

ПЕРЕМЕННЫЕ ВЕЛИЧИНЫ. ВВОД ДАННЫХ В ПРОГРАММУ. ОПЕРАТОР ПРИСВАИВАНИЯ

Туменбай Онгар Айдосулы
СОШ № 51 им. Т. Рыскулова

Аннотация: Переменные величины (или переменные) – величины, которые при выполнении программы могут принимать различные значения. Эти значения (в любой момент времени – какое-то одно) хранятся в памяти компьютера и могут быть использованы, поэтому говорят, что переменные используются в программе для хранения информации.

Ключевые слова: величины, ввод данных, программа, оператор, присваивание.

VARIABLES. ENTERING DATA INTO THE PROGRAM. ASSIGNMENT OPERATOR

Tumenbay Ongar Aydosuly

Annotation: Variables (or variables) are quantities that can take different values when executing a program. These values (at any given time – some one) are stored in the computer's memory and can be used, so they say that variables are used in the program to store information.

Keywords: values, data entry, program, operator, assignment.

При использовании переменных величин пользователь должен ввести размеры сторон прямоугольника, которые могут быть любыми; эти значения будут использованы для расчета площади. Значение площади, которое тоже может быть любым, надо запомнить, а затем вывести на экран. Можем сказать, что в этой программе используются три переменные.

Каждая переменная величина характеризуется именем и типом. В именах переменных можно использовать латинские буквы, цифры (но имя не может начинаться с цифры) и знак подчеркивания «_». Примеры: n,

Есть набор слов, которые нельзя использовать в качестве имен переменных, так как эти слова «зарезервированы» в языке для определенных

целей (эти слова называют «зарезервированными», или «служебными», или «ключевыми»). С перечнем таких слов в используемом вами языке программирования ознакомьтесь самостоятельно.

Основные типы данных приведены в таблице 1.

Таблица 1

Тип	К нему относятся	Примеры	Примечание
Целый	Целые числа (положительные и отрицательные, а также 0)	4, -45, 0, 687	
Вещественный	Вещественные числа (могут быть с дробной частью)	1.45, 0.00453, -3.789,	Видно, что разделителем целой и дробной частей вещественного числа является точка
Логический	Величины, которые могут принимать значения True (Истина, да) или False (Ложь, нет)		Величины логического типа возможны не во всех языках программирования
Строковый	Последовательность (строка) символов, в том числе один символ или пустая строка (""/"")	"Школа", 'красный', "ф", ""	Вид кавычек зависит от используемой системы программирования

Тип величины определяет:

- какие значения может принимать величина (область допустимых значений переменной), например, значением переменной строкового типа может быть только строка (последовательность) символов;

- какие операции можно проводить над переменной (множество допустимых операций с величиной); например, над величинами вещественного типа можно проводить операции сложения, вычитания, умножения, деления и возведения в степень, а над величинами целого типа – еще две операции (см. далее);

- какой объем памяти компьютера требуется для хранения значения данной переменной и в каком формате будут храниться данные. Например, переменные вещественного типа, как правило, занимают 8 байтов.

Значение переменной хранится в каком-то месте памяти, которое можно смоделировать в виде прямоугольника (ячейки), рядом с которым указано имя этой переменной:

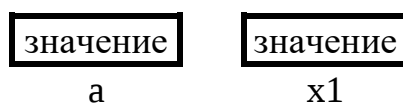


Рис. 1

Как значение «попадает» в соответствующий прямоугольник?

Это происходит с помощью двух операторов:

- 1) оператора присваивания;
- 2) оператора ввода данных.

Начнем со второго. Он используется для ввода данных в программу в ходе ее выполнения с помощью клавиатуры (как говорят в программировании – «с клавиатуры»).

Таблица 2

Язык программирования	Оператор
Школьный алгоритмический язык	ввод
Паскаль	readln, read
Python	input()

Например, чтобы ввести значение переменной *a* строкового типа, нужно записать в программе:

Таблица 3

Язык программирования	Оператор
Школьный алгоритмический язык	ввод a
Паскаль	readln(a)
Python	a = input()

При выполнении этой строки на экране появится курсор, и система будет ожидать ввода значения с клавиатуры. Когда пользователь введет его и нажмет клавишу **<Enter>**, система запишет это значение в память в переменную *a*. До нажатия клавиши **<Enter>** можно удалять символы с помощью клавиши **<Backspace>**.

