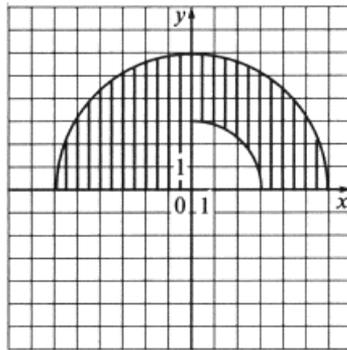
17)



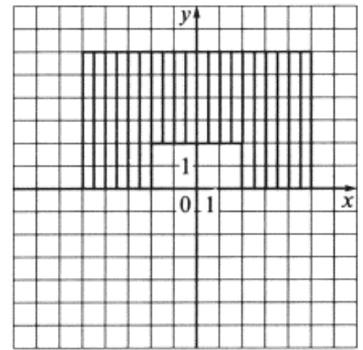
18)



19)



20)



3. Реализация линейных и разветвляющихся алгоритмов

Пример 3.1. Записать условный оператор в виде обычной математической формулы

```
if x >= 1 then y:= ln(sqrt(x*x + 1)) else y:= sin(x - 2)**5.
```

Решение. $y = \begin{cases} \sin(x - 2)^5, & x < 1 \\ \ln \sqrt{x^2 + 1}, & x \geq 1 \end{cases}$

Пример 3.2. Написать код программы, которая выводит на экран значение *true*, если сумма цифр заданного трехзначного числа больше 10, и значение *false* в противном случае.

Решение.

1 вариант

```
• program ex321;
  begin
    var a1 := ReadInteger('Введите трехзначное число');
    var b1 := a1 div 100;           // первая цифра
    var b2 := a1 div 10 mod 10;    // вторая цифра
    var b3 := a1 mod 10;           // третья цифра
    if b1 + b2 + b3 > 10 then writeln('true')
      else writeln('false')
    end. •
```

2 вариант

```
• program ex322;
  begin
    var a1 := ReadInteger('Введите трехзначное число');
    {логической переменной b присваивается значение
     условия «сумма цифр больше 10»}
    var b := (a1 div 100 + a1 div 10 mod 10 + a1 mod 10) > 10;
    writeln(b)
  end. •
```

Пример 3.3. По заданному году и номеру месяца вывести на экран количество дней в этом месяце.

Решение.

```
• program ex33;
  begin
    var (a, m) := ReadInteger2('Введите год и номер месяца');
    case m of
      1, 3, 5, 7, 8, 10, 12: writeln('в этом месяце 31 день');
      4, 6, 9, 11: writeln('в этом месяце 30 дней');
      2: if a mod 4 = 0 then writeln('в этом месяце 29 дней')
          else writeln('в этом месяце 28 дней')
    end;
  end. •
```

3.1. Присвоить переменной A значение 1, если:

- 1) переменная X принадлежит интервалу $(3; 14)$;
- 2) переменная X принадлежит отрезку $[-1; 5]$;
- 3) переменная X принадлежит промежутку $[-2; +\infty)$;
- 4) переменная X принадлежит одному из промежутков $[-3; 0)$ или $[5; +\infty)$;
- 5) точка с координатами x, y принадлежит кругу радиуса R с центром в начале координат;
- 6) точка с координатами x, y принадлежит окружности радиуса R с центром в начале координат.

3.2. Записать условный оператор в виде обычной математической формулы

- 1) if $x < 0$ then $y := \sin(x)$ else $y := (x*x + 1)**(1/3)$
- 2) if $x < 0$ then $y := \ln(\sqrt{\text{abs}(x)})$ else $y := \arcsin(\sqrt{x})$
- 3) if $x <= -2$ then $y := \ln(\text{abs}(\sin(x))+2)$ else $y := \cos(\sqrt{\text{abs}(x)})$
- 4) if $x >= 1$ then $y := \text{abs}(x**(1/5) - x*x)$ else $y := \ln(\text{abs}(x - 5))$
- 5) if $x >= 0$ then $y := \tan(x**(1/4))$ else $y := \ln(\text{abs}(x - \sqrt{\sin(x)}))$
- 6) $a := x > 1$; if a then $y := \sqrt{x**4 - x}$ else $y := \text{abs}(x - 1)$
- 7) $a := (x >= 0)$ and $(x <= 5)$; if a then $y := 1.5**x$ else $y := 1$
- 8) $a := (x < -2)$ or $(x > 2)$; if a then $y := x*x$ else $y := 4$
- 9) $a := (x >= -3)$ and $(x <= -1)$; $b := (x > 0)$; if a or b then $y := 1$ else $y := 0$
- 10) $a := x <= 0$; $b := (x = 1)$ or $(x = 3)$; if a or b then $y := x$ else $y := 1$

3.3. Написать оператор, в результате выполнения которого логическая переменная t принимает значение *true*, если выполняется указанное условие, и значение *false*, если оно не выполняется:

- 1) числа x, y, z равны между собой;
- 2) хотя бы два из чисел x, y, z равны между собой;
- 3) целые числа m и n имеют одинаковую четность;
- 4) только одна из двух заданных логических переменных имеет значение *true*;
- 5) x — положительное число;
- 6) цифра 7 входит в десятичную запись трехзначного числа k хотя бы один раз;
- 7) из чисел x, y, z хотя бы два положительные;
- 8) точка с координатами $(x; y)$ попадает во внутреннюю часть круга радиуса r с центром в начале координат;
- 9) число a является четным и отрицательным;
- 10) числа a и b имеют одинаковый знак и хотя бы одно из них кратно 5.

3.4. Написать код программы, которая выводит на экран значение *true*, если высказывание является истинным, и значение *false* в противном случае.

- 1) сумма двух первых цифр данного четырехзначного числа равна сумме двух его последних цифр;
- 2) сумма цифр заданного трехзначного числа является:
 - а) четным числом,
 - б) нечетным числом;

- 3) сумма цифр данного трехзначного числа больше квадрата его первой цифры;
- 4) заданное целое число является нечетным двухзначным;
- 5) среди чисел a, b, c есть хотя бы два одинаковых;
- 6) все цифры заданного четырехзначного числа различны;
- 7) точка с координатами $(x; y)$ находится во второй четверти координатной плоскости;
- 8) точка с координатами $(x; y)$ не принадлежит ни одной из координатных осей;
- 9) точка $A(x; y)$ принадлежит прямоугольнику, заданному координатами левого верхнего угла $(x_1; y_1)$ и правого нижнего угла $(x_2; y_2)$;
- 10) цифры заданного трехзначного числа образуют арифметическую прогрессию;
- 11) по крайней мере одна из цифр заданного четырехзначного числа является четной;
- 12) одна из двух последних цифр целой части действительного числа является единицей;
- 13) одна из трех первых цифр дробной части действительного числа является нулем;
- 14) заданное трехзначное число делится без остатка на свою среднюю цифру.

3.5. Составить программу для вычисления значений функции (предусмотреть ввод значения x и вывод на экран значений x и y)

$$1) y = \begin{cases} 2^x, & x \leq 0 \\ x + 1, & x > 0 \end{cases}$$

$$2) y = \begin{cases} \operatorname{tg}(x - 1), & x \leq 1 \\ \ln x, & x > 1 \end{cases}$$

$$3) y = \begin{cases} x + \sqrt{|x|}, & x \leq 0 \\ e^{x-2}, & x > 0 \end{cases}$$

$$4) y = \begin{cases} -1, & x < -1 \\ x^2 - \sin x, & -1 \leq x \leq 1 \\ 0, & x > 1 \end{cases}$$

$$5) y = \begin{cases} x^2 - 1, & x \leq -2 \\ (x - 5)^2, & -2 < x \leq 5 \\ |\cos x|, & x > 5 \end{cases}$$

$$6) y = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

$$7) y = \begin{cases} \operatorname{tg}(x^2 + x), & |x| \leq 1 \\ \ln|x|, & |x| > 1 \end{cases}$$

$$8) y = \begin{cases} \ln|x^2 - 1|, & x \neq \pm 1 \\ 1, & |x| = 1 \end{cases}$$

$$9) y = \begin{cases} -x, & |x| < 1 \\ 0, & x \leq -1 \\ e^{x-1}, & x \geq 1 \end{cases}$$

$$10) y = \begin{cases} \sin(2^{\cos x}), & |x| \leq 2 \\ \operatorname{tg} 2^{\cos x}, & x < -2 \\ |\cos x|, & x > 2 \end{cases}$$

3.6. Для произвольных действительных чисел a, b, c определить, имеет ли уравнение $ax^2 + bx + c = 0$ хотя бы один действительный корень.

3.7. Определить, есть ли среди цифр заданного трехзначного числа одинаковые.

3.8. Для четырехзначного числа проверить, равна ли сумма двух его крайних цифр сумме двух средних.

3.9. Ввести 3 числа и определить, могут ли они быть длинами сторон треугольника. Если да, то определить вид треугольника (остроугольный, прямоугольный, тупоугольный).

3.10. Ввести 3 числа. Если они являются длинами сторон прямоугольного треугольника, вывести их в порядке возрастания и вычислить площадь треугольника.

3.11. Ввести 3 числа. Если они могут быть длинами сторон остроугольного треугольника, вывести их в порядке убывания и вычислить периметр треугольника.

3.12. Ввести 3 числа. Если они могут быть длинами сторон треугольника, найти синус меньшего из его углов.

3.13. Ввести 3 числа. Если они могут быть длинами сторон равнобедренного треугольника, вычислить его длины его высот. Вывести на экран длину основания и длины высот.

3.14. Ввести координаты трех точек на плоскости. Если они являются вершинами прямоугольного треугольника, вычислить его площадь.

3.15. Ввести координаты трех точек на плоскости. Если они являются вершинами вывести треугольника, вывести на экран соответствующее сообщение и длины сторон в порядке убывания.

3.16. Ввести координаты трех точек на плоскости и действительное число R . Вывести на экран координаты тех точек, которые лежат внутри круга радиуса R с центром в начале координат, если таких точек нет, вывести число 0.

3.17. Ввести координаты трех точек на плоскости. Если они являются вершинами равностороннего треугольника, вычислить и вывести на экран его площадь и высоту.

3.18. Ввести координаты трех точек на плоскости. Если они являются вершинами равнобедренного треугольника, вычислить длины его высот. Вывести длины высот и боковой стороны в порядке убывания значений.

3.19. Даны три целых числа a, b, c . Если минимум для a и b больше, чем минимум для a и c , то присвоить переменной t значение 0, иначе — значение 1. Вывести на экран значение t .

3.20. Даны четыре целых числа a, b, c, d . Отсортировать эти числа по убыванию и вывести на экран.

3.21. Дан квадрат с центром в начале координат и сторонами, образующими угол 45° с осями координат. Сторона квадрата равна заданному действительному числу b . Дана точка M с координатами (x, y) . Если точка M находится внутри квадрата, то вывести на экран сообщение «Да», иначе — сообщение «Нет».

3.22. Даны три действительных числа a, b, c . Вывести на экран сообщение «Да», если хотя бы одно из этих чисел:

- 1) не содержит дробной части;
- 2) является квадратом целого числа;
- 3) отрицательно.

В противном случае вывести сообщение «Нет».

3.23. Даны три целых числа a, b, c . Если два из них являются отрицательными и четными, то вывести на экран сообщение «Да», иначе — сообщение «Нет».

3.24. Даны три целых числа a, b, c . Если среди них два одинаковых, то вывести на экран сообщение «Да», иначе — сообщение «Нет».

3.25. Даны два натуральных числа a и b . Максимальное из чисел возвести в квадрат, из минимального извлечь квадратный корень. Если числа равны — возвести a в степень b . Результаты вывести на экран.

3.26. Даны три целых числа a, b, c . Если все модули разностей этих чисел меньше заданного числа K , то вывести на экран число 3, иначе — число 1.

3.27. Даны четыре натуральных трехзначных числа a, b, c, d . Если во всех числах вторая цифра является нулем, то вывести на экран сообщение «Да», иначе — сообщение «Нет».

3.28. Программа запрашивает у пользователя номер месяца и выводит на экран соответствующее название времени года. В случае, если пользователь укажет недопустимое число, программа должна вывести сообщение «Ошибка ввода данных».

3.29. Запросить у пользователя номер дня недели и вывести соответствующее сообщение: «Рабочий день», «Суббота» или «Воскресенье». Предусмотреть проверку правильности ввода данных.

3.30. По заданному номеру месяца определить и вывести на экран количество дней в этом месяце в невисокосный год. Предусмотреть проверку правильности ввода данных.

3.31. Ввести номер времени года (1 — зима, 2 — весна, 3 — лето, 4 — осень) и вывести на экран соответствующие этому времени года месяцы и число дней в каждом месяце (в невисокосном году).

3.32. Программа выводит на экран сообщение «До окончания работы осталось N дней». Число N вводится с клавиатуры и в зависимости от его значения слово «день» должно иметь правильную форму (21 день, 53 дня, 5 дней и т. д.). Выполнить задание с помощью

- а) оператора выбора;
- б) условного оператора.

3.33. Ввести целое число k от 1 до 99 и вывести на экран фразу «Мне k лет», где слово «лет» имеет правильную форму (год, года, лет). Предусмотреть проверку правильности ввода данных.

3.34. Программа запрашивает у пользователя 2 действительных числа и знак арифметической операции, и в зависимости от введенного знака выводит на экран результат соответствующей операции.

3.35. По заданной последней цифре целого числа определить последнюю цифру квадрата этого числа.

3.36. Прочитать двухзначное число и вывести на экран его словесное описание (например, 75 — семьдесят пять).

3.37. Элементами круга являются: 1 — радиус, 2 — диаметр, 3 — длина окружности. По заданному номеру элемента выполнить запрос соответствующего значения и вычислить площадь круга.

3.38. Определить сумму цифр заданного k -значного числа, где $1 \leq k \leq 4$.

3.39. По введенной дате (число и номер месяца) определить номер дня в невисокосном году. Выполнить проверку правильности ввода данных.

4. Программирование циклических алгоритмов

Пример 4.1. Вывести на экран 4 раза в одной строке через пробел слово «Привет!».

Решение

1 вариант — с помощью оператора `loop`:

```
• program ex411;  
  begin  
    loop 5 do println('Привет!')  
  end. •
```

2 вариант — с помощью оператора `for`:

```
• program ex412;  
  begin  
    for var i := 1 to 5 do  
      println('Привет!')  
    end. •
```

Пример 4.2. Вывести на экран таблицу умножения натуральных чисел.

Решение

Вариант 1 (вывод в столбик в формате ' $i*j = \dots$ ', используется процедура `print`)

```
• program ex421;  
  begin  
    for var i := 1 to 9 do  
      for var j := 1 to 9 do  
        begin  
          print(i, '*', j, '=', i*j);  
          writeln;  
        end;  
      end. •
```

Вариант 2 (вывод в виде таблицы, используется процедура write с форматом)

```
• program ex422;
  begin
    write(' '); // отступ строки заголовка
    for var i := 1 to 9 do write(i:4); // строка заголовка
    writeln; // перевод курсора в новую строку
    for var i := 1 to 9 do
      begin
        write(i:4);
        for var j := 1 to 9 do write(i*j:4);
        writeln;
      end;
    end. •
```

Пример 4.3. С клавиатуры по одному вводятся 10 целых чисел. Вывести на экран все числа, которые при делении на 9 дают остаток 4 или 5.

Решение

```
• program ex43;
  const n = 10;
  begin
    for var i := 1 to n do
      begin
        var k := ReadInteger('Введите целое число');
        if (k mod 9 = 4) or (k mod 9 = 5) then println(k)
      end;
    end. •
```

Пример 4.4. Составить таблицу значений функции

$$y = \begin{cases} e^{x^2+1}, & x < 0 \\ e + \ln(x+1), & x \geq 0 \end{cases} \text{ на отрезке } [-2; 2] \text{ с шагом } h = 0,2.$$

Решение

Вариант 1 (с помощью оператора цикла с предусловием)

```
• program ex441;
  begin
    { В следующем операторе присваивания используется кортеж из
      трех действительных переменных }
    var (a, b, h) := ReadReal3('Введите границы отрезка a и b
      и шаг h');
    var x := a; // аргумент x, тип определяется по присваиванию
    var y: real; // переменная для функции y
    while x <= b do
      begin
        if x < 0 then y := exp(x*x + 1)
        else y := exp(1) + ln(x + 1);
      end;
    end;
  end.
```

```

        writeln('x = ', x:4:2, ' y = ', y:6:3);
        x := x + h
    end;
end. •

```

Вариант 2 (с помощью оператора цикла с параметром, определение переменных a, b, h, x, y выполняется так же, как в первом варианте)

```

• program ex442;
  begin
    var (a, b, h) := ReadReal3('Введите границы отрезка a и b
                               и шаг h');

    var x := a;
    var y: real;
    {переменная n показывает, сколько раз шаг h «помещается» на отрезке
[a; b]}
    var n := trunc((b - a)/h); //тип n определяется по присваиванию
    for var i := 0 to n do
      begin
        if x < 0 then y := exp(sqrt(x + i*h) + 1)
        else y := exp(1) + ln((x + i*h) + 1);
        writeln('x = ', (x + i*h):4:2, ' y = ', y:6:3);
        x := x + h;
      end;
    end. •

```

Пример 4.5. Вычислить сумму сходящегося ряда с заданной точностью ε

$$\sum_{i=1}^{\infty} \frac{i^3}{i^8 + 1}$$

Решение

```

• program ex45;
  begin
    var eps := ReadReal('Введите точность eps');
    var s := 0.0; // инициализация переменных s и i
    var i := 1; // с одновременным определением типов
    var a: real; // вспомогательная переменная для очередного
                // слагаемого

    repeat
      a := i*i*i/(exp(8*ln(i)) + 1);
      s += a;
      inc(i)
    until a < eps;
    writeln('s = ', s:6:4)
  end. •

```

- 4.1. Вывести на экран 10 раз в столбик свое имя и фамилию.
- 4.2. Вывести на экран таблицу квадратов первых десяти натуральных чисел.
- 4.3. Вывести на экран таблицу кубов первых десяти положительных нечетных чисел.
- 4.4. Написать код программы, которая вычисляет сумму первых n целых положительных чисел. Количество суммируемых чисел должно вводиться во время работы программы.
- 4.5. Написать код программы, которая вычисляет сумму первых n положительных четных чисел. Количество суммируемых чисел должно вводиться во время работы программы.
- 4.6. Написать код программы, которая вычисляет сумму квадратов первых n членов ряда 1, 3, 5, 7, Количество суммируемых членов ряда задается во время работы программы.
- 4.7. Вывести на экран таблицу степеней двойки (от нулевой до десятой).
- 4.8. С клавиатуры по одному вводятся 20 действительных чисел. Вывести на экран
- 1) все отрицательные числа,
 - 2) сумму всех положительных чисел;
 - 3) все числа, принадлежащие промежутку $[-1; 1]$;
 - 4) среднее арифметическое всех чисел;
 - 5) все числа, которые меньше заданного числа (число вводится пользователем);
 - 6) все числа, которые больше первого введенного числа;
 - 7) наибольшее и наименьшее числа;
 - 8) количество положительных чисел, количество отрицательных чисел и количество нулей;
 - 9) сумму всех чисел с нечетными номерами;
 - 10) последние 10 чисел.
- 4.9. С клавиатуры по одному вводятся 20 целых чисел. Вывести на экран:
- 1) все положительные числа;
 - 2) все четные числа
 - 3) все отрицательные нечетные числа;
 - 4) все числа, кратные 7;
 - 5) сумму всех отрицательных чисел, кратных 3;
 - 6) те числа, которые по модулю меньше своего порядкового номера;
 - 7) наибольшее из отрицательных чисел;
 - 8) все числа, не принадлежащие отрезку $[2; 12]$;
 - 9) все нечетные числа и сумму чисел с одиннадцатого по двадцатое;
 - 10) если три первые числа четные, то сумму последних 10 чисел, в противном случае — сумму первых 10 чисел;
 - 11) если первое число больше второго, то сумму последних 10 чисел, в противном случае — сумму всех чисел;
 - 12) если два первых числа отрицательные и четные, то сумму модулей всех чисел, в противном случае — все отрицательные числа из первых 15 чисел.

4.10. Фрагмент кода выполняет вычисление конечной суммы

$$\sum_{n=1}^7 (n^2 + 1) \cdot e^{-n} :$$

```
var s := 0.0;
```

```
for var n := 1 to 7 do
```

```
s += (n*n + 1)/exp(n);
```

Замените оператор цикла for оператором цикла

а) с предусловием;

б) с постусловием.

4.11. Написать программу для вычисления конечной суммы

$$1) \sum_{i=1}^{50} \frac{\sin i}{i}$$

$$2) \sum_{i=1}^{20} \frac{\ln i^3}{i^2 + 1}$$

$$3) \sum_{i=1}^{25} (\sqrt{i} + 5 \sin i)$$

$$4) \sum_{i=1}^{40} \frac{2^{i-1}}{i^{12} + 1}$$

$$5) \sum_{i=1}^7 \left(3^{-i} + \ln \frac{2}{i} \right)$$

$$6) \sum_{i=2}^{12} \frac{\operatorname{tg} i^4}{i^2 - 1}$$

$$7) \sum_{i=1}^{18} (\sqrt[3]{i} + \cos^2 i)$$

$$8) \sum_{i=1}^{10} \frac{4^{i+1} - 4i^2}{\sqrt[5]{(i+2)}}$$

$$9) \sum_{i=1}^{70} \left(\sqrt[8]{i} + \ln \frac{\sqrt{i}}{8} \right)$$

$$10) \sum_{i=1}^5 \left(\frac{i^2}{7} + \ln(i^2 - 7) \right)$$

$$11) \sum_{i=1}^{15} \frac{2^{-i+2}}{\cos(i^2 + 3)}$$

$$12) \sum_{i=1}^9 (i + \operatorname{tg} i)$$

4.12. Вычислить конечные суммы из задания 4.11 с помощью операторов цикла с предусловием и постусловием. l'm

4.13. Составить таблицу значений функции $y = f(x)$ на отрезке $[a; b]$ с шагом h . Выполнить задание с помощью оператора цикла

а) с предусловием; б) с постусловием; в) с параметром.

