

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 предопределенные 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) $\text{sqr}(\cos(x))/(x - 1)**(1/5)$;

2) $\sin(\text{Pi}/6)*\cos(\text{Pi}/4)$;

3) $(\text{power}(a, 9) + 4)**0.1$.

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 :

$m := k \text{ div } 100$.

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

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

2) $m := k \text{ div } 1000 + k \text{ mod } 100 \text{ div } 6$

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) $\text{trunc}(-3,8) + \text{round}(-2,7) + \text{abs}(-2)$

2) $5*14 \text{ div } 7 \text{ mod } 7 - \text{trunc}(\sin(2))$

3) $\text{succ}(\text{round}(17/3)) - \text{pred}(-2)$

Решение. 1) $\text{trunc}(-3,8) = -3$; $\text{round}(-2,7) = -3$; $\text{abs}(-2) = 2$; **ответ: -4**;

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

$\text{trunc}(\sin(2)) = 0$; **ответ: 3**;

3) $\text{round}(17/3) = 6$; $\text{succ}(6) = 7$; $\text{pred}(-2) = -3$; $7 - (-3) = 10$; **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) $a \text{ and } \text{false} \equiv \text{false}$ |
| 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 $(a \text{ and } \text{not } b) \text{ or } c$, 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) $\text{not } (x * x + \text{abs}(x) < 0) \text{ and } \text{true}$
- 4) $(1/x > x) \text{ and } \text{not } (1 + x * x > 0) \text{ or } (132 < 13.2 * 10) \text{ or } (x * x - 2 * x + 1) < 0)$
- 5) $\text{not } (12.5 > 25/2) \text{ and } (x < x * x) \text{ and } \text{not } (-5 * 6 = (7.5 * (-4)))$
- 6) $(a \text{ or } (\text{not } a)) \text{ and } \text{odd}(345) \text{ or } (\text{sqr}(x) + 3 \leq 4)$

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru