

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДВОРЕЦ ДЕТСКОГО (ЮНОШЕСКОГО) ТВОРЧЕСТВА ВЫБОРГСКОГО РАЙОНА
САНКТ-ПЕТЕРБУРГА
(ГБУ ДО ДДОТ)

Практические занятия
Модуль «Программирование на языке Pascal»

«Основы компьютерных технологий»

Второй год обучения
Группа № 1
Возраст учащихся 9 — 12 лет

Ухтина Елена Николаевна,
педагог дополнительного образования

Санкт-Петербург
2016 / 2017 учебный год

Цель занятий: познакомить учащихся с линейным алгоритмом

Задачи:

Ⓢ *Обучающие:*

Ⓢ сформировать у учащихся первичные знания по применению изученного материала.

Ⓢ *Развивающие:*

Ⓢ учить анализировать, обобщать и систематизировать;

Ⓢ обогащать словарный запас учащихся.

Ⓢ *Воспитательные:*

Ⓢ развивать информационную культуру учащихся, способность к самостоятельной и коллективной деятельности, рефлексия.

Тип занятия: изучения и первичного закрепления новых знаний.

Оборудование: мультимедийный проектор, компьютеры с установленной средой программирования Pascal, настроенная локальная сеть, конспект для учащегося раздаточные карточки-задания.

Занятие 1. Линейные алгоритмы. Обмен значений численных переменных

Задача

Пользователь вводит два числа. Одно присваивается одной переменной, а второе - другой. Необходимо поменять значения переменных так, чтобы значение первой оказалось во второй, а второй - в первой.

Пояснение к задаче и алгоритм решения

Существует два способа обменять значения переменных. Первый подходит для любых типов данных, а второй - только для численных.

1-й способ. Через "буферную" переменную

1. Пусть первое введенное значение присвоено переменной a , второе - b .
2. Присвоим некоей третьей переменной (например, c) значение хранимое в a . Таким образом окажется, что одно и то же значение будет сохранено в двух переменных.
3. Запишем значение переменной b в a . Таким образом, мы "затрем" прежнее содержимое a , а значение b будет храниться в двух переменных.
4. Однако значение a не потеряно, т.к. заранее было сохранено в c . Теперь мы можем присвоить переменной b значение хранимое в c .

В результате данных трех присвоений окажется, что в переменной a будет значение b , в b - значение a .

2-й способ. Путем арифметических операций

1. Запишем в переменную a сумму значений двух переменных: a и b .
2. Переменной b присвоить разность между новым значением переменной a и значением b . Таким образом в b окажется старое значение a .
3. Переменной a присвоить разность между ее текущим значением и новым значением переменной b . В результате в a окажется старое значение b .

Исходный код на языке программирования Pascal

// 1-й способ:

```
var a,b,c: integer;
begin
  readln(a);
  readln(b);
  writeln('a=',a,'; b=',b);
  c := a;
  a := b;
  b := c;
  writeln('a=',a,'; b=',b);
end.
```

// 2-й способ:

```
var a,b: integer;
```

```
begin  
readln(a);  
readln(b);  
writeln('a=',a,'; b=',b);  
a := a + b;  
b := a - b;  
a := a - b;  
writeln('a=',a,'; b=',b);  
end.
```

Пример(ы) выполнения программы на языке Pascal

5

6

a=5; b=6

a=6; b=5

Задача

Продемонстрировать возможности

- ⓐ вывода данных в табличной форме (с выровненными столбцами),
- ⓐ выравнивания слева,
- ⓐ вывода вещественных чисел с заданным количеством знаков после запятой.

Пояснение к задаче и алгоритм решения

В языках программирования, имеющих практическое значение, предусмотрены специальные синтаксические конструкции, позволяющие выводить данные в поля заданной ширины, выравнивать их по разным краям поля, а также выводить вещественные числа в различных форматах.

В языках КуМир и Basic-256 подобных языковых возможностей найти не удалось. Если их действительно нет, то видимо потому, что данные языки являются исключительно учебными.