Значения a, b, h задаются во время работы программы.

1) $y = \cos x^2 + \ln(x^2 + 1); a = -2, b = 4, h = 0,5$

2) $y = \operatorname{tg}^2(x - 1); a = -4, b = 0, h = 0,4$

3) $y = \frac{x}{2x^2+2}; a = 0, b = 2, h = 0,2$

4) $y = \sqrt[4]{\ln x + x^2}, a = 1, b = 3, h = 0,2$

5) $y = \sqrt[5]{\sin x - \cos x^2}, a = -1, b = 1, h = 0,2$

$$6) y = \frac{2^{x+1} - x}{x^2 + 1}, a = -2, b = 2, h = 0,4$$

$$7) y = \frac{\operatorname{tg}^2(x+5)}{\sin x + 2}, a = -4, b = 0, h = 0,4$$

$$8) y = \frac{\ln|1 - (x+5)^{12}|}{e^{x+1}}, a = 1, b = 3, h = 0,2$$

$$9) y = \begin{cases} \sqrt{x^3 - 1}, & x \leq 2 \\ \ln|1 - x|, & x > 2 \end{cases}, a = 1, b = 4, h = 0,3$$

$$10) y = \begin{cases} \ln(1 - x), & x \leq 0 \\ x^2 - 1, & x > 0 \end{cases}; a = -5, b = 3, h = 0,8$$

$$11) y = \begin{cases} 3^{x^2+x}, & x < 1 \\ \sqrt{(x^2+x)}, & x \geq 1 \end{cases}; a = -1, b = 3, h = 0,4$$

$$12) y = \begin{cases} \frac{x + \sin x}{x^2 + 1}, & x \leq -1 \\ \sqrt{(x^2 + x)}, & x > -1 \end{cases} \quad a = -3, b = 3, h = 0,5$$

$$13) y = \begin{cases} \operatorname{tg} x, & |x| \leq 1 \\ \cos^2 x - \sin x, & |x| > 1 \end{cases}, a = -2, b = 4, h = 0,5$$

$$14) y = \begin{cases} \sqrt[4]{(x+2)^3 - \ln x}, & x > 2 \\ 3^{-x}, & x \leq 2 \end{cases}, a = 1, b = 4, h = 0,3$$

$$15) y = \begin{cases} (x^5 + \sqrt{x}), & x > 0 \\ 5^{2-x}, & x \leq 0 \end{cases}, a = -2, b = 2, h = 0,4$$

$$16) y = (x + 1)^{21} - 21^{x+1}, a = -2, b = 1, h = 0,3$$

4.14. Вывести на экран наибольшее натуральное значение n , удовлетворяющее условию:

$$1) 0,8 e^n - 120 n < 12,4$$

$$2) -2,4 n + 12\sqrt{n} + 3,41 \geq 0$$

$$3) 9 n^5 - 81,4 n^3 < 0$$

$$4) 3 n^2 - 45 \ln n \leq 7,5$$

$$5) -n^3 + 2,5 n^2 - n e^n > 15$$

$$6) -e^n + 2n - n^3 < 1000$$

$$7) n^5 - e^{n-1} \geq -2$$

$$8) \operatorname{tg} \frac{\pi n}{2(n+1)} \leq 100$$

$$9) 2^{n-1} - \ln(n^3 + n^2) < 30$$

$$10) \sqrt[5]{n^3 + n^2} - \ln n < 20$$

4.15. В первый тренировочный день спортсмен пробежал 8 км. Каждый следующий день он увеличивал дневную норму на 10%. Определить, какой путь он пробежит в день с номером s и суммарный путь за все s дней.

4.16. Определить количество знаков в заданном натуральном числе.

4.17. Определить количество знаков после запятой в десятичной записи действительного числа.

4.18. Вводится последовательность целых чисел, ввод заканчивается, когда вводится ноль. Вывести на экран:

- 1) количество введенных чисел;
- 2) сумму всех чисел;
- 3) первое из чисел, кратных 5;
- 4) среднее арифметическое всех отрицательных чисел;
- 5) количество чисел, принадлежащих промежутку $(-4; 16)$;
- 6) количество чисел, которые по модулю меньше первого введенного числа;
- 7) среднее арифметическое всех чисел;
- 8) все числа, которые больше полусуммы первых двух;
- 9) номера всех чисел, которые отличаются от первого не более, чем на 2;
- 10) номера и сумму всех чисел, совпадающих с первым.

4.19. Написать программу для вычисления суммы сходящегося ряда с заданной точностью ε .

$$1) \sum_{i=1}^{\infty} \frac{i+1}{i^3 - i^2 + 1}$$

$$2) \sum_{i=1}^{\infty} \frac{\sin i}{i^2 + i}$$

$$3) \sum_{i=1}^{\infty} \frac{\ln(i^2 + 2)}{i^2 + 1}$$

$$4) \sum_{i=1}^{\infty} \frac{\sqrt{i-1}}{i^3 + 1}$$

$$5) \sum_{i=1}^{\infty} \frac{1}{\sqrt{i^5 + i^2}}$$

$$6) \sum_{i=1}^{\infty} \frac{i^2 - 1}{i^5 + i^3 + 1}$$

$$7) \sum_{i=1}^{\infty} (i+1)^{-2} \cdot \sin 3^{i+2}$$

$$8) \sum_{i=1}^{\infty} (\sqrt{i} + 3)^{-4} \cdot |\cos 2^i|$$

$$9) \sum_{i=1}^{\infty} (i+1)^2 \cdot 2^{-i}$$

$$10) \sum_{i=1}^{\infty} i^3 \cdot 3^{-2i}$$

$$11) \sum_{i=1}^{\infty} \frac{|\sin^5 i|}{\sqrt[3]{i^7 + i}}$$

$$12) \sum_{i=1}^{\infty} \frac{\sin i}{i^2 + i}$$

$$13) \sum_{i=1}^{\infty} \frac{\ln(1 + |\cos i|)}{(i+1)^2 + 4}$$

$$14) \sum_{i=1}^{\infty} \frac{\ln(1 + |\ln i|)}{(i+2)^2}$$

$$15) \sum_{i=1}^{\infty} \frac{|\sin^3(i+2)|}{\sqrt[4]{i^8 + 2}}$$

$$16) \sum_{i=1}^{\infty} 5^{-i^2+2}$$

$$17) \sum_{i=1}^{\infty} \frac{\sqrt{|\cos i|}}{e^{i-1}}$$

$$18) \sum_{i=1}^{\infty} \frac{\sqrt{|1 - \cos i|}}{e^{2i^2-i}}$$

$$19) \sum_{i=2}^{\infty} (i^2 + 4)^{-4} \cdot \ln|i - i^2| \quad 20) \sum_{i=1}^{\infty} \sqrt[3]{\sin(i^2 + 10)} \cdot e^{7-2i}$$

4.20. Найти минимальное значение n , при котором сумма расходящегося ряда превосходит заданное число A .

$$\begin{array}{ll} 1) \sum_{n=1}^{\infty} \frac{2}{n+1}, A = 4 & 2) \sum_{n=1}^{\infty} \frac{n^2 + 1}{\sin n + 2}, A = 20 \\ 3) \sum_{n=1}^{\infty} \frac{\ln|n - n^2|}{n + 12}, A = 12 & 4) \sum_{n=1}^{\infty} \sqrt[6]{|n^{-3} - n^3|}, A = 15 \\ 5) \sum_{n=1}^{\infty} \frac{\sqrt[4]{n^2 + 2n}}{\ln\left(n + \frac{1}{n}\right)}, A = 22 & 6) \sum_{n=1}^{\infty} \ln\left(n^2 - \frac{1}{3n}\right), A = 30 \\ 7) \sum_{n=1}^{\infty} \sqrt[4]{\left|\ln n + \frac{1}{n}\right|}, A = 25 & 8) \sum_{n=1}^{\infty} \frac{n}{|\sin n|}, A = 18 \\ 9) \sum_{n=1}^{\infty} \sqrt{\ln n + \frac{n}{n+1}}, A = 9 & 10) \sum_{n=1}^{\infty} \ln \frac{\sqrt{n}}{n^{-2} + n}, A = 27 \end{array}$$

4.21. Задана числовая последовательность формулой n -го члена. Вывести на экран количество и сумму тех членов последовательности, которые по модулю больше заданного числа ε .

$$\begin{array}{ll} 1) a_n = \frac{(-1)^{n-1}}{n^n} & 2) a_n = 2^{-n} + 3^{-n+1} \\ 3) a_n = \frac{n^2+1}{2^n} & 4) a_n = \frac{1}{(3n-2)(3n+1)} \\ 5) a_n = \frac{2^n}{n!} & 6) a_n = 1 - \frac{n}{n-1} \\ 7) a_n = \frac{(n+1)!}{n^n} & 8) a_n = 2^{-n} + n^{-1} \\ 9) a_n = \frac{n^2+2n}{n!} & 10) a_n = \frac{1}{(3n+1)!} \end{array}$$

4.22. Для числовых последовательностей найти наименьший номер n , для которого выполняется неравенство $|a_n - a_{n-1}| < \varepsilon$ для заданного значения ε . Вывести на экран ε и n .

$$\begin{array}{ll} 1) a_n = 2 + \frac{1}{a_{n-1}}, a_1 = 2 & 2) a_n = \frac{1}{2^{n+1}} \\ 3) a_n = \frac{1}{3} \operatorname{tg} a_{n-1}, a_1 = 0,5 & 4) a_n = \frac{1}{2} \cos a_{n-1}, a_1 = 0,5 \\ 5) a_n = \frac{a_{n-1} + a_{n-1}}{2}, a_1 = 1, a_2 = 2 & 6) a_n = 1 + \frac{1}{a_{n-1} + a_{n-2}}, a_1 = 0, a_2 = 2 \end{array}$$

7) $a_n = 1 + \sin \frac{1}{n}$

8) $a_n = \frac{n}{n+4}$

9) $a_n = 3 - 2^{1-n}$

10) $a_n = \sqrt{\frac{n}{2n+1}}$

5. Массивы

Пример 5.1. Дана последовательность целых чисел a_1, a_2, \dots, a_m . Вывести на экран все числа, для которых выполнено условие $a_i \geq i$, и их количество.

Решение

```

• program ex51;
  const m = 20;      // длина последовательности чисел
  begin
    println('Введите', m, 'целых чисел');
    var k := 0; { переменная для искомой суммы, тип определяется
                  по инициализации — integer }
    var a: array[1..m] of integer;      // описание массива
    for var i := 1 to m do              // начало цикла
      begin
        readln(a[i]);                  // заполнение массива
        if a[i] >= i then              // проверка условия
          begin
            k += 1; // если условие выполнено, сумма увеличивается на 1,
            println('i =', i, a[i])    // номер элемента и его значение
            // выводятся на экран
          end;
        end;
      writeln('k = ', k:4)              // вывод суммы на экран
    end. •

```

Пример 5.2. Дан одномерный массив целых чисел. Если в нем содержится не более 4 чисел, кратных 5, заменить эти числа единицами и вывести на экран новый массив, в противном случае — вывести на экран исходный массив.

Решение

```

• program ex52;
  const m = 20;
  begin
    var k := 0;      // переменная для подсчета чисел, кратных 5
    println('Введите', m, 'целых чисел');
    var a: array[1..m] of integer; // описание исходного массива
    //далее — ввод массива и определение количества элементов, кратных 5
    for var i := 1 to m do
      begin
        read(a[i]);
        if a[i] mod 5 = 0 then k += 1;
      end;
    end;

```

```

    {если k <= 4, значения элементов массива a передаются
      в массив b, массив a остается без изменений}
  if k <= 4 then
    begin
      var b := a;    // тип массива b определяется по присваиванию
      for var i := 1 to m do // замена элементов в массиве b
        begin
          if b[i] mod 5 = 0 then b[i] := 1;
          print(b[i]);
        end;
      end
    else for var i := 1 to m do print(a[i])
  end. •

```

Пример 5.3. Дан одномерный массив символов. Если он содержит нули или круглые скобки, заменить их вопросительными знаками и вывести на экран новый массив, иначе — вывести на экран восклицательный знак.

Решение

```

• program ex53;
  const m = 20;
  begin
    var p := 0; // начальное значение вспомогательной переменной p
    println('Введите', m, 'символов');
    var a: array[1..m] of char; // описание массива
    for var i := 1 to m do
      begin
        read(a[i]); // ввод элементов массива
        if (a[i] = '0') or (a[i] = '(') or (a[i] = ')') then
          {если массив содержит элементы «0», «(», «)», они заменяются
            на «?», а p изменяет значение на 1}
          begin
            a[i] := '?';
            p := 1
          end;
        end;
      { если переменная p изменила значение на 1, то массив был изменен,
        и он выводится на экран, если нет — массив не менялся }
    if p = 1 then for var i := 1 to m do print(a[i])
    else writeln('!')
  end. •

```

Пример 5.4. Дан двумерный массив действительных чисел. Вычислить среднее геометрическое значение элементов каждой строки и вывести результат на экран в виде нового массива. (Средним геометрическим N чисел называется корень степени N из модуля произведения всех чисел)

Решение

```
• program ex54;
  const m = 5;           // количество строк
      n = 4;           // количество столбцов
begin
  println('Введите', m, '*', n, 'действительных чисел');
  var a: array[1..m, 1..n] of real; // исходный 2-мерный массив
  var b: array[1..m] of real;      // новый 1-мерный массив
  for var i := 1 to m do          // ввод 2-мерного массива
    for var k := 1 to n do readln(a[i, k]);
  for var i := 1 to m do
    begin
      // вычисление произведения элементов i-й строки
      var h := 1.0;           // определение и инициализация переменной
                              // для среднего геометрического
      for var k := 1 to n do h *= a[i, k];
      {вычисление среднего геометрического и передача значений
       в массив b}
      b[i] := abs(h)**(1/n)
    end;
    // вывод нового одномерного массива
  for var i := 1 to m do writeln('b[', i, ']' = ', b[i]:5:2)
end. •
```

5.1. Дан одномерный массив целых чисел. Вывести на экран

- 1) количество четных чисел в массиве;
- 2) количество нулей;
- 3) количество чисел, кратных 3 и их сумму;
- 4) все числа, которые при делении на 7 дают остаток 3 или 4 и их количество;
- 5) все отрицательные числа;
- 6) все нечетные положительные числа, кратные 5 и их сумму;
- 7) сумму всех чисел, которые больше своего порядкового номера;
- 8) сумму всех чисел, которые принадлежат промежутку (-2; 8);
- 9) количество чисел, которые больше среднего арифметического всех элементов массива;
- 10) значения максимального и минимального элементов.

5.2. Дан одномерный массив целых чисел. Заменить в нем все числа, кратные 5, нулями и вывести на экран новый массив.

5.3. Дан одномерный массив целых чисел. Если в нем содержится более 3 отрицательных чисел, вывести на экран сообщение «Да», иначе — сообщение «Нет».

5.4. Дан одномерный массив действительных чисел. Если в нем содержится не более 3 отрицательных чисел, вывести на экран сообщение «Да», иначе — сообщение «Нет».

- 5.5. Дан одномерный массив A действительных чисел. Вывести на экран:
- 1) количество положительных чисел,
 - 2) сумму отрицательных чисел,
 - 3) количество нулей;
 - 4) наибольшее число;
 - 5) среднее арифметическое всех чисел;
 - 6) среднее геометрическое всех чисел;
 - 7) все числа, которые больше своего порядкового номера;
 - 8) все числа, которые меньше среднего арифметического;
 - 9) все числа, которые отличаются от минимального элемента в массиве не более, чем на 4
 - 10) наибольшее из отрицательных чисел и наименьшее из положительных.
- 5.6. Дан одномерный массив действительных чисел.
- 1) Переписать его элементы в обратном порядке в новый массив.
 - 2) Изменить порядок элементов на противоположный, не используя дополнительный массив.
- 5.7. Дан одномерный массив действительных чисел. Если в нем содержится более 3 отрицательных чисел, заменить отрицательные числа нулями и вывести на экран новый массив, в противном случае — вывести на экран исходный массив.
- 5.8. В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных и нечетных местах (первый элемент со вторым, третий с четвертым и т. д.)
- 5.9. При поступлении в вуз абитуриенты, получившие два балла на первом экзамене, ко второму экзамену не допускаются. В массиве $A[n]$ записаны оценки экзаменуемых, полученные на первом экзамене. Определить, сколько человек не допущено ко второму экзамену.
- 5.10. Дана последовательность чисел, содержащая один ноль. Вывести на экран все числа до нуля включительно.
- 5.11. В первой половине одномерного массива размещены значения аргумента, а во второй — соответствующие им значения функции в той же последовательности. Вывести на экран элементы этого массива в виде двух столбцов: аргумент и соответствующее значение функции.
- 5.12. Последовательность действительных чисел записана в виде массива. Определить, сколько раз изменяется знак в этой последовательности и вывести на экран номера позиций, в которых происходит смена знака.
- 5.13. Определить количество членов последовательности натуральных чисел, кратных числу M и заключенных в интервале от L до N .
- 5.14. Определить, сколько процентов от всего количества членов заданной последовательности целых чисел составляют нечетные числа.
- 5.15. Написать программу вычисления скалярного произведения двух векторов.
- 5.16. Дан одномерный массив действительных чисел. Найти сумму максимального и минимального элемента.
- 5.17. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Вывести на экран все элементы, принадлежащие отрезку $[c; d]$.

5.18. Последовательность $a_1 a_2, \dots, a_n$ состоит из нулей и единиц. Поставить в первую часть этой последовательности все нули, а во вторую — все единицы.

5.19. Дана последовательность целых чисел $a_1 a_2, \dots, a_n$. Создать новую последовательность, удалив из заданной все члены, равные минимальному значению.

5.20. Дан одномерный массив из N элементов. Сформировать из него два массива: первый — из элементов исходного массива с четными номерами, а второй — из элементов с нечетными номерами.

5.21. Дана последовательность целых чисел $a_1 a_2, \dots, a_n$. Вывести на экран все пары чисел, сумма которых равна заданному числу m .

5.22. Дан одномерный массив натуральных чисел. Вывести на экран все числа, которые оканчиваются на заданную цифру k .

5.23. Дана последовательность натуральных чисел. Найти сумму тех элементов последовательности, которые больше заданного числа M . Если таких чисел нет, вывести на экран соответствующее сообщение.

5.24. Дан одномерный массив натуральных чисел. Вывести на экран

- 1) все трехзначные числа, содержащиеся в массиве;
- 2) номера всех четырехзначных чисел;
- 3) все двухзначные числа, у которых сумма цифр равна заданному числу;
- 4) все трехзначные числа, в записи которых есть ноль;
- 5) все четырехзначные числа, у которых сумма двух первых цифр равна сумме двух последних;
- 6) первое из четырехзначных чисел, у которых первая и последняя цифра одинаковые;
- 7) первое из чисел, у которых последняя цифра 3 или 9;
- 8) наименьшее из чисел, начинающихся с двойки;
- 9) число с максимальной суммой цифр;
- 10) все числа, кратные 3, в порядке возрастания.

5.25. Дан одномерный массив символов. Вывести на экран количество восклицательных знаков в нем.

5.26. Дан одномерный массив символов. Если он содержит цифры, вывести их на экран в том же порядке, в каком они расположены в массиве, если цифр нет — вывести сообщение «Нет».

5.27. Дан одномерный массив символов. Если он содержит знаки арифметических операций, заменить их вопросительными знаками и вывести на экран новый массив, иначе — вывести на экран восклицательный знак.

5.28. Дан одномерный массив символов. Все символы, которые повторяются три раза подряд, вывести на экран, если таких символов нет, вывести на экран ноль.

5.29. Дан одномерный массив символов. Вывести на экран сообщение «Да», если выполнено условие и сообщение «Нет» в противном случае:

- 1) массив содержит удвоенные символы;
- 2) массив содержит идущие подряд запятую и тире;
- 3) массив содержит последовательности из трех и более цифр;
- 4) первый и последний символы одинаковые;

- 5) на четных местах не стоят пробелы;
- 6) первый и последний символы являются цифрами;
- 7) количество знаков препинания больше, чем количество цифр;
- 8) сумма всех цифр, содержащихся в массиве, меньше размера массива;
- 9) восклицательный и вопросительный знаки находятся только на нечетных местах;
- 10) буква «b» встречается чаще, чем буква «f».

5.30. Вывести на экран квадратную матрицу порядка n , сформированную по следующему образцу.

1)	1	2	3	...	n
	2	4	6	...	$2n$
	3	6	9	...	$3n$
	⋮	...
	n	$2n$	$3n$...	n^2
2)	1	2	3	...	n
	2	3	4	...	$n + 1$
	3	4	5	...	$n + 2$
	⋮	...
	n	$n + 1$	$n + 2$...	$2n$
3)	1	1	1	...	1
	2	2	2	...	2
	3	3	3	...	3
	⋮	...
	n	n	n	...	n
4)	1	2	3	...	n
	0	1	2	...	$n - 1$
	0	0	1	...	$n - 2$
	⋮	...
	0	0	0	...	1

5)

1	1	1	...	1
0	2	0	...	0
0	0	3	...	0
...	⋮	...
0	0	0	...	n

6)

1	2	3	...	n - 1	n
2	1	0	...	0	0
3	0	1	...	0	0
...	⋮	0	...
n - 1	0	0	...	1	0
n	0	0	...	0	1

7)

n	0	0	...	0	0
n - 1	n	0	...	0	0
n - 2	n - 1	n	...	0	0
...	⋮
2	3	4	n - 1	n	0
1	2	3	n - 2	n - 1	n

5.31. Дан двумерный массив целых чисел. Вывести на экран новый массив, элементами которого являются квадратные корни из элементов исходного массива.

5.32. Дан двумерный массив целых чисел. Заменить все его отрицательные элементы нулями и вывести на экран новый массив.

5.33. Дан двумерный массив действительных чисел. Вычислить среднее арифметическое элементов каждой строки и вывести результат на экран в виде нового массива.

5.34. Дан двумерный массив действительных чисел. Вычислить сумму элементов каждого столбца и вывести результат на экран в виде нового массива.

5.35. Дан двумерный массив действительных чисел. Посчитать в нем количество положительных чисел, количество отрицательных чисел и количество нулей и результаты вывести на экран.

5.36. Дан двумерный массив целых чисел. Заменить все элементы первой строки нулями и вывести на экран новый массив.

5.37. Дан двумерный массив целых чисел. Заменить все его отрицательные элементы значением -1 и вывести на экран новый массив.

5.38. Дана матрица A действительных чисел размера $m \times n$ и вектор b , содержащий n элементов. Написать программу для вычисления произведения $A \cdot b$.

5.39. Даны две квадратные матрицы действительных чисел одинакового размера. Написать программу для получения:

- 1) суммы матриц;
- 2) произведения матриц;
- 3) транспонированной матрицы (поменять местами строки и столбцы);
- 4) суммы транспонированных матриц;
- 5) произведения транспонированной первой матрицы на вторую исходную.

5.40. Дана прямоугольная матрица целых чисел A . Если она содержит отрицательные элементы, заменить их абсолютными величинами и вывести на экран новую матрицу. Если отрицательных элементов нет, вывести на экран матрицу такого же размера, как A , все элементы которой равны 1.

5.41. Дана вещественная прямоугольная матрица. Определить в ней максимальный элемент и переставить строки и столбцы так, чтобы максимальный элемент оказался в левом верхнем углу.

5.42. Дана целочисленная квадратная матрица. В каждой строке найти максимальный элемент и поменять его местами с элементом главной диагонали.

5.43. Для квадратной матрицы символов построить 2 одномерных массива: первый — из элементов главной диагонали, второй — из их кодов.

5.44. В прямоугольной матрице целых чисел найти максимальный и минимальный элементы и поменять их местами.

5.45. Дана прямоугольная матрица. Найти ее строки с наибольшей и наименьшей суммой элементов. Вывести на экран эти строки и суммы их элементов.

5.46. Определить наименьший элемент каждого нечетного столбца целочисленной квадратной матрицы и вывести на экран сам элемент и номер строки, в которой он находится.

5.47. Для целочисленной прямоугольной матрицы определить количество строк, которые содержат заданное число k и его квадрат.

5.48. Дана квадратная матрица действительных чисел. Вывести на экран:

- 1) сумму элементов, которые находятся на пересечении четных строк и столбцов с номерами, кратными 3;
- 2) все элементы, у которых сумма номера строки и номера столбца больше 5;

- 3) строку, содержащую максимальный элемент, в которой все элементы, стоящие на четных местах, заменены максимальным;
- 4) строки, в которых заданное число x встречается не более 1 раза;
- 5) строки, в которых знаки чередуются.

6. Записи, множества

Пример 6.1. Создать список студентов с указанием фамилии, имени, года рождения и номера группы. Вывести на экран список студентов, у которых фамилия начинается на букву «Д», с указанием года рождения.

Решение

```

• program ex61;
  const m = 20;
  type
    student = record           // определение типа запись
      fam: string[25];        // поле Фамилия
      name: string[20];       // поле Имя
      db: integer;            // поле Год рождения
      gn: integer;            // поле Номер группы
    end;

  begin
    var a: array[1..m] of student; // описание массива записей
    writeln('Введите список студентов');
    for var i:= 1 to m do // заполнение массива
      begin
        a[i].fam := ReadLnString('Введите фамилию');
        a[i].name := ReadLnString('Введите имя');
        a[i].db := ReadLnInteger('Введите дату рождения');
        a[i].gn := ReadLnInteger('Введите номер группы');
      end;
    var j := 1;
    repeat // поиск в массиве записей
      if copy(a[j].fam, 0, 1) = 'Д' then
        begin
          println(a[j].fam, a[j].db);
          inc(j)
        end
      until j > m;
    end. •
  
```

Пример 6.2. Даны два множества, содержащие по 8 случайных целых чисел из промежутка $(a; b)$, где a и b — действительные числа. Определить, сколько в этих множествах одинаковых элементов.

Решение

```

• program ex62;
  
```

```

const p = 8;           //количество элементов в множествах
begin
  randomize;          // подключение генератора случайных чисел
  var (a, b) := ReadlnReal2('Введите 2 действительных числа');
  if a > b then swap(a, b); { левая граница промежутка должна
  быть меньше правой, если это не так, меняем их местами }

  {преобразуем действительные границы промежутка a и b в целые m и n}
  var m := trunc(a);
  var n := trunc(b) + 1;
  {заполнение множеств s1, s2 случайными целыми
  числами из отрезка [m; n]}
  var s1, s2: set of integer; // описание множеств s1 и s2
  s1 := [];           // инициализация множеств s1, s2
  s2 := [];           // пустыми множествами
  for var i := 1 to p do
  begin               // заполнение множеств s1, s2 случайными целыми
    s1 += [random(n, m)]; // числами из промежутка (n; m)
    s2 += [random(n, m)];
  end;
  foreach var k in s1 do print(k); // вывод на экран s1
  println;
  foreach var k in s2 do print(k); // вывод на экран s2
  println;
  {определение количества целых чисел из отрезка [m; n],
  одновременно принадлежащих множествам s1 и s2}
  var k := 0;        // счетчик количества одинаковых элементов
  for var i := m to n do // подсчет одинаковых элементов
    if (i in s1) and (i in s2) then inc(k);
  println('k =', k);
end. •

```

Замечание. Все элементы, входящие в множество, различны, поэтому если генератор случайных чисел даст два одинаковых числа, в множество будет включено только одно из них. Это означает, что множества s1, s2 вообще говоря, содержат не 8 целых чисел, а не более 8 целых чисел.

6.1. Дать описание переменной типа запись с 4 полями для записи фамилии, имени, года рождения, адреса. Полю Фамилия присвоить значение «Иванов», полю Год рождения — значение 1997, остальные поля оставить без изменений.

6.2. Дать описание переменной типа запись с 3 полями для записи фамилии, места рождения, возраста. Полю Возраст присвоить значение «25», остальные поля оставить без изменений.

6.3. Дать описание переменной типа запись с 4 полями для записи фамилии, имени, номера группы, оценки за экзамен. Полю Номер группы присвоить значение «5», остальные поля оставить без изменений.

6.4. Дать описание переменной типа запись с 4 полями для записи фамилии, имени, номера квартиры, номера телефона. Полю Номер телефона присвоить значение «123–321», остальные поля оставить без изменений.

6.5. Дать описание переменной типа запись с 4 полями для записи курса, номера группы, фамилии, оценки за экзамен. Полю Оценка за экзамен присвоить значение «4», остальные поля оставить без изменений.

6.6. Создать список сотрудников из n строк (число n ввести с клавиатуры), содержащий следующую информацию: фамилия, место рождения, возраст. Вывести на экран фамилии сотрудников, возраст которых меньше 30 лет.

6.7. Создать список студентов из n строк (число n ввести с клавиатуры), содержащий следующую информацию: фамилия, курс, номер группы, оценка за экзамен. Вывести на экран фамилии студентов 3 группы 1 курса.

6.8. Создать список жильцов из n строк (число n ввести с клавиатуры), содержащий следующую информацию: фамилия, имя, номер квартиры, номер телефона. Вывести на экран список жильцов, проживающих в квартирах с номерами от 21 до 60.

6.9. Разработать программу для ведения журнала успеваемости группы студентов, в котором содержится информация о фамилии, номере группы и оценках за 3 экзамена. Вывести на экран фамилии студентов в порядке убывания среднего балла по результатам трех экзаменов.

6.10. Разработать программу для хранения следующей информации об абитуриентах: фамилия и имя, год рождения, результат ЕГЭ по математике, суммарный балл ЕГЭ (по трем предметам)

Выполнить следующие задания:

- 1) определить и вывести на экран количество абитуриентов 1997 и 1998 годов рождения;
- 2) вывести на экран фамилии абитуриентов, у которых результат ЕГЭ по математике выше 85 баллов, а суммарный балл выше 250;
- 3) вывести на экран фамилию и год рождения абитуриентов, у которых суммарный балл ЕГЭ выше 270
 - а) не используя оператор присоединения;
 - б) с помощью оператора присоединения;
- 4) вывести на экран фамилии абитуриентов в порядке убывания их суммарного балла по ЕГЭ, если суммарные баллы совпадают, сравнить баллы по математике.

6.11. Переменная `abc4` имеет тип данных запись. Дайте возможные описания переменной `abc4` (2–3 варианта), если в исполняемой части кода использован следующий оператор:

```
with abc4 do  
begin  
  n1 := 'steel plane';  
  n2 := 2010;  
  n3 := false;  
  n4 := '12–45';  
  n5 := '+'  
end;
```

6.12. Переменная *xu* имеет тип данных запись. Дайте возможные описания переменной *xu*, если в исполняемой части кода использованы следующие операторы:

```
...  
xu.k1:= 1225;  
xu.k2:= '1225';  
xu.k3:= b > 0;  
xu.k4:= 'yes';  
...
```

6.13. Переменная *sr3* имеет тип данных запись. Дайте возможные описания переменной *sr3* (2–3 варианта), если в исполняемой части кода использован следующий оператор:

```
with sr3 do  
begin  
m1:= '6';  
m2:= x*sin(x);  
m3:= x <= 1;  
m4:= round(x);  
m5:= 'finish'  
end;
```

6.14. В таблице содержится информация о результатах сессии (фамилия, имя, номер группы, результаты четырех экзаменов). Назначить студентам стипендии, используя следующие правила:

- а) если все оценки 5, назначается повышенная стипендия;
- б) если все оценки не ниже 4, назначается обычная стипендия;
- в) если есть оценка 3, стипендия не назначается.

6.15. В таблице содержится следующая информация об учениках: фамилия, имя, возраст, рост, вес. Выполнить следующие задания:

- 1) определить средний рост учеников;
- 2) определить рост самого высокого и самого низкого учеников;
- 3) вывести на экран фамилии учеников, у которых рост выше среднего;
- 4) определить количество учеников, у которых рост выше 170 см.

6.16. *N* учеников прошли тестирование по *M* темам предмета. Определить:

- 1) сколько баллов набрал каждый ученик по всем темам;
- 2) средний балл, полученный учениками;
- 3) разницу между лучшим результатом и средним баллом.

6.17. Описать переменную «экзаменационная ведомость», содержащую следующие сведения: наименование предмета, номер группы, номера зачетных книжек, фамилии и инициалы студентов, а также их оценки по итогам текущей сессии. Определить, сколько в группе отличников, хорошистов, троечников и двоечников.

6.18. Описать переменную «служащий», включающую в себя фамилию, имя, отчество служащего, дату его рождения, сведения об образовании, профессии и домашний адрес.

- 1) Определить имена служащих с высшим образованием.
- 2) Получить данные о служащих, имеющих указанную профессию.

6.19. Дан массив записей, содержащих дату (число, месяц, год) и время (час, минута, секунда). Упорядочить этот массив данных в порядке возрастания, т. е. от более ранних значений к более поздним.

6.20. Описать переменную «расписание», содержащую следующие данные: день недели, количество учебных пар в указанный день, время начала и конца каждой пары, название предмета, фамилию преподавателя. Вывести всю информацию о занятиях, относящихся к предмету «Информатика».

6.21. Множество состоит из 5 случайных целых чисел в диапазоне от 1 до 20. Вывести их на экран.

6.22. Создать множество, содержащее точно 10 случайных чисел из диапазона (0, 100) и вывести его на экран.

6.23. Создать два множества, содержащих по 10 случайных целых чисел из диапазона 1..20. Вывести на экран те числа, которые входят в оба множества.

6.24. Создать множество, которое содержит случайный набор строчных букв русского алфавита и

- 1) вывести на экран все гласные буквы, содержащиеся в этом множестве,
- 2) проверить, можно ли из букв, содержащихся в множестве, составить слово «факел», если нет, то добавить в множество необходимые буквы.

6.25. Множество содержит случайный набор строчных букв русского алфавита. Проверить,

- 1) все ли гласные буквы в нем содержатся,
- 2) можно ли из этих букв составить слово «привет».

6.26. Множество A содержит арабские цифры. Проверить, все ли цифры от 0 до 9 входят в множество и вывести соответствующее сообщение.

6.27. Дан некоторый набор товаров. Определить, какие товары имеются в каждом из n магазинов, какие есть хотя бы в одном магазине и каких нет ни в одном из магазинов.

6.28. Множество содержит натуральные числа из некоторого диапазона. Сформировать из него два множества, первое из которых содержит все числа, кратные 3 или 7, а второе — все остальные числа.

6.29. Множество содержит натуральные числа из некоторого диапазона. Сформировать из него два множества, первое из которых содержит все простые числа, а второе — все составные числа.

6.30. Дан двумерный массив действительных чисел. Вычислить сумму элементов, номера строк и столбцов которых принадлежат соответственно непустым множествам S_1 и S_2 .

6.31. Из диапазона целых чисел $m..n$ выделить множество чисел, делящихся без остатка или на k , или на p (k, p — простые числа), и множество чисел, делящихся на произведение kp .

6.32. Создать два множества из символов, которые вводятся пользователем с клавиатуры. Вывести на экран элементы, которые содержатся

- 1) в первом и втором множествах одновременно;
- 2) только в первом множестве;
- 3) только во втором множестве.

6.33. Множество состоит из n целых случайных чисел из отрезка $[a; b]$ (a и b — действительные числа). Выполнить следующие задания:

- 1) вывести на экран все элементы множества, являющиеся нечетными положительными числами;
- 2) определить, сколько в данном множестве четных чисел из отрезка $[c; d]$;
- 3) определить, сколько всего элементов содержится в множестве;
- 4) определить, сколько целых чисел из отрезка $[a; b]$ не попало в множество;
- 5) разбить полученное множество на два: множество, содержащее числа, в записи которых есть ноль, и множество, содержащее все остальные числа.

6.34. Дан текст из цифр и строчных латинских букв, который заканчивается точкой. Определить, каких букв, гласных (а, е, и, о, у) или согласных, больше в этом тексте.

6.35. Определить количество различных цифр в десятичной записи натурального числа.

6.36. Вывести на экран все цифры, не входящие в запись заданного натурального числа,

- 1) в порядке возрастания;
- 2) в порядке убывания.

7. Обработка символов и строк

Пример 7.1. Дана последовательность символов, состоящая из слов (под словом понимается любая последовательность символов между двумя соседними пробелами). Определить количество слов, начинающихся с буквы b .

Решение

```
• program ex71;
  begin
    var str1 := ReadString('Введите текст');
    var m := 0; // счетчик
    if str1[1] = 'b' then inc(m); // проверка 1-го символа в строке
    // проверка остальных символов
    for var k := 1 to length(str1) - 1 do
      if (str1[k] = ' ') and (str1[k+1] = 'b') then inc(m);
    println('m =', m);
  end. •
```

Пример 7.2. Дана строка, содержащая открывающую и закрывающую круглые скобки. Удалить текст, расположенный между скобками, вместе со скобками и вывести новую строку на экран.

Решение

```
• program ex72;
  begin
    var str := ReadString('Введите текст');
```

```
// определение номеров позиций открывающей и закрывающей скобок
var m := pos('(', str);
var n := pos(')', str);
delete(str, m, n - m + 1); // удаление текста между скобками
writeln(str)
end. •
```

7.1. Дана строка. Определить и вывести на экран количество знаков препинания в ней (точка, запятая, точка с запятой, вопросительный и восклицательный знаки, тире, двоеточие).

7.2. Дана строка. Посчитать и вывести на экран количество гласных букв в ней.

7.3. Дана строка. Написать программу, которая определяет, является ли строка

- 1) целым числом (содержит только цифры),
- 2) действительным числом (содержит только цифры и десятичную точку).

Вывести на экран соответствующее сообщение.

7.4. Дана строка. Написать программу, которая определяет количество удвоенных согласных и выводит его на экран, а также сами согласные по одному разу.

7.5. Ввести строку, которая заканчивается точкой. Определить, сколько пробелов в строке и удалить их. Вывести на экран количество пробелов и новую строку.

7.6. Ввести строку, которая заканчивается точкой. Определить, содержит ли строка цифры и знаки арифметических операций и вывести их на экран; если их в строке нет, вывести соответствующее сообщение.

7.7. Дана строка, состоящая из строчных английских букв и цифр 0..9, не содержащая пробелов. Вывести на экран подряд все цифры, встречающиеся в строке.

7.8. Дана строка, содержащая открывающую и закрывающую круглые скобки.

- 1) Вывести на экран все символы, расположенные между скобками.
- 2) Удалить все символы, расположенные между скобками:
 - а) оставив скобки на месте;
 - б) вместе со скобками.

7.9. Дана строка, состоящая из строчных английских букв без пробелов. Удалить все символы, расположенные между первым и вторым вхождением буквы 'a'. Вывести на экран новую строку.

|| *Словом в тексте считается любая последовательность символов, не содержащая пробелов, отделенная пробелами от других символов.*

7.10. Дана последовательность символов, состоящая из слов. Удалить из текста все слова, начинающиеся с буквы a. Вывести на экран полученную стро-

ку. Если в тексте слова с указанным свойством отсутствуют, вывести сообщение «Нет».

7.11. Дана строка, состоящая из слов. Определить и вывести на экран:

- 1) количество слов;
- 2) количество символов в самом коротком и самом длинном слове;
- 3) количество слов, содержащих одновременно буквы «у» и «е»;
- 4) количество слов, начинающихся с гласной буквы;
- 5) количество символов в первом и последнем слове.

7.12. Дана строка, состоящая из слов. Вывести на экран все слова:

- 1) начинающиеся с буквы «с»;
- 2) состоящие из четырех символов;
- 3) не содержащие гласных букв;
- 4) состоящие более чем из трех символов;
- 5) содержащие цифры.

7.13. Дана последовательность символов, состоящая из букв английского алфавита без пробелов. Удалить все буквы, которые встречаются более 1 раза. Вывести на экран полученную строку.

7.14. Дана последовательность символов, состоящая из слов. Если существует буква, входящая во все слова, вывести на экран сообщение «Да», в противном случае — сообщение «Нет».

7.15. Дана строка, состоящая из строчных английских букв без пробелов. Оставить в строке только первое вхождение каждого символа и вывести полученную строку на экран.

7.16. Дана строка, состоящая из строчных английских букв без пробелов. После каждой буквы 'а' вставить вторую такую же и вывести новую строку на экран.

7.17. Дана строка, которая содержит комментарии, расположенные между символами «(*)» и «*)». Удалить все комментарии и вывести на экран новую строку.

7.18. Дана последовательность символов, состоящая из слов. Определить содержит ли строка слова, состоящие исключительно из цифр, если да, то

- 1) вывести эти слова на экран;
- 2) удалить их из строки и вывести на экран новую строку;
- 3) заменить все такие слова словом «число» и вывести на экран новую строку.

7.19. Строка содержит латинские буквы и цифры и состоит из слов, разделенных пробелами. Выполнить следующие задания:

- 1) определить количество слов, состоящих исключительно из букв, и вывести их на экран;
- 2) определить количество разных цифр, встречающихся в строке;
- 3) вывести на экран все слова, содержащие только цифры;
- 4) определить количество слов, содержащих сочетание букв adf;
- 5) вывести на экран все слова, содержащие удвоенные символы;
- 6) определить количество слов, содержащих хотя бы одну букву «z».

7.20. В заданной строке заменить все знаки препинания (, , — : ? !) на точку с запятой. Определить и вывести на экран количество замен, а также новую строку.

7.21. Строка содержит латинские буквы, цифры и открывающие и закрывающие скобки. Определить, одинаковое ли количество открывающих и закрывающих скобок содержится в строке. Рассмотреть 3 случая:

- 1) строка содержит только круглые скобки;
- 2) строка содержит круглые и квадратные скобки;
- 3) строка содержит круглые, квадратные и фигурные скобки.

7.22. Текст состоит из слов, разделенных одним или несколькими пробелами. Отредактировать текст, удалив лишние пробелы (удалить пробелы в начале и в конце, если они есть, и между словами оставить по одному пробелу).

7.23. Дан текст из строчных латинских букв, который заканчивается точкой. Вывести на экран

- 1) все буквы, входящие в этот текст не менее двух раз;
- 2) все буквы, входящие в этот текст по одному разу.

7.24. Дана строка из строчных русских букв. Вывести на экран в алфавитном порядке все гласные буквы (а, е, и, о, у, ы, э, ю, я), входящие в эту строку.

7.25. Задан текст, состоящий из слов. Определить:

- 1) какие цифры содержатся в этом тексте;
- 2) с каких букв (русских) начинаются слова в тексте;
- 3) содержит ли текст другие символы, кроме латинских букв и цифр;
- 4) какие латинские буквы есть в каждом слове этого текста;
- 5) какие буквы встречаются только в одном слове этого текста.

Результаты вывести на экран.

7.26. В строке содержится запись арифметического выражения. Определить:

- 1) какие арифметические операции использованы в выражении;
- 2) какие цифры есть в выражении;
- 3) каких цифр нет в выражении;
- 4) есть ли в записи этого выражения скобки (круглые).

Результаты вывести на экран.

8. Файлы

Пример 8.1. В файловую переменную записать 20 действительных чисел, которые последовательно вводятся с клавиатуры.

Решение

```
• program ex81;
  const m = 20;
  begin
    var fx1: file of real; // описание файла
    assign(fx1, 'd:\f1'); // связывание с файловой переменной
    rewrite(fx1); // открытие файла для записи
    for var i := 1 to m do
      begin
        var x := ReadReal('Введите действительное число');
        write(fx1, x); // запись в файл
      end;
    end. •
```

Пример 8.2. В переменную *b* записать элемент файла действительных чисел Rnum.dat с номером *n*.

Решение

```
• program ex82;
  begin
    var fr: file of real;
        // инициализация файловой переменной
    assign(fr, 'd:\f1');
    var n := ReadInteger('Введите номер элемента');
    reset(fr);      // открытие файла для чтения и записи
    var b: real;
        // чтение из файла в переменную b
    for var i := 1 to n do read(fr, b);
    write(b);
    close(fr)      // закрытие файла
  end. •
```

Пример 8.3. Найти сумму всех чисел из файла действительных чисел d:\f1.