Исходный код на языке программирования Pascal

```
var
m, n: real;
s1, s2: string[10];
begin
m := 1.45678;
n := -0.23451;
s1 := 'Hello';
s2 := 'world';
writeln(m:10:2, n:10:2);
writeln(s1:10, s2:10);
writeln;
write(s1:-9, s2);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
1.46 -0.23
Hello world
Helloworld
```

Особенности решения на языке программирования Pascal

В Pascal при выводе в поля заданной ширины следует указать через двоеточие после переменной (или непосредственно числа или строки) количество знакомест. При выводе вещественных чисел после указания количества знакомест также через двоеточие указывается количество знаков после запятой. Они не являются добавочными к количеству знакомест, а включены в это количество. Если для вещественного числа не указать количество цифр после запятой, то число будет выведено в экспоненциальной форме. Для выравнивания по левому краю, по аналогии с другими языками программирования, следует использовать минус перед числом-количеством знакомест. Однако для FreePascal это не сработало. Хотя компилятор не сообщил ни о какой ошибке.

Занятие 3. Линейные алгоритмы. Вычисление площадей и периметров фигур

Задача

Вычислить площадь и периметр

- ⓐ треугольника по данным трем сторонам,
- ⓑ прямоугольника по данным ширине и высоте,
- ⓒ круга по заданному радиусу.

Длины сторон и радиус вводятся пользователем.

Пояснение к задаче и алгоритм решения

Площади и периметры фигур можно найти по следующим формулам.

Периметр треугольника: $P = a + b + c$

Площадь треугольника: $S = \sqrt{p(p-a)(p-b)(p-c)}$, где $p = P/2$

Периметр прямоугольника: $P = 2(a + b)$

Площадь прямоугольника: $S = ab$

Периметр круга: $P = 2\pi r$

Площадь круга: $S = \pi r^2$

Исходный код на языке программирования Pascal

```
var
  a,b,c: word;
  w,h: word;
  r: word;
  P,S: real;
begin
  readln(a,b,c);
  P := (a + b + c) / 2;
  S := sqrt(P * (P - a) * (P - b) * (P - c));
  writeln('P=',a+b+c,'; S=',S:5:2);
  readln(w,h);
  writeln('P=',(w+h)*2,'; S=',w*h);
  readln(r);
  P := 2 * pi * r;
  S := pi * r * r;
  writeln('P=',P:5:2,'; S=',S:5:2);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
5 3 5
P=13; S= 7.15
3 9
P=24; S=27
5
P=31.42; S=78.54
```

Занятие 4. Линейные алгоритмы. Сумма и произведение цифр числа

Задача

Найти сумму и произведение цифр трехзначного числа, которое вводит пользователь.

Пояснение к задаче и алгоритм решения

Если трехзначное число нацело разделить на 100, то получим число сотен, т.е. первую цифру этого числа.

Если найти остаток от деления на 10 любого целого числа, то этот остаток покажет число единиц в этом числе. Таким образом извлекается последняя цифра числа.

Чтобы извлечь среднюю цифру трехзначного числа (показывающую количество десятков), можно выполнить два действия:

1. Разделить число нацело на 10. Тем самым избавиться от последней цифры трехзначного числа, превратив его в двузначное.
2. Найти остаток от деления полученного двузначного числа на 10. Этот остаток и будет средняя цифра первоначального трехзначного числа.

Извлекая вышеописанным способом цифры числа их надо присвоить переменным. Т.к. цифр три, то и переменных будет три. После этого выполнить сложение и умножение значений переменных.

Исходный код на языке программирования Pascal

```
var  
n: integer;  
a,b,c: byte;  
begin  
  readln(n);  
  a := n div 100;  
  b := n div 10 mod 10;  
  c := n mod 10;  
  writeln('Сумма: ',a+b+c);  
  writeln('Произведение: ',a*b*c);  
end.
```

Задача

Выполнить логические побитовые операции "И", "ИЛИ" и др. над числами 5 и 6. Выполнить над числом 5 побитовый сдвиг вправо и влево на два знака. Объяснить полученный результат.

Пояснение к задаче и алгоритм решения

Обычно в языках программирования предусмотрены так называемые логические побитовые операции. Они выполняются не над самими числами, а над их двоичным представлением. Например, число 5 в двоичной системе счисления выражается как 101, а число 6 - как 110. Выполняя логическую побитовую операцию "И" получим число 4, т.к. в младшем разряде числа 5 стоит 1, а числа 6 - 0. Выражение "1 и 0" возвращает 0. Продолжая поразрядно выполнять логическое "И" в среднем разряде получим 0, а в старшем 1. Можно записать так:

101

110

100

Двоичное число 100 - это десятичное число 4.

Выполним операцию побитового логического "ИЛИ":

101

110

111 - это число 7.

"Исключающее ИЛИ":

101

110

011 - это число 3.

При сдвиге биты просто сдвигаются на указанное количество ячеек в освободившиеся ячейки дописываются нули или единицы (это зависит от ряда причин):

$110 \ll 2 = 11000$ (число 24),

$110 \gg 2 = 001$ (число 1).

Исходный код на языке программирования Pascal

```
Var a, b: byte;  
begin  
a := 5;  
b := 6;  
writeln('a=', a);  
writeln('b=', b);  
writeln('a and b = ', a and b);  
writeln('a or b = ', a or b);  
writeln('a xor b = ', a xor b);  
writeln('b shl 2 = ', b shl 2);  
writeln('b shr 2 = ', b shr 2);  
end.
```

Пример(ы) выполнения программы на языке Pascal

a=5

b=6

a and b = 4
a or b = 7
a xor b = 3
b shl 2 = 24
b shr 2 = 1

Особенности решения на языке программирования Pascal

В Паскале обычные логические операции и побитовые обозначаются одинаково. Первые применяются, если операнды имеют булевый тип (они получаются в результате простых логических операций $>$, $<$ и др.), а вторые - если операнды представляют собой целые числа (неважно какого типа).

Занятие 6. Линейные алгоритмы. Вывести уравнение прямой по координатам двух точек

Задача

По введенным пользователем координатам двух точек вывести уравнение прямой, проходящей через эти точки.

Пояснение к задаче и алгоритм решения

Общее уравнение прямой имеет вид $y = kx + b$. Для какой-то конкретной прямой в уравнении коэффициенты k и b заменяются на числа, например, $y = 4x - 2$. Задача сводится именно к нахождению этих коэффициентов.

Так как координаты точки это значения x и y , то мы имеем два уравнения. Пусть, например, координаты точки А(3;2), а координаты В(-1;-1). Получаем уравнения:

$$2 = k*3 + b,$$

$$-1 = k*(-1) + b.$$

Решая полученную систему уравнений находим значения k и b :

$$b = 2 - 3k$$

$$-1 = -k + 2 - 3k$$

$$4k = 3$$

$$k = 3/4 = 0.75$$

$$b = 2 - 3 * 0.75 = 2 - 2.25 = -0.25$$

Таким образом, получается уравнение конкретной прямой, проходящей через указанные точки: $y = 0.75x - 0.25$.

Вывод общих выражений для вычисления b и k :

$$/ y_1 = kx_1 + b$$

$$/ y_2 = kx_2 + b$$

$$b = y_2 - kx_2$$

$$y_1 = kx_1 + y_2 - kx_2$$

$$k = (y_1 - y_2) / (x_1 - x_2)$$

Алгоритм решения данной задаче на языке программирования будет таков:

1. Получить значения координат первой точки и присвоить их переменным, например $x1$ и $y1$.
2. Получить значения координат $(x2, y2)$ второй точки.
3. Вычислить значение k по формуле $k = (y1 - y2) / (x1 - x2)$.
4. Вычислить значение b по формуле $b = y2 - k * x2$.
5. Вывести на экран полученное уравнение.

Исходный код на языке программирования Pascal

var

$x1, y1, x2, y2$: **real**;

k, b : **real**;

begin

write('A(x1;y1): '); **readln**($x1, y1$);

write('B(x2;y2): '); **readln**($x2, y2$);

$k := (y1 - y2) / (x1 - x2)$;

$b := y2 - k * x2$;

writeln('y = ', k :0:2, 'x + ', b :0:2);

end.

Пример(ы) выполнения программы на языке Pascal

A(x1;y1):

1.2

5.6

B(x2;y2):

-3.45 8.2

$y = -0.56x + 6.27$

Занятие 7. Линейные алгоритмы. Найти длину гипотенузы

Задача

По двум введенным пользователем катетам вычислить длину гипотенузы.

Пояснение к задаче и алгоритм решения

Катеты и гипотенуза - это стороны прямоугольного треугольника. Если известны длины катетов, то длина гипотенузы находится по теореме Пифагора:

"Квадрат гипотенузы равен сумме квадратов катетов"

С помощью формулы это выражается так:

$$c^2 = a^2 + b^2$$

Отсюда следует, что длина гипотенузы равна квадратному корню из суммы квадратов катетов:

$$c = \text{sqrt}(a^2 + b^2)$$

(sqrt() - обозначение функции извлечения корня).