Решение

```
• program ex83;
  begin
    var f1: file of real;
    assign(f1, 'd:\f1');
    var sr := 0.0;    // переменная для суммы
    reset(f1);
    var b: real;
    repeat           // вычисление суммы
      read(f1, b);
      sr += b
    until Eof(f1);
    write(sr:7:2)
  end. •
```

Пример 8.4. Определить количество строк в текстовом файле mail.txt.

Решение

```
• program ex84;
  begin
    var mail2: text;
    assign(mail2, 'D:\mail.txt');
    reset(mail2);
    var n := 0;
    while not Eof(mail2) do
      begin
        readln(mail2);
        inc(n)
      end
    end
  end
```

```
end;  
println('Количество строк n =', n);  
close(mail2);  
end. •
```

8.1. Создайте файл целых чисел. Числа вводятся с клавиатуры. Признак конца ввода — число 101.

8.2. Задайте 2 файла целых чисел и определите, в каком из них больше чисел.

8.3. Задать файл целых чисел и определить, сколько отрицательных чисел и нулей он содержит.

8.4. Заполнить файл f последовательного доступа целыми числами, полученными с помощью генератора случайных чисел. Записать в новый файл g компоненты файла f , являющиеся четными числами.

8.5. Записать в файл последовательного доступа N действительных чисел. Вычислить произведение компонентов файла и вывести их на экран.

8.6. Заполнить файл последовательного доступа случайными целыми числами из диапазона $[m; n]$. Выполнить следующие задания:

- 1) переписать в новый файл все компоненты исходного файла, кратные p , которые не делятся на q ;
- 2) определить количество пар противоположных чисел среди компонентов этого файла;
- 3) исключить все повторные вхождения чисел и вывести новый файл на экран;
- 4) найти сумму минимального и максимального элементов;
- 5) переписать в новый файл все числа отличные от нуля.

8.7. Задан файл целых чисел. Определить, являются ли числа упорядоченными по возрастанию.

8.8. Дан файл целых чисел. Переписать в новый файл те из них, которые принадлежат отрезку $[a; b]$. Значения a и b вводятся с клавиатуры.

8.9. Дан файл целых чисел. Записать в новый файл действительный числа, являющиеся средним арифметическим каждых пяти идущих подряд чисел первого файла (последняя пятерка может быть неполной).

8.10. Заполнить файл случайными натуральными числами. Выполнить следующие задания:

- 1) найти количество чисел, являющихся квадратами нечетных чисел;
- 2) найти наибольшее из чисел, кратных 7;
- 3) переписать в новый файл суммы пар соседних чисел;
- 4) вывести на экран разность первого и последнего чисел.

8.11. Даны два файла целых чисел. Объединить их в один и упорядочить по убыванию.

8.12. Матрица A размером 40×40 записана в файл по строкам. Определите, является ли она единичной (элементы главной диагонали равны 1, остальные нули).

8.13. Дан текстовый файл, состоящий из строк. Определите и выведите на экран количество строк в файле.

- 8.14. Дан текстовый файл, состоящий из строк. Определите наибольшую длину строки и выведите строку максимальной длины на экран.
- 8.15. Дан текстовый файл, состоящий из строк. Выведите на экран строки, которые имеют наибольшую и наименьшую длину.
- 8.16. Дан текстовый файл. Определите и выведите на экран
- 1) количество цифр, которые в нем содержатся;
 - 2) количество различных цифр;
 - 3) количество слов (словом считается последовательность символов, не содержащая пробелов).
- 8.17. Дан текстовый файл. Определите и выведите на экран все целые числа, которые в нем содержатся (целым числом считается последовательность цифр, ограниченная с обеих сторон пробелами).
- 8.18. Дан текстовый файл. Определите и выведите на экран все действительные числа, которые в нем содержатся (действительным числом считается последовательность цифр, содержащая не более одной точки, которая занимает не первую и не последнюю позицию, ограниченная с обеих сторон пробелами).
- 8.19. Дан текстовый файл, состоящий из русских букв и цифр. Определить, чего в нем больше: букв или цифр.
- 8.20. Дан текстовый файл, состоящий из слов. Определить:
- 1) сколько знаков содержит самое длинное слово;
 - 2) сколько слов максимальной длины содержит текст.
- 8.21. Дан файл, содержащий текст на русском языке и два слова. Определить:
- 1) сколько раз эти два слова входят в текст;
 - 2) сколько раз эти два слова располагаются непосредственно друг за другом.
- 8.22. Дан файл, содержащий текст на русском языке. Выбрать из него буквы, встречающиеся только один раз в порядке их расположения.
- 8.23. Дан файл, содержащий текст, включающий русские и английские слова. Определить, каких букв в тексте больше: русских или латинских.
- 8.24. Дан файл, содержащий текст. Определить, сколько слов в этом тексте и сколько цифр.
- 8.25. Дан файл Bibl, содержащий сведения о книгах. Сведения о каждой из книг включают в себя фамилию автора, название книги и год издания. Определить:
- 1) названия книг указанного автора, изданных с 1960 года по настоящее время;
 - 2) имеется ли книга с названием «Информатика». Если да, то вывести на экран фамилию автора и год издания. Если таких книг несколько, вывести все имеющиеся сведения обо всех этих книгах.
- 8.26. Дан файл T, содержащий номера телефонов сотрудников фирмы с указанием их фамилий и инициалов. Найти номер телефона сотрудника по его фамилии и инициалам.
- 8.27. Файл RT содержит различные даты, включающие число, месяц и год. Найти:
- a) год с наименьшим номером;

- б) все весенние даты;
- в) самую позднюю дату.

8.28. Файл Assort содержит сведения об игрушках: название игрушки, ее стоимость в рублях и возрастные границы (например, игрушка может предназначаться для детей от двух до пяти лет). Определить:

- 1) названия игрушек, цена которых не превышает 5 000 руб. и которые подходят детям пяти лет;
- 2) стоимость самого дорогого конструктора;
- 3) названия наиболее дорогих игрушек (цена которых отличается от цены самой дорогой игрушки не более чем на 3 000 руб.);
- 4) названия игрушек, которые подходят детям и четырех, и десяти лет;
- 5) можно ли подобрать игрушку (любую, кроме мяча), подходящую ребенку трех лет;
- 6) название самой дешевой игрушки;
- 7) название самой дорогой игрушки для детей до четырех лет;
- 8) названия игрушек для детей от четырех до пяти лет;
- 9) название самой дорогой игрушки, подходящей детям от двух до трех лет;
- 10) стоимость самой дорогой куклы;
- 11) стоимость кукол для детей шести лет;
- 12) для детей какого возраста предназначается конструктор;
- 13) для детей какого возраста предназначены кубики, указать их среднюю стоимость.

8.29. Файл TV содержит сведения о программах телепередач на неделю: день недели, время, канал, вид и название телепередачи. Определить:

- 1) названия телепередач, которые идут в указанный день в указанный промежуток времени;
- 2) названия телепередач, которые идут в указанный день на указанном канале;
- 3) информацию об указанном фильме; если этот фильм не указан в телепрограмме, вывести на экран сообщение «Такой передачи на этой неделе нет»;
- 4) на каком канале и в какое время будет транслироваться развлекательная передача «Поле чудес»;
- 5) названия телепередач, идущих в указанное время на разных каналах;
- 6) название самой продолжительной передачи в понедельник;
- 7) название передачи, завершающей эфир в каждый день недели.

8.30. Файл содержит сведения о безработных: специальность, опыт работы, образование, пол, возраст. Требуется:

- 1) подобрать кандидатов на должность врача с опытом работы не менее пяти лет;
- 2) найти работников с высшим экономическим образованием не старше 35 лет;
- 3) найти работников, имеющих опыт работы в сфере торговли;

- 4) получить полную информацию обо всех женщинах в возрасте от 20 до 40 лет;
- 5) определить средний возраст всех мужчин, ищущих работу;
- 6) выяснить, кого в базе данных больше с высшим образованием: женщин или мужчин;
- 7) найти n самых молодых работников.

9. Процедуры и функции

Пример 9.1. По приведенной части заголовка подпрограммы определить ее вид (процедура или функция) и заполнить пропуски, если это необходимо.

... ex91(x, y: real): ...

Решение

После имени подпрограммы стоит двоеточие, следовательно, вначале пропущено слово `function`, и после двоеточия должен быть указан тип возвращаемого функцией результата.

Пример 9.2. Составить программу для вычисления площади кольца. Вычисление площади круга оформить в виде функции.

Решение

```

• program ex92;
  function krug(r: real): real; // описание функции
  begin
    krug:= pi*r*r
  end;

begin
  var (r1, r2) := ReadlnReal2('Введите
                             значения внутреннего и внешнего радиусов');
  if r1 > r2 then swap(r1, r2); { r1 — внутренний радиус,
                               должно выполняться условие r1 < r2 }
  var s := krug(r2) - krug(r1); // вычисление площади кольца
  writeln('s = ', s:6:2)
end. •

```

Пример 9.3. Решим задачу из примера 9.2, заменив функцию `krug` на процедуру с тем же именем.

```

• program ex93
var a: real;
procedure krug(r: real);
begin
  a := pi*r*r
end;
begin
  var (r1, r2) := ReadlnReal2('Введите значения внутреннего и
                             внешнего радиусов');

```

```

krug(r1);
var s1 := a;      // вычисление площади внутреннего круга
krug(r2);
var s2 := a;      // вычисление площади внешнего круга
writeln('s = ', (s2 - s1):6:2)
end. •

```

Рекурсия — способ организации вычислительного процесса, при котором подпрограмма в ходе выполнения составляющих ее операторов обращается сама к себе.

Пример 9.4. Вычислить сумму элементов одномерного массива, используя рекурсивную функцию. Массив заполнить случайными числами.

```

• program ex94;
  const m = 50;      // количество элементов в массиве
  type
  mass = array[1..m] of integer; // определение типа массива
  function summa(n: byte; r: mass): integer;
  begin
    if n = 0 then summa:= 0
    else summa := r[n] + summa(n - 1, r) // использование
                                         // рекурсии
  end;

begin
  randomize;
  var a: mass;
      // заполнение массива случайными числами
  for var i := 1 to m do a[i]:= random(200);
  println('s = ', summa(m, a))
end. •

```

9.1. Вычислить значение выражения

$$z(x) = (\text{sign}(x) + \text{sign}(y)) \cdot \text{sign}(x + y).$$

Определить и использовать при решении функцию sign:

$$\text{sign}(x) = \begin{cases} -1, & x < 0 \\ 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

9.2. Вычислить значение выражения

$$Z(x) = \text{sh}^2(x) + \text{ch}(x)(\text{ch}^2(x) + \text{sh}(x^2)),$$

где $\text{sh}(x)$ и $\text{ch}(x)$ задаются формулами

$$\text{sh}(x) = \frac{e^x - e^{-x}}{2}; \quad \text{ch}(x) = \frac{e^x + e^{-x}}{2}.$$

Вычисление $\text{sh}(x)$ и $\text{ch}(x)$ оформить в виде функций.

9.3. Разработать функцию, которая вычисляет значение наибольшего элемента одномерного массива действительных чисел с количеством элементов $n \leq 100$. Количество элементов в массиве должно задаваться при обращении к функции.

9.4. Составить программу, которая определяет минимальный элемент в каждой строке заданного двумерного массива целых чисел и записывает его в новый одномерный массив. Вычисление минимального элемента оформить в виде функции.

9.5. Разработать функцию, которая в заданной строке находит натуральное число (последовательность цифр между двумя пробелами) и определяет его значение. Если в строке целого числа нет, значение функции равно -1 .

9.6. Дан одномерный массив строк. Записать в числовой массив количество знаков препинания в каждой строке, причем для определения этого количества использовать функцию.

9.7. Дан одномерный массив строк, состоящих из цифр и букв латинского алфавита. Записать в числовой массив номера строк, которые содержат не более двух цифр. Для определения количества цифр использовать функцию.

9.8. Элементами массива являются строки. Написать процедуру, которая записывает в числовой массив длины строк и выводит этот массив на экран.

9.9. Двухмерный массив состоит из целых чисел в диапазоне 0..255.

1) Написать процедуру, которая заменяет эти числа соответствующими им символами.

2) Решить обратную задачу.

9.10. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Для каждого элемента a_i определить количество элементов слева, которые по модулю меньше a_i , и количество элементов справа, которые по модулю больше a_i . Использовать подпрограмму, которая производит вычисления для каждого элемента последовательности.

9.11. Разработать процедуру, которая в квадратной матрице целых чисел

1) все элементы главной диагонали заменяет единицами;

2) заменяет все элементы, лежащие ниже главной диагонали, нулями;

3) заменяет каждый элемент главной диагонали суммой всех элементов столбца и строки, на пересечении которых он находится.

9.12. Дана строка, состоящая из слов (под словом понимается последовательность символов между двумя пробелами).

1) Разработать функцию, которая проверяет, содержит ли данное слово цифры.

2) Разработать процедуру, которая удаляет из строки все слова, не содержащие цифр.

3) Написать код, который использует эти подпрограммы и выводит на экран получившуюся в результате строку.

9.13. Написать процедуру, которая генерирует единичную матрицу заданного размера (на главной диагонали единицы, остальные элементы — нули).

9.14. Вычислить площадь правильного шестиугольника со стороной a , используя подпрограмму вычисления площади треугольника.

9.15. На плоскости заданы координаты k точек. Определить пару точек, расстояние между которыми самое большое.

Указания. Координаты точек рассматривать как двумерный массив. Для вычисления расстояния между точками использовать функцию.

9.16. Написать рекурсивную функцию для вычисления факториала ($n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$).

9.17. Написать рекурсивную функцию для вычисления заданного числа элементов последовательности, в которой a_1 и a_2 единицы, а каждый элемент, начиная с третьего, равен разности двух предыдущих.

9.18. Написать рекурсивную функцию для вычисления заданного числа элементов последовательности, в которой a_1 и a_2 единицы, каждый нечетный элемент равен удвоенному предыдущему, а каждый четный — сумме двух предыдущих.

10. Графика

Пример 10.1. Установить в центре экрана графическое окно размером 400×300 пикселей зеленого цвета с красными точками, расположенными на расстоянии 5 пикселей одна от другой.

Решение

Откроем консольное приложение командой

Файл → Новый проект → Консольное приложение

```
• program ex101;
  uses GraphABC; //подключение графического модуля
  begin
    SetWindowTitle('Графическое окно'); // заголовок окна
    SetWindowWidth(400); // установка ширины и
    SetWindowHeight(300); // высоты окна
    Window.CenterOnScreen; // установка окна по центру экрана
    Window.Clear(color.Green); // цвет фона
    // установка точек в окне
    for var x := 1 to ScreenWidth do
      for var y := 1 to ScreenHeight do
        PutPixel(5*x, 5*y, Color.Red);
      readln;
    end. •
```

Пример 10.2. Построить график функции $y = 2 \cos x$ на промежутке $\left[-\frac{\pi}{2}; \frac{3\pi}{2}\right]$.

Решение

Откройте консольное приложение.

```
• program ex101;
  uses GraphABC;
  begin
    SetPenColor(Color.Red); // цвет линий красный
    Window.Clear(Color.Aqua); // цвет фона голубой
    Line(10, 120, 400, 120); // ось x
```

```

Line(10, 20, 10, 250);           // ось y
    // построение графика синими точками
var x := -pi/2;
for var i := 1 to 400 do
begin
    var xg := 10 + round(95/pi*x);
    var yg := 120 - round(100*cos(x));
    PutPixel(xg, yg, color.Blue);
    x += pi/95
end;
    // разметка осей, надписи
TextOut(15, 10, 'Y');
TextOut(400, 120, 'X');
TextOut(130, 40, 'Y = 2 cos x');
readln;
end. •

```

10.1. Нарисовать окружность радиуса 50 пикселей красного цвета с центром в середине экрана.

10.2. Нарисовать окружность радиуса 80 пикселей, которая касается верхней и левой границ экрана.

10.3. Нарисовать окружность радиуса 70 пикселей, которая касается нижней и правой границ экрана.

10.4. Нарисовать полуокружность, центр которой находится в середине нижней границы экрана.

10.5. Нарисовать прямоугольник со сторонами 80 и 120 пикселей, верхний левый угол которого совпадает с верхним левым углом экрана.

10.6. Нарисовать прямоугольник со сторонами 80 и 120 пикселей, верхний правый угол которого находится в точке с координатами (100; 40)

10.7. Нарисовать прямоугольный треугольник с катетами, длиной 50 и 100 пикселей, параллельными координатным осям, у которого вершина прямого угла совпадает с правым верхним углом экрана.

10.8. Нарисовать отрезок, концы которого расположены в случайных точках.

10.9. Нарисовать зеленый треугольник на желтом фоне с вершинами в точках с координатами (10; 5), (20; 20), (15; 30).

10.10. Нарисовать 3 отрезка, цвет которых выбирается случайным образом и концы расположены в случайных точках.

10.11. Нарисовать треугольник зеленого цвета на сером фоне с вершинами в точках (0; 0), (20; 10), (20; 40) и задать его движение

- 1) по горизонтали;
- 2) по вертикали;
- 3) случайным образом.

10.12. Используя только процедуру PutPixel, верхнюю половину экрана разлиновать горизонтальными линиями, а нижнюю — вертикальными

10.13. Составить программу «Звездное небо», которая на черном фоне

- 1) случайным образом рисует белые точки;
- 2) случайным образом рисует точки случайного цвета;

3) рисует случайные точки случайного цвета, а уже светящиеся убирает (закрашивает цветом фона).

10.14. Используя графические примитивы, нарисовать дом и раскрасить его.

10.15. Построить графики функций на заданных промежутках:

1) $y = \ln(|\sin x| + 1)$, $x \in [0; 5]$;

2) $y = \frac{x-1}{x^2+x+1}$, $x \in [2; 8]$;

3) $y = 3^{x-x^2}$, $x \in [-3; 3]$;

4) $y = |\cos^3(x + 2)|$, $x \in [-4; 2]$;

5) $y = \operatorname{arctg} \sqrt{x^{25} + 1}$, $x \in [-2; 2]$.

10.16. Нарисовать на экране шахматную доску.

10.17. Нарисовать горизонтальный отрезок, движущийся от левой границы экрана к правой.

10.18. Нарисовать вертикальный отрезок, движущийся от левой границы экрана к правой.

10.19. Используя только процедуру Line, нарисовать закрашенный прямоугольник, движущийся

1) от левой границы экрана к правой;

2) от верхней границы экрана к нижней.

10.20. Нарисовать мигающую (появляющуюся и исчезающую) окружность в центре экрана. Предусмотреть ввод во время работы программы радиуса окружности и периода мигания (времени между появлением и исчезновением).

III. Комплексные задания по разным темам

Вариант 01

Изучите программный код и выполните задания.

```
begin  
  var n := readinteger;  
  var k := 1;  
  while n div 10 <> 0 do  
    begin  
      n := n div 10;  
      k += 1  
    end;  
  println('k =', k)  
end.
```

1. В программном коде

- 1) вводится действительное число n и определяется его знак
- 2) вводится символьная переменная n и определяется ее код
- 3) вводится натуральное число n и определяется сумма его цифр
- 4) вводится целое число n и определяется количество цифр в нем

2. Перечислите переменные, которые использованы в коде, и назовите их типы

3. Если в коде используются выражения логического типа, назовите их.

4. Укажите операторы, которые использованы в коде

- | | |
|--------------------------|----------------------------------|
| 1) оператор выбора | 2) оператор цикла с параметром |
| 3) оператор присваивания | 4) оператор цикла с постусловием |
| 5) условный оператор | 6) оператор цикла с предусловием |

5. Перепишите программный код, заменив оператор while..do оператором repeat..until.

Вариант 02

Изучите программный код и выполните задания.

```
begin  
  var (a, b, c) := Read...;  
  var rs1 := (a + b > c) and (a + c > b) and (b + c > a);  
  var rs2 := (a = b) or (b = c) or (a = c);  
  if not rs1 then writeln('нет') else  
    if rs2 then writeln('...')  
  end.  
end.
```

1. Фрагмент программного кода выполняет следующие действия

- 1) определяет, существует ли треугольник со сторонами a, b, c
- 2) вычисляет площадь треугольника со сторонами a, b, c
- 3) вычисляет четвертую сторону описанного четырехугольника, три стороны которого равны a, b, c
- 4) определяет, является ли треугольник прямоугольным

- 5) определяет, является ли треугольник равнобедренным
 - 6) определяет, является ли треугольник тупоугольным или остроугольным
 - 7) вычисляет площадь равнобедренной трапеции с основаниями a и b и боковой стороной c
2. Выберите возможные продолжения, которые могут быть использованы в операторе присваивания во 2-й строке (вместо многоточия)
- | | | | |
|------------|-------------|----------|-----------|
| 1) Real | 2) Integer3 | 3) Byte | 4) Char |
| 5) Boolean | 6) Char3 | 7) Real3 | 8) String |
3. Определите тип переменных $rs1$ и $rs2$.
4. В процедуре вывода в предпоследней строке вместо точек следует вписать текст
- | | |
|---------------------|------------------------------|
| 1) равносторонний | 2) прямоугольный |
| 3) тупоугольный | 4) остроугольный |
| 5) равнобедренный | 6) треугольник не существует |
| 7) неравносторонний | 8) треугольник существует |
5. Можно ли заменить условные операторы, использованные в коде, оператором выбора? Ответ обоснуйте.

Вариант 03

Изучите программный код и выполните задания.

begin

var (a, b) := Read...;

var h := Read...;

var x := a;

var y := 0.0;

repeat

if $x \leq 1$ **then** $y := x*x - 1$

else $y := \ln(x)$;

writeln('x = ', x:6:2, ' y = ', y:8:2);

x += h

until $x > b$;

end.

1. Фрагмент программного кода реализует следующие действия:
 - 1) вычисляет корень уравнения с заданной точностью
 - 2) выводит на экран таблицу значений функции
 - 3) вычисляет величины членов последовательности, не превосходящих заданного значения
 - 4) выводит на экран значения последовательных приближений суммы ряда
 - 5) выводит на экран значения членов последовательности, которые принадлежат заданному промежутку
2. Запишите формулу, которая используется в этом фрагменте, в обычной математической форме.
3. Оформите вычисление этой формулы в виде отдельной функции
4. Определите тип переменных h , a , b , x , y

5. Чем можно заменить оператор присваивания $x += h$? Укажите все возможности.

6. Замените оператор `repeat..until` оператором `while..do`

Вариант.04

Изучите программный код и выполните задания.

begin

```
var x := Read...;
```

```
var k: ...;
```

```
if x <= 1 then k := 0;
```

```
if (x > 1) and (x <= 5) then k := 1;
```

```
if x > 5 then k := 2;
```

```
var y: ...;
```

```
case k of
```

```
1: y := sin(sqrt(x));
```

```
2: y := exp(2/x);
```

```
3: y := sqrt(x*x - ln(x))
```

```
end;
```

```
Writeln ('x = ', x:5:2, ' y = ', y:6:2);
```

end.

1. Фрагмент программы выполняет следующие действия:

1) выводит на экран значения x , не превосходящие 1

2) вычисляет значение числовой функции и выводит его на экран

3) определяет, принадлежит ли значение x промежутку $(1; 5]$

4) вычисляет значение логической функции y и выводит его на экран

5) вычисляет значения числовых функций x , y и k

2. Замените 3 условных оператора (строки 4–6) отдельной функцией или объясните, почему этого сделать нельзя.

3. Как следует изменить код, чтобы исключить переменную k ?

4. Запишите формулу, которая использована в приведенном коде, в обычной математической форме.

5. Определите тип переменных k , x , y ;

Вариант 05

Изучите программный код и выполните задания.

```
const n = 20;
```

```
begin
```

```
var a: array[1..n, 1..n] of integer;
```

```
var b: ...;
```

```
for var i := 1 to n do
```

```
  for var j := 1 to n do
```

```
    a[i, j] := random(300);
```

```
for var i := 1 to n do
```

```
  begin
```

```
    var R := a[i, 1];
```

```
    for var j := 2 to n do
```

```

*   if a[i, j] > R then R := a[i, j];
      b[i] := R
    end;
  for var i := 1 to n do print(b[i]);
end.

```

1. Код выполняет действия
 - 1) заполняет одномерный массив случайными числами
 - 2) 1-й элемент каждой строки записывает в одномерный массив и выводит этот массив на экран
 - 3) заполняет случайными числами квадратную матрицу
 - 4) определяет наименьший элемент в каждой строке матрицы
 - 5) определяет наибольший элемент в каждой строке матрицы, записывает в новый массив и выводит его на экран
 - 6) определяет наибольший элемент в каждом столбце матрицы, записывает его номер в новый массив и выводит его на экран
 - 7) заносит в новый массив среднее арифметическое элементов каждой строки матрицы
2. Оформите условие $a[i, j] > R$ в виде функции и покажите, как изменится строка кода, отмеченная звездочкой, если использовать эту функцию.
3. Определите тип переменной b .
4. Выберите все верные утверждения
 - 1) переменная a может иметь тип `array[1..n, 1..n] of char`
 - 2) переменная a может иметь тип `array[1..n, 1..n] of real`
 - 3) переменные a и b имеют один и тот же тип данных
 - 4) матрица a заполнена случайными числами
 - 5) условный оператор в строке, помеченной звездочкой, нельзя заменить оператором выбора
 - 6) массив b содержит n элементов
 - 7) массивы a и b имеют один и тот же базовый тип данных
5. Замените оператор цикла с параметром в предпоследней строке кода оператором цикла `foreach`.

Вариант 06

Изучите программный код и выполните задания.

```

begin
  var s: ...;
  for var i := 1 to 4 do
    for var j := 1 to 6 do
      begin
        read(s[i, j]);
        if s[i, j] = '+' then s[i, j] := '1';
      end;
    for var i := 1 to 4 do
      begin
        for var j := 1 to 6 do write(s[i, j]);

```

```
println
end;
end.
```

1. Фрагмент программы выполняет следующие действия:
 - 1) считывает числовой массив 4×6 по столбцам
 - 2) считывает символьный массив 4×6 по строкам
 - 3) считывает символьный массив 6×4 по столбцам
 - 4) считывает массив целых чисел 6×4 по строкам
 - 5) заменяет все элементы 1-й строки на единицы
 - 6) определяет количество знаков «плюс» и заменяет их на «1»
 - 7) заменяет все знаки «плюс» на единицы
 - 8) заменяет все единицы на знак «плюс»
 - 9) выводит на экран полученный массив в строку
 - 10) выводит на экран полученный массив в виде таблицы
2. Определите тип переменной s
3. Замените выражение логического типа логической переменной a. Как изменится код?
4. Что изменится, если процедуру write заменить на print?
5. Определите тип переменных i и j. Измените код так, чтобы переменные i и j получили тип byte.

Вариант 07

Изучите программный код и выполните задания.

```
begin
  var st := Read...;
  var k := 0;
  var st1 := '';
  for var i := 1 to length(st) - 1 do
    begin
      var a := st[i] = st[i+1];
      if a then
        begin
          inc(k);
          st1 += st[i]
        end
      end;
    println(st1);
    println('k =', k);
  end.
```

1. Программный код выполняет следующие действия
 - 1) считывает строку, удаляет из нее повторяющиеся символы и выводит на экран их количество и измененную строку
 - 2) считывает строку, повторяющиеся символы переносит в новую строку и выводит на экран новую строку и ее длину

- 3) считывает символьный массив, определяет код каждого символа, записывает в целочисленный массив и выводит на экран этот массив и его размер
- 4) считывает целочисленный массив, преобразует его в символьный, удаляет повторяющиеся символы и выводит на экран новый массив и его размерность
- 5) считывает строку, удаляет из нее повторяющиеся символы, кроме одного и выводит измененную строку

2. Установите соответствие

1	a	A	string
2	st	B	boolean
3	k	C	integer
4	st1		
5	i		

3. Выберите все операторы, используемые в программе

- 1) оператор выбора
- 2) оператор цикла с параметром
- 3) оператор присваивания
- 4) оператор цикла с постусловием
- 5) условный оператор
- 6) оператор безусловного перехода
- 7) оператор цикла с предусловием
- 8) оператор обращения к процедуре

4. Перепишите программный код, заменив оператор цикла for оператором while..do.

5. Исключите из программного кода переменную a.

6. Измените код так, чтобы переменные i и k имели тип smallint.

Вариант 08

Изучите программный код и выполните задания.

```

const n = 5;
begin
  var s: ...;
  for var i := 1 to n do readln(s[i]);
  var m := 0;
  var k := 1;
  repeat
    if s[k] in ['a'..'l'] then
      begin
        s[k] := '0';
        inc(m);
      end
    else
      k := k + 1;
  until k > n;
end;

```

```

    end;
    k += 1
until k > n;
println('m =', m);
foreach var a in s do print(a)
end.

```

1. Программный код выполняет следующие действия

- 1) считывает строку, определяет в ней количество m букв $a...l$, вставляет после каждой из них пробел и выводит на экран значение m и новую строку
- 2) считывает массив символов, определяет в нем количество m букв от a до l , выводит на экран значение m и элементы массива в столбец
- 3) считывает строку, определяет в ней количество букв $a...l$, удаляет эти буквы из строки и новую строку выводит на экран
- 4) считывает массив символов, определяет в нем количество m букв $a...l$, заменяет их нулями и выводит на экран значение m и элементы массива в строку
- 5) считывает строку, определяет в ней количество m букв $a...l$, заменяет их пробелами и выводит на экран значение m и новую строку

2. Определите тип переменных s , k , m и a .

3. Выпишите строку, содержащую конструктор множества, и измените ее так, чтобы множество не использовалось

4. Замените оператор цикла с постусловием оператором цикла с предусловием

5. Замените оператор `foreach`

- а) оператором `for`;
- б) оператором `repeat..until`.

6. Можно ли заменить оператор `foreach` оператором `loop`? Ответ обоснуйте.

Вариант 09

Изучите программный код и выполните задания.

```

begin
    var eps := Readln...;
    var a: ... := 1;
    var sr: ... := 0;
    var n: ... := 0;
    while a > eps do
        begin
            inc(n);
            a := ln(n*n + 1)/sqr(n + 1/n);
            sr += a;
        end;
        writeln('sr = ', sr:5:2, ' n = ', n:3);
    end.

```

1. Программный код выполняет следующие действия
 - 1) вычисляет сумму конечного числа элементов последовательности
 - 2) вычисляет корень уравнения с заданной точностью
 - 3) вычисляет сумму сходящегося ряда с заданной точностью
 - 4) определяет номер элемента числовой последовательности, начиная с которого все члены последовательности удовлетворяют условию $a_n < \varepsilon$
2. Определите тип переменных eps, sg, a, n. Может ли переменная sg иметь тип integer?
3. Запишите формулу, которая используется в программе, в обычной математической форме.
4. Замените оператор цикла while оператором repeat.
5. Можно ли заменить оператор while оператором for? оператором foreach? Ответ обоснуйте.

Вариант 10

Изучите программный код и выполните задания.

```

begin
  var s := Readln...;
  var k := 1;
  var t1 := '';
  var m := 0;
  repeat
    if (s[k] >= '0') and (s[k] <= '9') then
      begin
        inc(m);
        t1 += s[k];
      end;
    inc(k)
  until k > length(s);
  println(s);
  println('m =', m, t1)
end.

```

1. Программный код выполняет следующие действия
 - 1) ввод действительной переменной
 - 2) ввод строки
 - 3) вычисляет количество положительных элементов в массиве
 - 4) определяет, содержит ли строка знаки препинания
 - 5) определяет количество цифр в строке
 - 6) выводит цифры на экран в виде целого числа
 - 7) записывает цифры в новую строку
 - 8) выводит на экран новую строку
 - 9) выводит на экран количество цифр в строке
 - 10) выводит на экран новый массив, составленный из цифр, содержащихся в строке

2. Определите тип переменных s, t1, k, m. Может ли переменная m иметь тип boolean? тип byte?
 3. Замените оператор цикла repeat оператором while.
 4. Оформите условие, которое используется в первой строке оператора цикла, в виде отдельной функции.
 5. Можно ли оператор println(s) оформить с помощью оператора цикла?
- Ответ обоснуйте.
6. Каким образом можно изменить код, чтобы постусловие имело вид until k > N?

Вариант 11

Изучите программный код и выполните задания.

```
function f1(a, b: real): boolean;
begin
    if a*b <= 0 then f1 := true else f1 := false
end;
begin
    var (x, y) := ReadReal2;
    var z: ...;
    if f1(x, y) then z := 5*tan(x + y)
    else
        if x >= y then z := x else z := y*2;
    writeln(z)
end.
```

1. Функция f1 принимает значение false в случае, если переменные a и b
 - 1) не равны одновременно нулю;
 - 2) имеют одинаковые знаки;
 - 3) имеют разные знаки;
 - 4) имеют общие делители, отличные от 1;
 - 5) положительны;
 - 6) упорядочены по возрастанию.
2. Выберите все верные утверждения:
 - 1) в программе выполняется ввод переменных a и b;
 - 2) в программе определяется знак произведения переменных a и b;
 - 3) переменные a и b являются формальными параметрами.
 - 4) присваивание $z = 2y$ выполняется только в случае, если $x \cdot y > 0$ и $x < y$;
 - 5) в программе выполняется определение знака каждой из переменных x и y;
 - 6) в случаях, когда $(x > 0$ и $y < 0)$ и $(x < 0$ и $y > 0)$, переменная z принимает одно и то же значение;
3. Измените описание функции f1 так, чтобы в нем использовалась неполная форма условного оператора.
4. Определите, какое значение у должно быть введено, если известно, что:
 - 1) $x = -1, z = 0$ 2) $x = 0, z = 0$
5. Замените в основной части кода условный оператор оператором выбора.

6. Перепишите код без использования функции (программа должна выполнять те же действия).

Вариант 12

Изучите программный код и выполните задания.

```
const m = 4; n = 6;
begin
  var a: ...;
  for var i := 1 to m do
    begin
      for var j := 1 to n do
        begin
          a[i, j] := random(-20, 20);
          write(a[i, j]:4);
        end;
      writeln
    end;
  for var i := 1 to m do
    for var j := 1 to n do
      if i = j then a[i, j] := 1 else
        if a[i, j] >= i + j then a[i, j] := i + j;
    writeln;
  .....
end.
```

1. В программе выполняются следующие действия:
 - 1) заполнение квадратной матрицы случайными действительными числами;
 - 2) ввод двумерного символьного массива;
 - 3) заполнение массива размером $m \times n$ случайными целыми числами;
 - 4) изменение значений некоторых элементов массива;
 - 5) изменение значений всех элементов массива;
 - 6) замена элементов, у которых первый и второй индексы совпадают, единицами;
 - 7) замена элементов, у которых первый и второй индексы различны, нулями;
 - 8) сравнение значения элемента с суммой его индексов.
2. Определите тип переменной *a*.
3. Выберите все верные утверждения:
 - 1) переменная $a[i, j]$ может иметь тип `boolean`;
 - 2) переменная $a[i, j]$ может иметь тип `real`;
 - 3) в результате выполнения программы все элементы массива положительны;
 - 4) при $m = n$ все элементы главной диагонали равны 1;
 - 5) базовый тип элементов массива `integer`.

4. Вместо многоточия в конце программы добавьте процедуру вывода так, чтобы массив был выведен на экран

- 1) в один столбец,
- 2) в одну строку через пробел;
- 3) в виде прямоугольной таблицы.

5. Измените код таким образом, чтобы значения элементов исходного массива вводились с клавиатуры. Выполните ввод следующей матрицы:

```
5   7   1
3   -2  8
-6  4   0
```

Вариант 13

Изучите программный код и выполните задания.

```
begin
  var s1 := ...;
  var i := 0;
  repeat
    i += 1;
    if s1[i] in ['0'..'9'] then delete(s1, i, 1)
  until i >= length(s1);
  case length(s1) of
    0:   writeln(0);
    1..8: writeln(s1)
  end;
  writeln(length(s1))
end.
```

1. Код выполняет действия

- 1) ввод строки текста и определение количества цифр в ней
- 2) замена в текстовой строке всех цифр пробелами
- 3) ввод строки и удаление из нее всех символов, отличных от цифр
- 4) определение количества символов в строке, отличных от цифр
- 5) ввод строки текста и удаление из нее цифр

2. Переменная s1 может иметь тип:

- | | | | |
|-------------------------|------------|-----------------|---------------|
| 1) char | 2) integer | 3) record | 4) string[30] |
| 5) array[1..20] of char | 6) string | 7) file of text | |

3. Что появится на экране после ввода следующих данных:

- | | |
|--------------|-----------------------|
| 1) 1928 | 2) 12 + 10 = 22 |
| 3) 22 января | 4) сегодня 22 января |
| 5) 145,293 | 6) 135,7 — 143,12 = ? |

4. Замените оператор цикла с условием оператором цикла с параметром.

5. Замените оператор выбора условным оператором.

6. На самом деле этот код работает некорректно. Программа должна удалить все цифры из переменной s1. Найдите причину некорректной работы кода и устраните ее.

Ответы и комментарии

Вариант 01

1. Ответ 4
2. n, k -integer, тип n задается явно, тип k определяется по присваиванию
3. $n \text{ div } 10 <> 0$
4. Ответ 3, 6
5. ...

```
repeat
  n := n div 10;
  k += 1
until n div 10 = 0;
...
```

Вариант 02

1. Ответ 1, 5
2. Ответ 2, 7 (выполняется ввод 3 числовых значений, причем в соответствии с остальными операторами они могут быть как целыми, так и вещественными)
3. Boolean
4. Ответ 5
5. Можно, например, так:

```
case rs1 of
  true: writeln('нет');
  false: if rs2 then writeln('равнобедренный')
end;
```

Вариант 03

1. Ответ 2
 2. $y = \begin{cases} x^2 - 1, & x \leq 1 \\ \ln x, & x > 1 \end{cases}$
 3. function f1(x: real): real;
- begin
- if $x \leq 0$ then $y := x**2 - 1$ else $y := \ln(x)$
- end;
4. Все переменные имеют тип real, что следует из операторов присваивания, которые к ним применяются, и оператора вывода, где для x задается формат действительного числа

5. $x := x + h$

6. repeat

while $x \leq b$ do

begin

if $x \leq 1$ then $y := x*x - 1$

else $y := \ln(x)$;

writeln('x = ', x:6:2, ' y = ', y:8:2);

$x += h$

end;

Вариант 04

1. Ответ 2

2.

```
function f2(y: real): byte;
begin
  if y <= 1 then f2 := 0;
  if (y > 1) and (y <= 5) then f2 := 1;
  if y > 5 then f2 := 2;
end;
```

3. Заменить оператор выбора case вложенными условными операторами

$$4. \quad y = \begin{cases} \sin(x^2), & x \leq 1 \\ e^{\frac{2}{x}}, & 1 < x \leq 5 \\ \sqrt{x^2 - \ln x}, & x > 5 \end{cases}$$

5. x, y — real; k — byte, integer

Вариант 05

1. Ответ 3, 5

2. **function** f5(k, m: integer): boolean;

begin

 f5 := true;

 if k < m then f5 := false

end;

* if f5(a[i, j], R) then R := a[i, j];

3. array[1..n] of integer

4. Ответ 2, 4, 6

5. **foreach** var i in b do print(i);

Вариант 06

1. Ответ 2, 7, 10

2. **var** s: array[1..4, 1..6] of char;

3. Выражение логического типа s[i, j] = '+' заменим логической переменной var a := s[i, j] = '+', тогда соответствующий участок кода примет вид

...

read(s[i, j]);

var a := s[i, j] = '+';

if a **then** s[i, j] := '1';

...

4. В каждой строке массива элементы будут выводиться через пробел

5. Переменные i и j по присваиванию получают тип integer. Изменить его на byte можно так:

for var i: byte := 1 to 4 **do**

for var j: byte := 1 to 6 **do**

Вариант 07

1. Ответ 2
2. В А С А С
3. Ответ 2, 3, 5, 8
- 4–5.

```
while i <= length(st) - 1 do
begin
  if st[i] = st[i+1] then
  begin
    inc(k);
    st1 += st[i]
  end
end;
```

В приведенном фрагменте выполнена замена операторов цикла, а также исключение переменной *a*

6. См. задание 5 из варианта 06.

Вариант 08

1. Ответ 4
2. *s*: array[1..n] of char; (массив — следует из первого оператора цикла for, в котором выполняется заполнение массива, базовый тип элементов char — следует из присваиваний в операторе repeat);
k, *m*: integer; (автоопределение по присваиванию);
a: char; (следует из того, что тип переменной *a* в операторе foreach совпадает с типом элементов контейнера, т. е. в данном случае — символьного массива *s*).
3.

```
if s[k] in ['a'..'l'] then s[k] := '0';
```

заменяем на

```
if (s[k] >= 'a') and (s[k] <='l') then s[k] := '0';
```
4. См. задание 6, вариант 03.
5.

```
foreach var a in s do print(a)
```

 - a)

```
for var i := 1 to n do print(s[i]);
```
 - б)

```
var i := 1;
repeat
  print(s[i]);
  i += 1
until i > n;
```
6. Можно, например, так:

```
loop 1 do print(s);
```

Вариант 09

1. Ответ 3
2. *sr*, *a*: real; — следует из операторов присваивания;
n: integer; (или другой целый тип) — следует, например, из оператора inc(*n*);
тип переменной *eps* логично вывести из условия задачи: заданная точность по смыслу — сколь угодно малая величина, поэтому тип *eps* — real.

$$3. \sum \frac{\ln(n^2+1)}{\left(n+\frac{1}{n}\right)^2}$$

4. См. задание 6, вариант 03.

5. Нельзя и нельзя. Для оператора `for` должно быть заранее известно количество повторений цикла, а для оператора `foreach` — количество элементов в контейнере. По смыслу задачи количество повторений меняется в зависимости от заданной точности, а контейнер вообще определить невозможно.

Вариант 10

1. Ответ 2, 5, 7, 8, 9

2. `s, t1: string;`

`k, m: integer;`

переменная `m` может иметь тип `byte`, так как по смыслу это целое число, соответственно, тип `boolean` иметь не может.

3. `while k <= length(s) do`

`begin`

`if (s[k] >= '0') and (s[k] <= '9') then`

`begin`

`inc(m);`

`t1 += s[k];`

`inc(k)`

`end;`

`end;`

4. `function f10(x: char): boolean;`

`begin`

`if (x >= '0') and (x <= '9')`

`then f10 := true`

`else f10 := false`

`end;`

5. Если заменить оператор `println(s)` оператором цикла

`for var i := 1 to n do print(s[i]);`

то элементы строки будут выведены через пробел. Поэтому лучше использовать процедуру `write`:

`for var i := 1 to n do write(s[i]);`

Кроме того, после такого оператора цикла следует перевести курсор в новую строку процедурой `writeln` или `println` без параметров.

Замечание. Если в приведенных операторах цикла использовать вместо процедур `write` (`print`) процедуры `writeln` (`println`), элементы строки будут выведены в столбец.

6. Следует ввести дополнительную переменную `var N := length(s)`.

Вариант 11

1. Ответ 2.

2. Ответ 3, 4, 6

3. `function f1(a, b: real): boolean;`

`begin`

```

f1 := false;
if a*b <= 0 then f1 := true
end;
4. 1) y = 1, тогда xy < 0, z = 5 tg 0 = 0;
2) y = 0, тогда xy = 0, z = 5 tg 0 = 0.
5. case a*b <= 0 of
true: z := 5*tan(x + y);
false: if x >= y then z := x else z := y*2
end;
6. ...
if a*b <= 0 then z := 5*tan(x + y) else
if x >= y then z := x else z := y*2;
...