Обычно в языках программирования предусмотрен оператор возведения в степень. Например, в языке программирования Python он обозначается двумя звездочками (**), а в Basic знаком ^. Однако в Pascal нет операции возведения в степень.

Для извлечения корня обычно существует специальная функция, а не оператор.

Исходный код на языке программирования Pascal

```
var
  a,b,c: real;
begin
  write('a=');
  readln(a);
  write('b=');
  readln(b);
  c := sqrt(a*a + b*b);
  writeln('c=',c:1:2);
end.
```

Пример(ы) выполнения программы на языке Pascal

```
a=3
b=4
c=5.00
```

Занятие 8. Линейные алгоритмы. Количество символов между двумя буквами алфавита. Определение буквы по ее номеру в алфавите

Задача

1. Пользователь вводит две буквы. Определить, на каких местах алфавита они стоят, и сколько между ними находится букв.
2. Пользователь вводит номер буквы в алфавите. Определить, какая это буква.

Пояснение к задаче и алгоритм решения

Чтобы определить позицию буквы в алфавите, надо знать по используемой таблице символов коды первой буквы алфавита и той, позиция которой определяется. Если вычесть из кода искомой буквы код первой и прибавить 1, то получим как раз номер буквы в алфавите. Почему нужно прибавлять 1? Представьте, что ищется позиция буквы 'b'. Если из ее кода вычесть код буквы 'a', то получится 1, но 'b' - это вторая буква.

Для того, чтобы определить количество символов между двумя буквами, надо найти разность их кодов или позиций в алфавите и вычесть из полученного значения 1.

Поиск буквы по ее номеру в алфавите также опирается на знание кода первой буквы. К коду первой буквы прибавляется номер в алфавите искомой буквы за вычетом 1.

Нижеприведенные программы правильно работают только с английскими строчными (маленькими) буквами.

Исходный код на языке программирования Pascal

```
var
  a,b: char;
  an, bn, n: byte;
begin
  write('Буква 1: '); readln(a);
  write('Буква 2: '); readln(b);
  an := ord(a);
  bn := ord(b);
  writeln('Позиция 1: ', an-ord('a')+1);
  writeln('Позиция 2: ', bn-ord('a')+1);
  n := abs(bn-an)-1;
  writeln('Количество букв между ними: ', n);
  writeln;
  write('Позиция буквы в алфавите: '); readln(n);
  n := n+ord('a')-1;
  writeln('Это буква - ', chr(n));
end.
```

Пример(ы) выполнения программы на языке Pascal

```
Буква 1: d
Буква 2: a
Позиция 1: 4
Позиция 2: 1
Количество букв между ними: 2 Позиция буквы в алфавите: 20 Это буква - t
```

Занятие 9. Линейные алгоритмы. Рассчитать выплаты по кредиту.

Задача

Рассчитать месячные выплаты (m) и суммарную выплату (s) по кредиту.

О кредите известно, что он составляет n рублей, берется на u лет, под p процентов.

Пояснение к задаче и алгоритм решения

Месячные выплаты находятся по формуле:

$m = (n * p * (1 + p)^y) / (12 * ((1 + p)^y - 1))$, где p выражается в долях единицы, а не процентах.

Суммарная выплата представляет собой выплаты за все месяцы каждого года:

$$s = (m * 12) * y$$

Исходный код на языке программирования Pascal

```
var
m, s, p, n, r, a: real;
begin
write('Величина заема : ');
readln(s);
write('Процент: ');
readln(p);
write('Количество лет: ');
readln(n);
r := p / 100;
a := exp(ln(1 + r) * n);
m := (s * r * a) / (12 * (a - 1));
m := trunc(100 * m + 0.5) / 100;
writeln;
writeln('Месячная выплата: ', m:5:2);
writeln('Прибыль банка: ', m * 12 * n - s:5:2);
readln
end.
```

Пример(ы) выполнения программы на языке Pascal

Величина заема: 1000000

Процент: 15

Количество лет: 20

Месячная выплата: 13313.46

Прибыль банка: 2195230.40

Особенности решения на языке программирования Pascal

Выражение $\exp(\ln(1 + r) * n)$ вычисляет $(1 + r)^n$.

Выражение $\text{trunc}(100 * m + 0.5) / 100$ выполняет округление до копейки.