```

Вариант 12

```

1. Ответ 3, 4, 6, 8
2. var a: array[1..m, 1..n] of integer;
3. Ответ 2, 4, 5
4. 1) for var i := 1 to m do
for var j := 1 to n do println(a[i, j]);
2) for var i := 1 to m do
for var j := 1 to n do print(a[i, j]);
3) for var i := 1 to m do
begin
for var j := 1 to n do write(a[i, j]:4);
writeln
end;

```

Замечание. При выводе элементов массива в виде таблицы существенным является расположение элементов в столбцах (один под другим), поэтому в последнем случае выбрана процедура вывода write, которая позволяет задать ширину поля вывода (форматный вывод). Процедура print такой возможности не дает.

5. Первый оператор цикла for выполняет 2 действия: заполнение массива случайными целыми числами и вывод его на экран. Для того, чтобы показать на экране введенный массив в виде таблицы, эти 2 действия следует разделить:

```

for var i := 1 to m do
for var j := 1 to n do readln(a[i, j]);
for var i := 1 to m do
begin
for var j := 1 to n do write(a[i, j]:4);
writeln
end;

```

Вариант 13

```

1. Ответ 5
2. Ответ 4, 5, 6

```

3.1) 98 2;

2) 9

3) 2 января 8

4) 16

5) 4,9 3

6) 12

4. ...

```
for var i := 1 to length(s1) do
  if s1[i] in ['0'..'9'] then delete(s1, i, 1);
```

...

5. ...

```
if length(s1) = 0 then writeln(0)
else if (length(s1) >= 1) and (length(s1) <= 8)
  then writeln(s1);
```

...

6. Если строка содержит цифры, идущие подряд, то этот код не удаляет вторую цифру, потому что ее номер в строке уменьшается на единицу, а код переходит к следующей позиции, т. е. вторую цифру пропускает. Поэтому код удаляет не все цифры. Изменим код следующим образом:

...

```
repeat
  i += 1;
  if s1[i] in ['0'..'9'] then
    begin
      delete(s1, i, 1);
      i -= 1
    end
until i >= length(s1);
```

...

IV. Тестовые задания по темам

Алфавит, идентификаторы, операции

1. Зарезервированными словами в PascalABC.net являются

- 1) div 2) array 3) at 4) when
5) string 6) sum 7) unit 8) while

2. Зарезервированными словами в PascalABC.net являются

- 1) it 2) label 3) case 4) subject
5) end 6) foreach 7) some 8) go

3. Зарезервированными словами в PascalABC.net являются

- 1) set 2) kind 3) record 4) do
5) dim 6) loop 7) go 8) every

4. Зарезервированными словами в PascalABC.NET *не* являются

- 1) at 2) for 3) type 4) unit
5) some 6) while 7) sum 8) when

5. Зарезервированными словами в PascalABC.NET *не* являются

- 1) mod 2) at 3) until 4) when
5) function 6) loop 7) all 8) while

6. Операция целочисленного деления обозначается ____

7. Операция нахождения остатка от деления обозначается ____

8. Установить соответствие

1	**	A	определение остатка от деления
2	/	B	уменьшение на ...
3	mod	C	возведение в степень
4	*	D	целочисленное деление
5	-=	E	умножение
6	div	F	деление

9. Установить соответствие

1	<>	A	логическое «и»
2	+=	B	условие «не равно»
3	or	C	целочисленное деление
4	mod	D	увеличение на ...
5	and	E	логическое «или»
6	div	F	определение остатка от деления

10. Выберите все верные утверждения. В PascalABC.net идентификатор
- 1) может начинаться с цифры
 - 2) может содержать латинские буквы
 - 3) может начинаться со знака подчеркивания
 - 4) может содержать пробелы
 - 5) не может иметь длину более 40 символов
11. Выберите все верные утверждения. В PascalABC.net идентификатор
- 1) может содержать цифры в любой позиции, кроме первой
 - 2) не может быть именем типа данных
 - 3) может содержать не более одного пробела
 - 4) может быть именем функции
 - 5) может содержать скобки
12. Выберите все верные утверждения
- 1) Операция `not` является бинарной
 - 2) Описание переменных может находиться в разделе операторов
 - 3) Имя константы не может начинаться с цифры
 - 4) В PascalABC.net символ `**` обозначает возведение в степень
13. Выберите все верные утверждения
- 1) Резервированное слово нельзя использовать в качестве имени переменной
 - 2) Имя идентификатора не может начинаться со знака подчеркивания
 - 3) Порядок следования разделов описания переменных, констант, типов и меток в PascalABC.net строго определен
 - 4) Приоритет унарных операций выше, чем операций сравнения
14. Выберите все верные утверждения
- 1) Пару символов, обозначающих операцию «больше или равно» следует записывать через пробел
 - 2) Метки предназначены для использования в операторе безусловного перехода
 - 3) В PascalABC.net отсутствует операция возведения в степень
 - 4) В PascalABC.net допускается множественное присваивание значений переменным
15. Выберите все верные утверждения
- 1) Имя программы не может содержать пробелов
 - 2) В PascalABC.net отсутствует функция `tg x`
 - 3) В PascalABC.net константы можно описывать в разделе операторов
 - 4) Обращение к функции может быть операндом в выражении

Стандартные функции

1. К вещественным аргументам применимы функции

- 1) $\text{abs}(x)$ 2) $\text{pred}(x)$ 3) $\text{trunc}(x)$
4) $\text{exp}(x)$ 5) $\text{dec}(x)$ 6) $\text{int}(x)$

2. К вещественным аргументам неприменимы функции

- 1) $\tan(x)$ 2) $\text{succ}(x)$ 3) $\text{inc}(x)$
4) $\arcsin(x)$ 5) $\text{random}(x)$ 6) $\text{int}(x)$

3. Результат действия функции $\text{odd}(x)$ имеет тип

- 1) целый 2) символьный 3) логический 4) вещественный

4. Результат действия функции $\text{ord}(x)$ имеет тип

- 1) целый 2) символьный 3) логический 4) вещественный

5. Результат действия функции $\text{chr}(x)$ имеет тип

- 1) целый 2) символьный 3) логический 4) вещественный

6. Результат действия функции $\text{int}(x)$ имеет тип

- 1) целый 2) символьный 3) логический 4) вещественный

7. Установить соответствие

1	$\text{frac}(x)$	A	целая часть x
2	$\text{trunc}(x)$	B	округление до ближайшего целого
3	$\text{round}(x)$	C	дробная часть x
4	$\text{int}(x)$	D	отсечение дробной части

8. Установить соответствие

1	$\text{succ}(x)$	A	уменьшение x на 1
2	$\text{pred}(x)$	B	следующее значение x
3	$\text{inc}(x)$	C	увеличение x на 1
4	$\text{dec}(x)$	D	предыдущее значение x

9. Установить соответствие

1	$\text{abs}(x)$	A	экспонента x
2	$\tan(x)$	B	знак x
3	$\text{power}(x, y)$	C	модуль x
4	$\text{exp}(x)$	D	случайное целое из диапазона $[0; x)$
5	$\text{random}(x)$	E	тангенс x
6	$\text{sign}(x)$	F	возведение в степень

10. Установить соответствие между функцией и типом возвращаемого результата

1	odd(x)	A	целый
2	trunc(x)	B	вещественный
3	frac(x)	C	логический
4	chr(x)	D	символьный
5	int(x)	E	
6	sqrt(x)	F	
7	ord(x)	G	

11. Установить соответствие

1	odd(x)	A	возводит x в квадрат
2	power(x, y)	B	тангенс x
3	swap(x, y)	C	нечетность значения x
4	tan(x)	D	определяет большее из чисел x и y
5	max(x, y)	E	порядковый номер значения x
6	ord(x)	F	возводит x в степень y
7	sqr(x)	G	меняет местами значения x и y

Простые типы данных. Описание переменных

1. К целому типу данных не относятся

- 1) shortint 2) byte 3) single 4) word
5) char 6) double 7) longword 8) decimal

2. К вещественному типу данных относятся

- 1) byte 2) double 3) word
4) longint 5) decimal 6) single

3. Порядковыми типами являются

- 1) целый 2) символьный 3) вещественный
4) строковый 5) множественный

4. К целым типам относятся

- 1) double 2) uint64 3) word 4) integer
5) char 6) single 7) shortint 8) decimal

5. К вещественным типам относятся

- 1) double 2) decimal 3) word 4) integer
5) real 6) single 7) record 8) byte

6. Установить соответствие

1	символьный	A	set
2	массив	B	integer
3	целый	C	string
4	вещественный	D	array
5	множество	E	real
6	строка	F	char

7. Установить соответствие

1	логический	A	array
2	массив	B	set
3	символьный	C	record
4	запись	D	char
5	множество	E	string
6	строка	F	boolean

8. Установить соответствие

1	double	A	целый
2	byte	B	вещественный
3	longword		
4	decimal		
5	single		
6	smallint		
7	uint64		
8	integer		

9. Выберите правильные описания констант

- 1) e = 2.72; 2) zero := 0; 3) at = 'at';
 4) Language = 'Pascal'; 5) no := false; 6) yes = true;

10. Выберите неправильные описания констант

- 1) al: const; 2) at = 'at'; 3) no = false;
 4) day = 'Monday'; 5) Language = Pascal; 6) m = 1..6;

11. Переменные объявлены правильно

- 1) var a: const; 2) var sum, zero: real;
 3) var Language = 'Pascal'; 4) var m, con: byte;
 5) var no = false; 6) var number: 1..9;

12. Допустимые описания переменных в PascalABC.net

- 1) var (st1, st2) := (true, true); 2) var (a, b) := ReadReal2;
 3) var (a, bl) := (2.5, false); 4) var (d2, d4) := ReadInteger;
 5) var (s1, s2, s3) := (1.2, 1.8, 2.4); 6) var d1, d2 := 5;

13. Допустимые описания переменных в PascalABC.net

- | | |
|---|-----------------------------|
| 1) var yes: true; | 2) var e: byte := 2.72; |
| 3) var at: char; | 4) var zero: boolean; |
| 5) var m, n, sum: real := 2, 12.3, 0.2; | 6) var (a1, a2) := (4, 12); |

14. Допустимые описания переменных в PascalABC.net

- | | |
|-----------------------------|----------------------------|
| 1) var ab: real := '3.3'; | 2) var r1: integer := 3.5; |
| 3) var d1 := ReadChar; | 4) var a: integer = 23; |
| 5) var b: boolean := false; | 6) var a := 'abc'; |

15. Тип integer могут иметь переменные

- | | | |
|----------------|---------------------|----------------|
| 1) A1 := 6.96; | 2) A2 := 'Integer'; | 3) A3 := true; |
| 4) A4 := 164; | 5) A5 := '5'; | 6) A6 := 1; |

16. Тип boolean имеют переменные

- | | | |
|---------------------|-----------------|-----------------|
| 1) E1 := 'Boolean'; | 2) E2 := true; | 3) E3 := 163; |
| 4) E4 := 'true'; | 5) E5 := false; | 6) E6 := x > 0; |

17. Тип char имеют переменные

- | | | |
|-----------------|------------------|--------------|
| 1) D1 := 'C'; | 2) D2 := 'Char'; | 3) D3 := 12; |
| 4) D4 := 754.3; | 5) D5 := '8'; | |

18. Тип byte могут иметь переменные

- | | | |
|----------------|------------------|-----------------|
| 1) F1 := 1080; | 2) F2 := 'Byte'; | 3) F3 := 0; |
| 4) F4 := 100; | 5) F5 := 0.5; | 6) F6 := false; |

19. Тип integer могут иметь переменные

- | | | |
|---------------|----------------|------------------|
| 1) A1 := 0; | 2) A2 := 0.0; | 3) A3 := -345; |
| 4) A4 := 300; | 5) A5 := '23'; | 6) A6 := 'byte'; |

20. Тип byte могут иметь переменные

- | | | |
|---------------|---------------|---------------|
| 1) B1 := -2; | 2) B2 := 256; | 3) B3 := 0; |
| 4) B4 := '0'; | 5) B5 := 1.0; | 6) B6 := 255; |

Ввод-вывод, присваивание

1. Результат действия оператора write('x=', x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

2. Результат действия последовательности операторов print('x='); read(x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

3. Результат действия оператора read(x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

4. Результат действия оператора println(x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

5. Процедура readln(x, y, ...) предназначена для

- 1) вывода на экран строки символов x y ...
- 2) ввода значений переменных x, y, ... из файла
- 3) ввода с клавиатуры строки
- 4) ввода с клавиатуры значений переменных x, y, ...
- 5) вывода на экран значений переменных x, y, ...

6. Установить соответствие

1	read	A	ВВОД
2	print	B	ВЫВОД
3	readinteger		
4	readchar		
5	println		
6	writeln		
7	readreal2		
8	writeformat		

7. Установить соответствие

1	writeln('x =', x)	A	12 22
2	print(a, b)	B	x =15.42
3	println('x =', x)	C	x y
4	writeln('x =', x:5:2)	D	x =22
5	println('x', 'y')	E	xу
6	writeln('x', 'y')	F	x = 22

8. Установить соответствие

1	ReadInteger3	A	значение логического типа
2	Readreal2	B	значение вещественного типа
3	ReadReal	C	3 значения целого типа
4	ReadChar	D	3 строки
5	ReadBoolean	E	значение символьного типа
6	ReadString3	F	2 значения вещественного типа

9. Установить соответствие

1	Writeln(x, y)	A	ввод значений x и y
2	Print(x, y)	B	вывод значений x и y
3	Readln	C	перевод курсора в новую строку
4	Read(x, y)	D	вывод последовательности символов x y
5	Writeln	E	вывод последовательности символов ху
6	Println('x', 'y')	F	пауза в выполнении программы
7	Write('x', 'y')		
8	Println		

10. Функциями ввода данных в PascalABC.net являются

- 1) Println 2) ReadReal3 3) Read 4) WriteFormat
5) ReadBoolean 6) Prtint 7) ReadChar

11. Функциями вывода данных в PascalABC.net являются

- 1) Write 2) ReadInteger3 3) Readln 4) ReadChar2
5) ReadBoolean 6) Prtint 7) WriteFormat

12. Правильно организован ввод данных

- 1) Readln(a, b, alfa); 2) var a4 := 3.5;
3) var zx: integer := 2.8; 4) var (a, b) := ReadChar;
5) var a := read(b); 6) var (a, b) := ReadChar2;

13. Правильно организован ввод данных

- 1) var a4: integer := '3';
2) var (zx, zy) := ReadInteger2 := 2, 8;
3) var (a, b) := ReadChar2('Введите 2 символа');
4) var st := ReadBoolean;
5) var (a, b) := ReadChar2;
6) var a := ReadReal;

14. Правильно организован вывод данных

- 1) Writeln; 2) Print(a, b);
3) Println(x:5:2); 4) Println('k =', k);
5) Writeln(g:6); 6) Writeln('a = ', a);

15. Правильно организован вывод данных

- 1) Write('f =', f:6:3); 2) Println;
3) Print(a, 'b =', b); 4) Writeln;
5) Println(a:4, b:4); 6) Writeln('h = ', h:5);

16. Выберите все верные утверждения

- 1) Процедура Println(a, b, c) выводит на экран строку 'abc'
2) Процедура WriteFormat(formatstr; params) выводит значения params в соответствии с форматной строкой formatstr
3) Функция ReadInteger4 позволяет одновременно вводить 4 значения переменных целого типа

4) Процедура `Writeln` без параметров приостанавливает работу программы до нажатия клавиши `Enter`

17. Выберите все верные утверждения

- 1) Процедура `Println(a, b, c)` выводит на экран значения переменных `a`, `b`, `c`
- 2) Функция `ReadChar2` возвращает кортеж из 2 символьных значений, введенных с клавиатуры
- 3) Процедура `Write(a, b)` добавляет пробелы при выводе значений на экран
- 4) Функция `ReadChar` позволяет одновременно ввести значения трех логических переменных

18. Выберите все верные утверждения

- 1) Процедура `Println(...)` допускает форматный вывод данных
- 2) Процедура `Println(a, b, c)` добавляет пробелы при выводе значений на экран
- 3) Функция `ReadReal` возвращает кортеж из 2 действительных значений, введенных с клавиатуры
- 4) Процедура `Readln` без параметров приостанавливает работу программы до нажатия клавиши `Enter`

19. Выберите все верные утверждения

- 1) Процедура `Writeln(...)` допускает форматный вывод данных
- 2) Процедура `WriteFormat(formatstr; params)` позволяет форматировать вывод значений `params`
- 3) Функция `ReadBoolean` без параметров позволяет ввести значение логической переменной из файла
- 4) Процедура `Println` без параметров переводит курсор в новую строку

20. В результате выполнения последовательности операторов

```
c := 3; a := c*2; c := a - 4;
```

переменные `a` и `c` примут значения соответственно (перечислите через пробел)

21. В результате выполнения последовательности операторов

```
a := 5; c := a + 1; a -= 2; c -= a;
```

переменные `a` и `c` примут значения соответственно (перечислите через пробел)

22. В результате выполнения последовательности операторов

```
a := 48; b := a div 5; c := b - 2; a := a mod c;
```

переменные `a` и `c` примут значения соответственно (перечислите через пробел)

23. В результате выполнения последовательности операторов
a := 24; c := a div 3; a /= c; c += a;
переменные a и c примут значения соответственно (перечислите через пробел)

Операторы ветвления

1. Для организации алгоритма ветвления применяются операторы

- 1) цикла с параметром
- 2) условный
- 3) выбора
- 4) безусловного перехода
- 5) цикла с постусловием

2. Для организации алгоритма ветвления применяются операторы

- 1) repeat
- 2) loop
- 3) case
- 4) foreach
- 5) if
- 6) while

3. В результате выполнения фрагмента программы

```
a := -4; b := 1;  
if a < 0 then b += a else b := 0;  
переменная b принимает значение ___
```

4. В результате выполнения фрагмента программы

```
a := 8; b := 0;  
if 3*a < 25 then b -= a else b := a;  
переменная b принимает значение ___
```

5. В результате выполнения фрагмента программы

```
a := -1; b := 0; c := 3;  
if 3*a = b + c then b := a*c else b := a + c;  
переменная b принимает значение ___
```

6. В результате выполнения фрагмента программы

```
a := 14; b := 3; c := 1;  
if a mod b = 1 then c += b else c += a;  
переменная c принимает значение ___
```

7. В результате выполнения фрагмента программы

```
a := 2; b := 7; c := -8;  
if a + c > b then c += a else c -= a;  
переменная c принимает значение ___
```

8. В результате выполнения фрагмента программы

```
a := 15; b := -4; c := 7;  
if a div c = 2 then c += b else c := a + b;  
переменная c принимает значение ___
```

9. В результате выполнения фрагмента программы

```
c := 15; a := 6; d := 3;  
if a + d > c then c := a*d;  
переменная c принимает значение ___
```

10. В результате выполнения фрагмента программы

```
c := 7; a := 20; d := 4;  
if a mod c < d then c := a div d;  
переменная c принимает значение ___
```

11. В результате выполнения фрагмента программы

```
read(a, b);  
if a > b then c := a else c := b;  
write(c);
```

на экран будет выведено

- 1) произведение чисел a и b, если a положительно, b отлично от нуля
- 2) меньшее из чисел a и b
- 3) сумма чисел a и b, если они оба положительны
- 4) большее из чисел a и b

12. В результате выполнения фрагмента программы

```
read(a, b);  
if (a > 0) and (b > 0) then c := a + b;  
write(c);
```

на экран будет выведено

- 1) произведение чисел a и b в том случае, если a положительно, b отлично от нуля
- 2) меньшее из чисел a и b
- 3) сумма чисел a и b в том случае, если они оба положительны
- 4) большее из чисел a и b

13. В результате выполнения фрагмента программы

```
read(a, b);  
if (a > 0) and (b <> 0) then c := a*b;  
write(c);
```

на экран будет выведено

- 1) произведение чисел a и b в том случае, если a положительно, b отлично от нуля
- 2) меньшее из чисел a и b
- 3) сумма чисел a и b в том случае, если они оба положительны
- 4) большее из чисел a и b

14. Правильно оформлен оператор выбора

```
1) var b, x: char; m, n: real;  
   case b of  
     '@': m := 100;  
     'y': n := 100;
```

```

    '+' : x := 'b';
end;
2) var a: char; m: real;
    case m of
        1: a := 'a';
        2: a := 'b';
        5: a := 'next';
    end;
3) var x: char; m, n: real; y: integer;
    case y of
        'a': m := 2.05;
        'm': n := 3.2;
        't': x := 'sand';
    end;
4) var x: char; y: integer;
    case x of
        2: y := 5;
        8: y := 2;
        9: y := 11;
    end;

```

15. Правильно оформлен оператор выбора

```

1) var a: real; m, n: integer; ar: boolean;
    case a of
        3: m := a + m;
        4: ar := true;
        5: n := 7;
    end;
2) var a: real; m, n: integer; ar: boolean;
    case m of
        4: a := cos(a);
        17: ar := false;
        21: n := m;
    end;
3) var a: real; m, n: integer; ar: boolean;
    case ar of
        'a': a := false;
        'b': n := m;
        'c': n := m + n;
    end;
4) var a: real; m, n: integer; ar: boolean;
    case n of
        1: a := false;
        2: m := m + a;
        3: ar := true;
    end;

```

16. В приведенном фрагменте кода
var x, y: real; m, n: integer ar: char;
case m **of**
 'a': x += 1;
 'b': y += x;
 'c': ar := 'no';
end;

неправильно:

- 1) описаны типы переменных
- 2) записаны стандартные имена в операторе выбора
- 3) использованы операторы присваивания
- 4) использованы типы переменных

17. Определите результат выполнения операторов при вводе значения

```
t = 8
read(t);
case t of
  1..6: y := t*t - 1;
  7, 9, 13: y := 7*t - t;
  10..12: y := abs(2 - t)
else y := 0
end;
```

18. Определите результат выполнения операторов при вводе значения

```
t = 5
read(t);
case t of
  1..6: y := t*t - 1;
  7, 9, 13: y := 7*t - t;
  10..12: y := abs(2 - t)
else y := 0
end;
```

19. Определите результат выполнения операторов при вводе значения

```
t = 11
read(t);
case t of
  1..6: y := t*t - 1;
  7, 9, 13: y := 7*t - t;
  10..12: y := abs(2 - t)
else y := 0
end;
```

20. Выберите все верные утверждения

1) В конструкции

if <условие> **then** <оператор_1> **else** <оператор_2>; <оператор_3>;
 оператор_2 выполняется, если условие принимает значение *false*

2) После исполнения фрагмента программы

```
var s := 0; var m := 0; var n := -4; var k := 2;  
if ((n + k) > 0) and ((n - k) < 0) then m := n*k;  
s := n + k;  
write(m, s);
```

на экране появятся значения 0 и -2

3) В конструкции

```
case <ключ> of  
  <список_1>: <оператор_1>  
  <список_2>: <оператор_2>  
  ...  
  <список_m>: <оператор_m>  
else <оператор_n>;  
end;
```

оператор_n выполняется в любом случае

21. Выберите все верные утверждения

1) В конструкции

```
if <условие> then <оператор_1> else <оператор_2>; <оператор_3>;
```

в качестве условия может использоваться константа логического типа

2) После исполнения фрагмента программы

```
s := 0; m := 0; n := -5; k := 3;  
if (n + k) > 0 and (n - k) < 0 then m := n*k;  
s := n + k; write(m, s);
```

на экране появятся значения -15 и -2

3) В конструкции

```
case <ключ> of  
  <список_1>: <оператор_1>  
  <список_2>: <оператор_2>  
  ...  
  <список_m>: <оператор_m>  
else <оператор_n>;  
end;
```

может выполняться более одного оператора

22. Выберите все верные утверждения

1) В конструкции

```
if <условие> then <оператор_1> else <оператор_2>; <оператор_3>;  
часть else <оператор_2> является обязательной
```

2) Фрагмент программы

```
if odd(x) then y := 2*x else y := x;  
при x = 12 присвоит переменной y значение 12
```

3) В конструкции

```
case <ключ> of  
  <список_1>: <оператор_1>  
  <список_2>: <оператор_2>
```

```
...
<список_m>: <оператор_m>
else <оператор_n>;
end;
```

ключ может быть переменной диапазонного типа

23. Выберите все верные утверждения

1) В конструкции

```
if <условие> then <оператор_1> else <оператор_2>; <оператор_3>;
```

если *условие* имеет значение *true*, *оператор_2* и *оператор_3* не выполняются

2) Фрагмент программы

```
var (a, b, c) := ReadReal3;
var d := a;
if b > d then d := b;
if c > d then d := c;
writeln(d);
```

выводит на экран наибольшее из трех действительных чисел

3) В конструкции

```
case <ключ> of
  <список_1>: <оператор_1>
  <список_2>: <оператор_2>
  ...
  <список_m>: <оператор_m>
else <оператор_n>;
end;
```

ни один из *операторов* не может быть оператором цикла

Операторы цикла

1. В операторе цикла с предусловием тело цикла

- 1) обязательно выполняется более 1 раза
- 2) выполняется по крайней мере 1 раз
- 3) может не выполниться вообще
- 4) выполняется строго определенное число раз

2. В операторе цикла с постусловием тело цикла

- 1) обязательно выполняется более 1 раза
- 2) выполняется по крайней мере 1 раз
- 3) может не выполниться вообще
- 4) выполняется строго определенное число раз

3. В операторе цикла с параметром *for* тело цикла

- 1) обязательно выполняется более 1 раза
- 2) выполняется по крайней мере 1 раз
- 3) может не выполниться вообще
- 4) выполняется строго определенное число раз

4. Установить правильную последовательность

- 1 **until**
- 2 *тело цикла*
- 3 **repeat**
- 4 *условие*

5. Установить правильную последовательность

- 1 *тело цикла*
- 2 *условие*
- 3 **while**
- 4 **do**

6. Выберите все верные утверждения

- 1) Оператор `loop` является оператором цикла
- 2) В описании оператора `foreach` тип переменной цикла должен совпадать с типом элементов контейнера
- 3) В операторе `foreach` изменение переменной цикла в общем случае изменяет элементы контейнера
- 4) Количество исполнений тела цикла в операторе `repeat` определяется до начала работы цикла

7. Выберите все верные утверждения

- 1) Оператор `loop` является оператором ветвления
- 2) В операторе `foreach` элементы контейнера доступны только для чтения
- 3) В операторе `foreach` строка не может быть контейнером
- 4) Операторы `while` и `repeat` относятся к группе операторов цикла с условием

8. Выберите все верные утверждения

- 1) В конструкции **loop** <выражение> **do** <оператор> *выражение* должно иметь целый тип данных
- 2) В описании оператора `foreach` переменная цикла и элементы контейнера могут иметь разные типы данных
- 3) В операторе `foreach` контейнером может быть статический или динамический массив
- 4) Операторы `loop` и `foreach` относятся к группе операторов цикла с условием

9. Выберите все верные утверждения

- 1) -В описании **loop** <выражение> **do** <оператор> *выражение* может иметь логический тип данных
- 2) -Оператор

```
foreach var r in t do  
  begin  
    r += 1; print(r)
```

end

в случае `var t := [1..8]` выводит на экран все целые числа от 2 до 9 через пробел

3) +В описании оператора `foreach` тип переменной цикла должен совпадать с типом элементов контейнера

4) +В операторе цикла с постусловием тело цикла выполняется по крайней мере один раз

10. Выберите все верные утверждения

1) Оператор `loop` используется, когда тело цикла не зависит от номера итерации

2) Оператор `foreach var r in t do print(r)` в случае `var t := [1..6]` выводит на экран все целые числа от 1 до 6 через пробел

3) Оператор `loop` можно использовать для вычисления конечных сумм

4) Количество исполнений тела цикла в операторе `repeat` определяется до начала работы цикла

11. Выберите все верные утверждения

1) Оператор `loop` можно использовать для вычисления конечных сумм

2) В операторе `foreach` изменение переменной цикла в общем случае изменяет элементы контейнера

3) В операторе `foreach` элементы контейнера доступны только для чтения

4) Операторы `while` и `repeat` относятся к группе операторов цикла с условием

12. В результате выполнения фрагмента программы

```
var y := 1; var i := 2;  
while i <= 5 do  
  begin  
    y := y*i;  
    i += 1;  
  end;  
  write(y);
```

на экран будет выведено значение

13. В результате выполнения фрагмента программы

```
var (z, n) := (2, 3);  
while n < 8 do  
  begin  
    z := 2*z - n;  
    n += 2;  
  end;  
  write(z);
```

на экран будет выведено значение

14. В результате выполнения фрагмента программы

```
var y := 1; var i := 1;
while i <= 4 do
begin
    y := y*i;
    i += 1;
end;
```

```
write(y);
```

на экран будет выведено значение

15. В результате выполнения фрагмента программы

```
var (z, n) := (2, 3);
repeat
    z:= 2*z - n;
    n += 2
until n >= 7;
write(z);
```

на экран будет выведено значение

16. В результате выполнения фрагмента программы

```
var y := 1; var i := 2;
repeat
    y := y*i;
    i += 1
until i >= 5;
write(y);
```

на экран будет выведено значение

17. Фрагмент программы выполняет ввод целых чисел и вывод на экран отрицательных из них. Процесс прекращается, когда вводится ноль. Установить правильную последовательность строк.

```
1  if m < 0
2  repeat
3  m = 0;
4  then print(m)
5  var
6  readln(m);
7  m: integer;
8  until
```

18. Фрагмент кода составляет таблицу значений функции.

Установить правильную последовательность строк.

```
1  println('x =', x, 'y =', y);
2  var x := a;
3  x > b
4  var (a, b, h) := ReadReal3('Введите 3 числа');
5  x += h
6  repeat
```

```

7  var y := x**5 * cos(x);
8  until

```

19. Дан фрагмент программного кода.

Установить правильную последовательность строк.

```

1  to length(st)
2  for var i := 1
3  var a in st
4  do
5  if st[i] in ['0'..'9']
6  var st := ReadString;
7  foreach
8  do print(a)
9  then st[i] := ' ';

```

20. Фрагмент кода вычисляет сумму квадратов n первых натуральных чисел. Установить правильную последовательность строк.

```

1  s += n
2  for var i := 1
3  print('s =', s)
4  var s := 0; var n := 0;
5  n := i*i;
6  begin
7  to 10 do
8  end;

```

21. Фрагмент кода вычисляет конечную сумму

$$\sum_{i=1}^{12} \frac{\sin i}{i}$$

Установить правильную последовательность строк.

```

1  to 12 do
2  n := cos(i)/i;
3  end;
4  var s := 0.0; var n := 0.0;
5  writeln('s = ', s:5:3)
6  for var i := 1
7  begin
8  s += n

```

22. Фрагмент кода выводит на экран все цифры заданного натурального числа. Установить правильную последовательность строк.

```

1  n := n div 10;
2  println(n mod 10);
3  until
4  var n := ReadInteger;
5  n = 0;
6  repeat

```

Структурированные типы данных

1. К структурированным типам относятся

- 1) вещественный 2) файл 3) множество
- 4) символьный 5) диапазон 6) целый

2. К структурированным типам относятся

- 1) символьный 2) строка 3) массив
- 4) диапазон 5) логический 6) запись

3. К структурированным типам не относятся

- 1) вещественный 2) множество 3) символьный
- 4) диапазон 5) массив 6) файл

4. Выберите все верные утверждения.

Тип данных определяет

- 1) формат их внутреннего представления в памяти компьютера
- 2) множество значений, которые может принимать данный объект
- 3) количество объектов в программе, которые могут принимать данные значения
- 4) количество символов в идентификаторе, присваиваемом объекту данного типа
- 5) множество допустимых операций, применяемых к данным

5. Выберите все верные утверждения.

- 1) логический тип данных относится к простым
- 2) строковый тип данных относится к простым
- 3) символьный тип данных относится к структурированным
- 4) указатель относится к структурированным типам данных
- 5) перечислимый тип данных относится к простым

6. Выберите все верные утверждения.

- 1) вещественный тип данных относится к порядковым
- 2) строковый тип данных относится к порядковым
- 3) файл относится к структурированным типам данных
- 4) целый тип данных относится к порядковым
- 5) запись относится к структурированным типам данных

7. Структура данных, состоящая из фиксированного количества компонентов, принадлежащих в общем случае к разным типам, называется ____

8. Неупорядоченный набор однотипных данных, имеющих общее имя, называется ____

9. Индексированный именованный набор однотипных данных называется ____

10. Именованный линейный набор символов называется ____

11. Последовательность однотипных данных, хранящихся во внешней памяти, называется ____

12. **type**

```
    Birthday = record
        day, month: byte;
        year: word;
        .....;
    var a, b: Birthday;
```

В приведенном фрагменте пропущено слово

13. Выберите все верные утверждения

- 1) Базовым типом множества в PascalABC может быть любой тип, кроме указателя
- 2) Строки могут быть элементами множества
- 3) Длина строки определяется на этапе компиляции и не может изменяться во время работы программы
- 4) Поля записи должны принадлежать к одному типу данных

14. Выберите все верные утверждения

- 1) В PascalABC существует тип данных запись с вариантами
- 2) К множествам операции сравнения неприменимы
- 3) Элементы множества неупорядочены
- 4) Нулевой байт строки содержит информацию о ее длине

15. Выберите все верные утверждения

- 1) Множество не может содержать одинаковых элементов
- 2) Процедура include предназначена для проверки принадлежности элемента множеству
- 3) Сравнение строк выполняется посимвольно
- 4) Записи не могут быть элементами множества

16. Выберите все верные утверждения

- 1) Доступ к элементу множества выполняется по его индексу
- 2) Номер элемента в строке может изменяться во время работы программы
- 3) Процедура Print позволяет вывести на экран все элементы множества
- 4) Базовым типом множества в PascalABC может только простой тип данных

Массивы

1. Выберите все правильные описания массивов

- 1) var b: array [-2..8] of integer;
- 2) var m: array [20..1] of char;
- 3) var day: array [1..'n'] of char;
- 4) var ch: array ['A'..'T'] of byte;
- 5) var letter: array ['m'..'b'] of integer;
- 6) var ord: array ['b'..'m'] of real;

2. Выберите все правильные описания массивов

- 1) var b: array ['a'..10] of integer;
- 2) var map: array ['a'; 'd'] of boolean;
- 3) var yes: array [10..20] of char;
- 4) var chili: array ['A'..'N'] of real;
- 5) var let: array [-8..-1] of boolean;
- 6) var d: array ['d'...'p'] of real;

3. Выберите все правильные описания массивов

- 1) var dis: array ['a'..'n'] of real;
- 2) var read: array ['t'..'a'] of integer;
- 3) var may: array [10..7] of char;
- 4) var base: array ['a'..'122'] of boolean;
- 5) var d: array ['1'..'7'] of real;
- 6) var abs: array [-10..0] of single;

4. Выберите все правильные описания массивов

- 1) var ch: array ['A'..'T'] of byte;
- 2) var map: array ['a'; 'd'] of boolean;
- 3) var n: array [-10..0] of single;
- 4) var m: array [20..1] of char;
- 5) var chili: array ['A'..'N'] of real;
- 6) var s1: array ['a'..'122'] of boolean;

5. Фрагмент кода

```
var s := a[0];  
for var i := 1 to 9 do if a[i] > s then s := a[i];
```

вычисляет

- 1) максимальный элемент в массиве из 9 элементов
- 2) минимальный элемент в массиве из 10 элементов
- 3) максимальный элемент в массиве из 10 элементов
- 4) минимальный элемент в массиве из 9 элементов

6. Фрагмент кода

```
var s := a[9];  
for var i := 8 downto 1 do if a[i] < s then s := a[i];
```

вычисляет

- 1) максимальный элемент в массиве из 9 элементов
- 2) минимальный элемент в массиве из 8 элементов
- 3) максимальный элемент в массиве из 8 элементов
- 4) минимальный элемент в массиве из 9 элементов

7. Фрагмент кода

```
var s := a[1]; var i := 2;
repeat
  if a[i] > s then s := a[i];
  i += 1;
until i >= 11;
```

вычисляет максимальный элемент в массиве из

- 1) 9 'ktvtynjd
- 2) 10 'ktvtynjd
- 3) 11 'ktvtynjd
- 4) 12 'ktvtynjd

8/ Ahfuvtyn rjlf

```
var s ^:= a[1]$ var i ^:= 1$
repeat
  i += 1$
  if a[i] > s then s := a[i];
until i >= 11;
```

вычисляет максимальный элемент в массиве из

- 1) 9 'ktvtynjd
- 2) 10 'ktvtynjd
- 3) 11 'ktvtynjd
- 4) 12 'ktvtynjd

9/ Ahfuvtyn rjlf

```
var s ^:= a[1]$ var i ^:= 2$
while i < 11 do
begin
  if a[i] < s then s ^:= a[i];
  i += 1;
end;
```

вычисляет минимальный элемент в массиве из

- 1) 8 'ktvtynjd
- 2) 9 'ktvtynjd
- 3) 10 'ktvtynjd
- 4) 11 'ktvtynjd

10/ Ahfuvtyn rjlf

```
var s ^:= 0$ var i ^:= 1$
while i < 9 do
begin
  s += a[i]$
  inc(i);
end;
```

вычисляет

- 1) сумму положительных элементов массива из 9 элементов
- 2) количество отрицательных элементов в массиве из 8 элементов
- 3) сумму всех элементов массива из 8 элементов
- 4) количество отрицательных элементов в массиве из 9 элементов

11. Фрагмент кода

```
var s := 0; var i := 1;
repeat
  if a[i] < 0 then s += a[i];
  inc(i)
until i = 10;
```

вычисляет

- 1) сумму положительных элементов массива из 10 элементов
- 2) сумму отрицательных элементов массива из 9 элементов
- 3) количество отрицательных элементов в массиве из 9 элементов
- 4) количество положительных элементов в массиве из 10 элементов

12. Фрагмент кода

```
var s := 0; var j := 1;
repeat
  j += 1;
  if a[j] > 0 then s += 1;
until j > 11;
```

вычисляет

- 1) сумму положительных элементов массива из 12 элементов
- 2) количество положительных элементов в массиве из 12 элементов
- 3) сумму положительных элементов массива из 11 элементов
- 4) количество положительных элементов в массиве из 11 элементов

13. Фрагмент кода вычисляет количество элементов одномерного массива из 10 элементов, которые меньше 2. Установите правильную последовательность строк.

```
1 then
2 i -= 1;
3 var s := 0; var i := 10;
4 until
5 s += 1;
6 if
7 repeat
8 i <= 0
9 a[i] < 2
```

14. Фрагмент программы вычисляет количество элементов в одномерном массиве из 12 элементов, которые больше 1. Установите правильную последовательность строк.

```
1 then
2 do
3 k += 1;
4 end;
5 i <= 12
6 var k := 0; var i := 1;
7 inc(i)
```

```
8  if a[i] > 19  while
10 begin
```

15. Фрагмент программы вычисляет минимальный элемент в одномерном массиве из 9 элементов. Установите правильную последовательность строк.

```
1  if
2  do
3  then
4  end;
5  x <= 10
6  s := a[1]; var i := 2;
7  i += 1
8  begin
9  while
10 a[i] < s
11 s := a[i];
```

16. Фрагмент программы вычисляет максимальный элемент в одномерном массиве из 10 элементов. Установите правильную последовательность строк.

```
1  s := a[i];
2  if
3  until
4  var s := a[10]; var i := 9;
5  i <= 0;
6  a[i] > s
7  repeat
8  then
9  i -= 1
```

Двумерные массивы

1. Задан двумерный массив 5×5 . В результате выполнения фрагмента программы

```
for var i := 1 to 5 do
  for var j := 1 to 5 do
    a[i, j] := i*j - 7;
```

значение элемента a_{25} будет равно

2. Задан двумерный массив 5×6 . В результате выполнения фрагмента программы

```
for var i := 1 to 5 do
  for var j := 1 to 6 do
    a[i, j] := i + 2*j;
```

значение элемента a_{53} будет равно

3. Задан двумерный массив 4×6 . В результате выполнения фрагмента программы

```
for var i := 1 to 4 do
  for var j := 1 to 6 do
    a[i, j] := (i + j)*3;
```

значение элемента a_{51} будет равно

4. Фрагмент программы

```
for var i := 1 to 5 do
  for var j := 1 to 7 do
    read(a[i, j]);
```

определяет ввод массива

- 1) размером 5×7 по столбцам
- 2) размером 5×7 по строкам
- 3) размером 7×5 по строкам
- 4) размером 7×5 по столбцам

5. Фрагмент программы

```
for var i := 1 to 7 do
  for var j := 1 to 5 do
    read(a[i, j]);
```

определяет ввод массива

- 1) размером 5×7 по столбцам
- 2) размером 5×7 по строкам
- 3) размером 7×5 по строкам
- 4) размером 7×5 по столбцам

6. Фрагмент программы

```
for var j := 1 to 7 do
  for var i := 1 to 5 do
    read(a[i, j]);
```

определяет ввод массива

- 1) размером 5×7 по столбцам
- 2) размером 5×7 по строкам
- 3) размером 7×5 по строкам
- 4) размером 7×5 по столбцам

7. Фрагмент программы

```
for var j := 1 to 5 do
  for var i := 1 to 7 do
    read(a[i, j]);
```

определяет ввод массива

- 1) размером 5×7 по столбцам
- 2) размером 5×7 по строкам
- 3) размером 7×5 по строкам
- 4) размером 7×5 по столбцам

8. Все элементы двумерного массива 4×4 первоначально равны нулю. Сколько элементов станут равными 1 в результате выполнения фрагмента программы

```
for var i := 1 to 4 do
  for var k := i to 4 do
    a[i, k] := 1;
```

9. Все элементы двумерного массива 7×7 первоначально равны нулю. Сколько элементов станут равными 1 в результате выполнения фрагмента программы

```
for var i := 1 to 4 do
  for var k := i to 4 do
    begin
      a[i, k] := a[i, k] + 1;
      a[k, i] := a[k, i] + 1;
    end;
```

10. Задан двумерный массив 7×7 . Следующий фрагмент программы

```
var s := 0;
for var i := 1 to 7 do
  for var j := 1 to 7 do
    if a[i, j] > 0 then s += a[i, j];
```

определяет

- 1) сумму всех положительных элементов
- 2) модуль суммы всех отрицательных элементов
- 3) сумму обратных величин всех ненулевых элементов
- 4) сумму всех отрицательных элементов

11. Задан двумерный массив 7×7 . Следующий фрагмент программы

```
var s := 0;
for var i := 1 to 7 do
  for var j := 1 to 7 do
    if a[i, j] < 0 then s -= a[i, j];
```

определяет

- 1) сумму всех положительных элементов
- 2) модуль суммы всех отрицательных элементов
- 3) сумму обратных величин всех ненулевых элементов
- 4) сумму всех отрицательных элементов

12. Задан двумерный массив 7×7 . Следующий фрагмент программы

```
var s := 0.0;
for var i := 1 to 7 do
  for var j := 1 to 7 do
    if a[i, j] <> 0 then s += 1/a[i, j];
```

определяет

- 1) сумму всех положительных элементов
- 2) модуль суммы всех отрицательных элементов
- 3) сумму обратных величин всех ненулевых элементов
- 4) сумму всех отрицательных элементов

13. Задан двумерный массив 7×7 . Следующий фрагмент программы

```
var s := 0;
for var i := 1 to 7 do
  for var j := 1 to 7 do
    if a[i, j] < 0 then s += a[i, j];
```

определяет

- 1) сумму всех положительных элементов
- 2) модуль суммы всех отрицательных элементов
- 3) сумму обратных величин всех ненулевых элементов
- 4) сумму всех отрицательных элементов

14. Задан двумерный массив 5×5 . Следующий фрагмент программы

```
var s := 0;  
for var i := 1 to 5 do  
  for var j := 1 to 5 do  
    if i = j then s += a[i, j];
```

определяет

- 1) сумму элементов главной диагонали
- 2) удвоенную сумму элементов 2-го столбца
- 3) сумму элементов 1-й строки
- 4) удвоенную сумму элементов главной диагонали

15. Задан двумерный массив 5×5 . Следующий фрагмент программы

```
var s := 0;  
for var i := 1 to 5 do  
  for var j := 1 to 5 do  
    if j = 2 then s += a[i, j]*2;
```

определяет

- 1) сумму элементов главной диагонали
- 2) удвоенную сумму элементов 2-го столбца
- 3) сумму элементов 1-й строки
- 4) удвоенную сумму элементов главной диагонали

16. Задан двумерный массив 5×5 . Следующий фрагмент программы

```
var s := 0;  
for var i := 1 to 5 do  
  for var j := 1 to 5 do  
    if i = 1 then s += a[i, j];
```

определяет

- 1) сумму элементов главной диагонали
- 2) удвоенную сумму элементов 2-го столбца
- 3) сумму элементов 1-й строки
- 4) удвоенную сумму элементов главной диагонали

17. Задан двумерный массив 5×5 . Следующий фрагмент программы

```
var s := 0;  
for var i := 1 to 5 do  
  for var j := 1 to 5 do  
    if i = j then s += a[i, j]*2;
```

определяет

- 1) сумму элементов главной диагонали
- 2) удвоенную сумму элементов 2-го столбца

- 3) сумму элементов 1-й строки
- 4) удвоенную сумму элементов главной диагонали

18. Задан двумерный массив 6*6. В результате выполнения фрагмента программы

```
for var i := 1 to 6 do
  for var j := 1 to 6 do
    if i = j then a[i, j] := 1 else a[i, j] := 2;
```

элементы массива примут следующие значения

- 1) элементы главной диагонали — значение 1, остальные — значение 2
- 2) элементы главной диагонали — значения, равные номеру строки, которой принадлежит данный элемент, остальные — значение 2
- 3) элементы главной диагонали — значение 2, остальные — значение 1
- 4) элементы главной диагонали — значения, равные номеру столбца, которому принадлежит данный элемент, остальные — значение 1

19. begin

```
var a: array[1..6, 1..4] of char;
var b: array [1..6, 1..4] of byte;
for var i := 1 to 6 do
  for var j := 1 to 4 do read(a[i, j]);
var i := 1;
repeat
  for var j := 1 to 4 do b[i, j] := ord(a[i, j]);
  inc(i)
until i = 7;
```

end.

Код выполняет

- 1) ввод целочисленного массива, замену его элементов символами, коды которых соответствуют элементам массива, и вывод полученного массива на экран
- 2) ввод символьного массива и преобразование его в целочисленный массив, состоящий из кодов соответствующих элементов исходного массива
- 3) ввод символьного массива, замену его элементов их кодами и вывод полученного массива на экран
- 4) ввод целочисленного массива и преобразование его в символьный массив, в котором коды символов соответствуют элементам исходного массива

20. begin

```
var a: array[1..6, 1..4] of char;
var b: array [1..6, 1..4] of byte;
for var i := 1 to 6 do
  for var j:= 1 to 4 do read(b[i, j]);
var i := 1;
repeat
```

```

    for var j := 1 to 4 do a[i, j] := chr(b[i, j]);
    i += 1
until i = 7;
end.

```

Код выполняет

- 1) ввод целочисленного массива, замену его элементов символами, коды которых соответствуют элементам массива, и вывод полученного массива на экран
- 2) ввод символьного массива и преобразование его в целочисленный массив, состоящий из кодов соответствующих элементов исходного массива
- 3) ввод символьного массива, замену его элементов их кодами и вывод полученного массива на экран
- 4) ввод целочисленного массива и преобразование его в символьный массив, в котором коды символов соответствуют элементам исходного массива

Строки, множества

1. Установить соответствие

1	concat(s1, s2)	A	вставка подстроки в строку
2	copy(s1, n, m)	B	удаление подстроки
3	delete(s1, n, m)	C	сцепление строк
4	insert(st1, st2, n)	D	отыскание подстроки в строке

2. Установить соответствие

1	+	A	включение элемента в множество
2	exclude	B	проверка принадлежности элемента множеству
3	in	C	исключение элемента из множества
4	-		
5	include		

3. begin

```

var s := ReadString;
var t := ReadString;
println(length(s) + length(t))
end.

```

Программа выводит на экран

- 1) количество символов в более длинной из двух строк
- 2) количество символов в более короткой из двух строк
- 3) число символов, на которое одна строка длиннее другой
- 4) суммарное количество символов в двух строках

4. **begin**

```
var s := ReadString;  
var t := ReadString;  
writeln(s + t);  
writeln(t + s)
```

end.

Программа выводит на экран

- 1) строку, полученную сцеплением строк s и t, из которой удалены повторяющиеся символы
- 2) строку, полученную в результате сцепления строк s и t, и ее длину
- 3) две строки — сначала s, потом t
- 4) две строки — результат сцепления s и t и результат сцепления t и s

5. **begin**

```
var s := ReadString;  
var t := ReadString;  
var m := length(s);  
var n := length(t);  
if m < n then writeln(m)  
else writeln(n);
```

end.

Программа выводит на экран

- 1) количество символов в более длинной из двух строк
- 2) количество символов в более короткой из двух строк
- 3) число символов, на которое одна строка длиннее другой
- 4) суммарное количество символов в двух строках

6. **begin**

```
var s := ReadString;  
var t := ReadString;  
var m := length(s);  
var n := length(t);  
if m > n then n := m;  
writeln(n)
```

end.

Программа выводит на экран

- 1) количество символов в более длинной из двух строк
- 2) количество символов в более короткой из двух строк
- 3) число символов, на которое одна строка длиннее другой
- 4) суммарное количество символов в двух строках

7. **begin**

```
var s := ReadString;  
var t := ReadString;  
var m := length(s);  
var n := length(t);  
n := abs(m - n);  
writeln(n)
```

end.

Программа выводит на экран

- 1) количество символов в более длинной из двух строк
- 2) количество символов в более короткой из двух строк
- 3) число символов, на которое одна строка длиннее другой
- 4) суммарное количество символов в двух строках

8. **begin**

```
var n := 0; var i := 1;
var s := ReadString;
while i <= length(s) do
  begin
    if s[i] <> '+' then n += 1;
    inc(i)
  end;
  println('n =', n)
```

end.

Программа выводит на экран

- 1) количество элементов в строке до символа '+', если он есть, и длину строки, если символ '+' в ней отсутствует
- 2) количество элементов в строке после символа '+', если он есть, и длину строки, если символ '+' в ней отсутствует
- 3) количество символов '+' в строке
- 4) количество символов в строке, отличных от '+'

9. **begin**

```
var s := ReadString;
var i := 1;
while i <= length(s) - 1 do
  begin
    if s[i] = '+' then
      begin
        s[i] := '?';
        s[i + 1] := ' ';
      end;
    inc(i)
  end;
  println(s)
```

end.

Программа выводит на экран

- 1) количество символов '+' и '?' в строке
- 2) строку, из которой удалены все символы '+' и '?'
- 3) строку, в которой все символы '?' заменены пробелами, и количество таких замен
- 4) строку, в которой каждый символ '+' заменен пробелом, а следующий за ним — символом '?'
- 5) количество символов '+' и '?' в строке и новую строку, в которой эти символы заменены пробелами

10. begin

```
var s: set of char;  
var n := ReadInteger;  
s := [];  
loop n do s += [chr(Random(32, 255))];  
foreach var k in s do print(k)
```

end.

Код выполняет

- 1) заполнение массива случайными символами
- 2) заполнение множества символами, вводимыми с клавиатуры
- 3) заполнение множества случайными целыми числами
- 4) заполнение массива случайными целыми числами
- 5) заполнение множества случайными символами

11. begin

```
var s: set of integer;  
s := [];  
var n := ReadInteger;  
loop n do  
s += [Random(-20, 20)];  
foreach var i in s do print(i)
```

end.

Код выполняет действия

- 1) заполнение множества случайными целыми числами и вывод его на экран
- 2) ввод целочисленного массива и вывод его на экран
- 3) заполнение множества случайными целыми числами и вывод на экран количества элементов в множестве
- 4) задание количества элементов множества и заполнение множества числами, вводимыми с клавиатуры
- 5) задание размера целочисленного массива, его заполнение и вывод на экран

12. begin

```
var s: set of integer;  
s := [];  
var n := ReadInteger;  
loop n do  
s += [Random(-30, 30)];  
foreach var i in s do  
if i mod 3 <> 0 then print(i)
```

end.

Код выполняет действия

- 1) задание количества элементов множества и заполнение множества числами, вводимыми с клавиатуры

- 2) заполнение массива случайными числами и вывод на экран тех, которые кратны 3
- 3) заполнение множества случайными целыми числами и вывод из них на экран некратных 3
- 4) задание количества элементов массива и вывод из них на экран некратных 3
- 5) заполнение множества случайными целыми числами и вывод на экран кратных 3

13. Фрагмент кода выполняет ввод целых чисел, заполнение множества символами с соответствующими кодами и вывод на экран цифр, которые есть в этом множестве. Установить правильную последовательность строк.

```

1  s += [chr(k)]
2  if i in ['0'..'9']
3  begin
4  var n := ReadInteger('Введите число от 33 до 255');
5  foreach var i in s do
6  then print(i)
7  var s: set of char; s := [];
8  var k := ReadInteger;
9  loop n do
10 end;
```

14. Фрагмент кода заполняет множество случайными действительными числами из промежутка [0; 10) и выводит на экран те из них, которые принадлежат отрезку [2; 4]. Установить правильную последовательность строк.

```

1  foreach var i
2  var k := Random;
3  var s: set of real; s := [];
4  writeln(i:5:3)
5  if (i >= 2) and (i <= 4)
6  loop n do
7  var n := ReadInteger;
8  then
9  end;
10 begin
11 s += [k*10]
12 in s do
```

15. Фрагмент кода заполняет множество строками, вводимыми с клавиатуры, и выводит на экран строки, длина которых больше 4 символов. Установить правильную последовательность строк.

```

1  in s do
2  end;
3  begin
4  length(i) >= 5
5  var s: set of string; s := [];
```

```

6   for var i := 1
7   then println(i)
8   var k := ReadLnString;
9   var n := ReadInteger;
10  if
11  s += [k]12 to n do
13  foreach var i

```

16. Фрагмент кода заполняет множество случайными целыми числами и выводит на экран те из них, которые положительны и при делении на 4 дают остаток 3. Установить правильную последовательность строк.

```

1   in s do
2   then
3   foreach var i
4   var n := ReadInteger;
5   (i > 0) and (i mod 4 = 3)
6   println(i)
7   var s: set of integer; s := [];
8   loop n do
9   if
10  s += [random(-10,30)];

```

Файлы

1. Правильно описаны переменные файлового типа

- | | |
|----------------------------|---------------------------|
| 1) var dik: file | 2) var dik = file of char |
| 3) var dik = text | 4) var dik: file of byte |
| 5) var dik: file of string | 6) var dik = file of text |

2. Правильно описаны переменные файлового типа

- | | |
|--------------------------|-----------------------------|
| 1) var dik: file | 2) var dik = file of record |
| 3) var dik: text | 4) var dik: file of integer |
| 5) var dik: file of file | 6) var dik = text |

3. Правильно описаны файловые типы данных

- | | |
|--------------------|----------------------------|
| 1) type dik: file | 2) type dik = file of char |
| 3) type dik: text | 4) type dik = file |
| 5) type dik = text | 6) type dik = file of file |

4. Правильно описаны файловые типы данных

- | | |
|--------------------|------------------------------|
| 1) type dik = file | 2) type dik = file of string |
| 3) type dik: text | 4) type dik: file of file |
| 5) type dik = text | 6) type dik: file of char |

5. Выберите допустимые описания типизированных файлов

- | | |
|-------------------|-------------------|
| 1) file of string | 2) file of char |
| 3) file of file | 4) file of record |

6. Процедура assign

- 1) присваивает имя создаваемому логическому файлу
- 2) изменяет имя объявленной ранее файловой переменной
- 3) открывает физический файл для записи информации
- 4) связывает объявленную файловую переменную с именем физического файла

7. Установить соответствие

1	rewrite	A	закрытие файла
2	reset	B	открытие файла для чтения и записи
3	append	C	удаление файла
4	erase	D	открытие файла для записи
5	close	E	переименование файла
6	rename	F	открытие файла для чтения

8. Выберите все верные утверждения

- 1) процедура append применима к любым типизированным файлам
- 2) процедура rewrite позволяет добавить информацию в ранее созданный файл
- 3) процедура reset иницирует файл для чтения
- 4) процедура close закрывает файл

Подпрограммы

1. Заголовок функции записан правильно

- 1) function ter(var a, b = real, n = integer)
- 2) function ter(a: real, b: real, n: integer): real
- 3) function ter(var a: real; b: real; n: integer): longint
- 4) function ter(a, b: real; n: integer): integer

2. Заголовок функции записан неправильно:

- 1) function ter(var a, b = real, n = integer): char
- 2) function ter(a: real, b: real, n: integer)
- 3) function ter(var a: real; b: real; n: integer): byte
- 4) function ter(a, b: real; n: integer): real

3. Заголовок процедуры записан правильно

- 1) procedure(a, b: real; m: char): char
- 2) procedure sept(var a: real; const c: string)
- 3) procedure sept(a, c: real; n: char)
- 4) procedure sept(var b: real, const c = 'day')

4. Заголовок процедуры записан неправильно

- 1) procedure(a, b: real; m: char): char
- 2) procedure sept(var a: real; const c: string)
- 3) procedure sept(a, c: real; n: char)
- 4) procedure sept(var b: real, const c = 'day'): real

5. Выберите правильное утверждение

Результатом обращения

- 1) к функции может быть число, символ, строка
- 2) к функции всегда является число, символ или массив
- 3) к процедуре может быть только число или запись
- 4) к процедуре может быть какое-либо действие

6. Задана процедура

```
procedure P(x, y:integer);  
  begin  
    y:= x + 1;  
  end;
```

В результате обращения:

```
var c := 2; var d := 0;
```

```
P(c*c + c, d);
```

```
println(d);
```

на экран будет выведен результат ___

7. Задана процедура

```
procedure Q(x, y: real);  
  begin  
    y := x*x - 2;  
  end;
```

В результате обращения:

```
var a := 2; var b := 0;
```

```
Q(a*2, b);
```

```
print(b);
```

на экран будет выведен результат ___

8. Задана процедура

```
procedure Q(x, y: real);  
  begin  
    y:= sqrt(x) + x;  
  end;
```

В результате обращения

```
var a := 9; var b := 0;
```

```
Q(a, b);
```

```
writeln(b);
```

на экран будет выведен результат ___

9. В результате выполнения фрагмента программы

```
function Z(x, y: real): real;  
  begin  
    if x > y then Z:= (x - y)*x else Z:= x*y;  
  end;
```

```
writeln(Z(2, 3));
```

на экран будет выведен результат ___

10. В результате выполнения фрагмента программы
function sept(a, b: real; n: integer): real;
 begin
 sept := a**n + b;
 end;
write(sept(3, 8, 2));
на экран будет выведен результат __

11. Задана функция. Определите тип возвращаемого результата
function f1(x, y: integer): ____;
begin
 f1 := x**y
end;

12. Задана функция. Определите тип возвращаемого результата
function f1(x, y: integer): ____;
begin
 f2 := x >= y
end;

Ответы к тестовым заданиям

Алфавит, идентификаторы, операции

1	2	3	4	5
1, 2, 5, 7, 8	2, 3, 5, 6	1, 3, 4, 6	1, 5, 7, 8	2, 4, 7
6	7	8	9	10
div	mod	CFAEBD	BDEFAC	2, 3
11	12	13	14	15
1, 4	2, 3, 4	1, 4	2, 4	1, 4

Стандартные функции

1	2	3	4
1, 3, 4, 6	2, 3, 5	3	1
5	6	7	8
2	4	CDBA	CFAEBD
9	10	11	
CEFADB	CABDBBA	CFGBDEA	

Простые типы данных. Описание переменных

1	2	3	4	5
3, 5, 6, 8	2, 5, 6	1, 2	2, 3, 4, 7	1, 2, 5, 6
6	7	8	9	10
FDBEAC	FADCBE	BAABBAAA	1, 3, 6	1, 5, 6
11	12	13	14	15
2, 4, 6	1, 2, 3, 5	3, 4, 6	3, 5, 6	4, 6
16	17	18	19	20
2, 5, 6	1, 5	3, 4	1, 3, 4	3, 6

Ввод-вывод, присваивание

1	2	3	4
4	2	1	3
5	6	7	8
4	ABAABBAB	DAFBCE	CFBEAD
9	10	11	12
BBFACDEC	2, 3, 5, 7	1, 6, 7	1, 2, 6
13	14	15	16
3, 4, 5, 6	2, 4, 5, 6	1, 3, 6	2, 3
17	18	19	20
1, 2	2, 4	1, 4	6 2
21	22	23	
3 3	6 7	3 11	

Операторы ветвления

1	2	3	4	5	6
2, 3	3, 5	-3	-8	2	15
7	8	9	10	11	12
-10	3	15	7	8	3
13	14	15	16	17	18
1	1	2	4	0	24
19	20	21	22	23	
9	1, 2	1	2, 3	2	

Операторы цикла

1	2	3	4
3	2	4	3 2 1 4
5	6	7	8
3 2 4 1	1, 2	2, 4	1, 3
9	10	11	12
3, 4	1, 2	3, 4	120
13	14	15	16
-13	24	-3	24
17	18	19	20
5 7 2 6 1 4 8 3	4 2 6 7 1 5 8 3	6 2 1 4 5 9 7 3 8	4 2 7 6 5 1 8 3
21	22		
4 6 1 7 2 8 3 5	4 6 2 1 3 5		

Структурированные типы данных

1	2	3	4
2, 3	2, 3, 6	1, 3, 4	1, 2, 5
5	6	7	8
1, 5	3, 4, 5	запись	множество
9	10	11	12
массив	строка	файл	end
13	14	15	16
1, 2	3, 4	1, 3	2, 3

Массивы

1	2	3	4
1, 4, 6	3, 4, 5	1, 5, 6	1, 3, 5
5	6	7	8
3	4	2	3
9	10	11	12
3	3	2	2
13		14	
3 7 6 9 1 5 2 4 8		6 9 5 2 8 1 1 0 3 7 4	
15		16	
6 9 5 2 8 1 1 0 3 1 1 7 4		4 7 2 6 8 1 9 5	

Двумерные массивы

1	2	3	4	5
3	11	18	2	3
6	7	8	9	10
1	4	10	12	1
11	12	13	14	15
2	3	4	1	2
16	17	18	19	20
3	4	1	2	4

Строки, множества

1	2	3	4
CDBA	ACBCA	4	4
5	6	7	8
2	1	3	4
9	10	11	12
4	5	1	3
13		14	
7 4 (4 7) 9 3 8 1 10 5 2 6		3 7 (7 3) 6 10 2 11 9 1 12 5 8 4	
15		16	
5 9 (9 5) 6 12 3 8 11 2 13 1 10 4 7		7 4 (4 7) 8 10 3 1 9 5 2 6	

Файлы

1	2	3	4
1, 4, 5	1, 3, 4	2, 4, 5	1, 2, 5
5	6	7	8
1, 2, 4	4	DFBCAE	3, 4

Подпрограммы

1	2	3	4
3, 4	1, 2	2, 3	1, 4
5	6	7	8
1, 4	7	14	12
9	10	11	12
6	17	real	boolean

Список рекомендуемой литературы

1. Аляев Ю. А., Гладков В.П., Козлов О. А. Практикум по алгоритмизации и программированию на языке Паскаль. — М.: Финансы и статистика, 2007. — 528 с.
2. Демидов Д. В. Основы программирования в примерах на языке Паскаль: Учебное пособие. — М.: НИЯУ МИФИ, 2010. — 172 с.
3. Долинер Л. И. Основы программирования в среде PascalABC.NET: учебное пособие. — Екатеринбург: Издательство УрГУ, 2014. — 129 с.
4. Комлев Н. Ю. Самоучитель игры на Паскале. ABC и немного Turbo. — М.: СОЛОН-ПРЕСС, 2013. — 256 с.
5. Мишенин А. И. Сборник задач по программированию: учеб. пособие. — М.: Финансы и статистика; ИНФРА-М, 2009. — 224 с.
6. Осипов А. В. PascalABC.NET. Введение в современное программирование. — Ростов-на-Дону, 2019. — 572 с. / Интернет-ресурс: <http://www.pascalabc.net/downloads/OsipovBook/ModernProgr.pdf>
7. Осипов А. В. PascalABC.NET: выбор школьника. Часть 1. — Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2020. — 152 с. / Интернет-ресурс: <http://www.pascalabc.net/downloads/OsipovBook/StudentChoice.pdf>
8. Рубанцев В. Программирование графики на примерах. — Интернет-ресурс: <http://www.pascalabc.net/downloads/Books/Rubantsev/GraphicsByExample.pdf>
9. Рубанцев В. Программирование на паскале. Графика для компьютерных игр. — Интернет-ресурс: <http://www.pascalabc.net/downloads/Books/Rubantsev/SFML.pdf>
10. Семакин И.Г., Шестаков А. П. Основы алгоритмизации и программирования, 3-е изд. — М.: Издательский центр «Академия», 2012. — 400 с.
11. Фаронов В.В. Turbo Pascal: Учебное пособие. — СПб.: Питер, 2009. — 367 с.

Книги Осипова А. В. и Рубанцева В. можно найти на официальном сайте проекта PascalABC.NET: www.pascalabc.net. Там же можно бесплатно скачать дистрибутив среды программирования PascalABC.NET или работать в режиме on-line, посмотреть и/или скачать презентации, статьи, примеры и много другой полезной информации по языку программирования PascalABC.NET.

Савина Елена Владимировна

**Практикум по программированию
на PascalABC.NET**

Текст приводится в авторской редакции

Ответственный редактор *С. Краснова*
Верстальщик *М. Глаголева*