В программах на школьном алгоритмическом языке и языке Паскаль ввод значений переменных *числового* типа проводится с помощью операторов, указанных в таблице чуть выше. В языке Python имеется особенность выполнения оператора `input`. Дело в том, что при нажатии клавиши на клавиатуре (в том числе клавиш с цифрами) в компьютер поступает код клавиши, то есть код соответствующего символа.

И входные данные воспринимаются оператором `input` именно как последовательность символов, в результате чего переменная получит значение строкового типа, даже если была введена последовательность цифр. Чтобы преобразовать полученное значение в целое число, оператор должен быть оформлен так:

`a = int(input())`

`a` в вещественное число:

`a = float(input())`

Во многих языках программирования все используемые в программе переменные величины должны быть предварительно объявлены (описаны) с указанием их типа (см. программу Расчет площади в главе 1). С особенностями объявления переменных в используемом вами языке программирования ознакомьтесь самостоятельно.

Когда приходится вводить значения нескольких величин, целесообразно перед оператором ввода записать оператор вывода на экран с сообщением-подсказкой, чтобы пользователь программы знал, какое значение вводится в тот или иной момент ее выполнения:

Таблица 4

Язык программирования	Оператор
Школьный алгоритмический язык	вывод нс , "Введите фамилию: " ввод фам вывод нс , "Укажите имя: " ввод им
Паскаль	<code>write('Введите фамилию: '); readln(fam);</code> <code>write('Укажите имя: '); readln(im);</code>
Python	<code>print("Введите фамилию") fam = input()</code> <code>print("Укажите имя: ") im = input()</code> или <code>fam = input("Введите фамилию: ") im = input("Укажите имя: ")</code>

Можно также вывести на экран общее сообщение:

Таблица 5

Язык программирования	Оператор
Школьный алгоритмический язык	вывод нс , "Введите фамилию, а затем имя " ввод фам ввод им
Паскаль	<code>write('Введите фамилию, а затем имя '); readln(fam);</code> <code>readln(im);</code>
Python	<code>print("Введите фамилию, а затем имя: ") fam = input()</code> <code>im = input()</code>

и (в некоторых языках программирования) указывать в операторе ввода два (и более) имени переменных:

Таблица 6

Язык программирования	Оператор
Школьный алгоритмический язык	вывод нс, "Введите фамилию, а затем имя "
Паскаль	вывод фам, им write('Введите фамилию, а затем имя '); readln(fam, im);

Примечание. Не выполняя программу (на языке программирования, который вы изучаете), определите, как будут выводиться приведенные в ней сообщения.

Теперь об операторе присваивания. Этот оператор позволяет изменить (или задать впервые) значение переменной величины.

Например, когда присваивается значение величине числового типа, оператор имеет вид:

– в школьном алгоритмическом языке и языке Паскаль:

<имя переменной> := <арифметическое выражение>

– в языке Python:

<имя переменной> = <арифметическое выражение>

где *<арифметическое выражение>* – одно или несколько чисел, имен переменных величин или имен функций, соединенных знаками арифметических операций. Примеры:

Таблица 7

Школьный алгоритмический язык, Паскаль	Python
c := 12 m := n сумма := a + b x := 2 * a – 3.6 * b/c sred := (a + b)/2	c = 12 m = n sum = a + b stepen3 = a ** 3 x = 2 * a – 3.6 * b/c sred = (a + b)/2

Обратим внимание на знак деления («/»), а также на использование в последнем примере круглых скобок для изменения порядка действий (во всех языках программирования выражения записываются в строку, без «многоэтажных» дробей).

Распространенной ошибкой начинающих программистов является запись операторов присваивания в виде:

c := 2a

или

c = 2a

(правильный вариант: $c := 2 * a$ или $c = 2 * a$; вариант оператора $c := a2$ или $c = a2$

возможен, если $a2$ – имя переменной, значение которой задано).

Когда в операторе присваивания записаны знаки нескольких операций, то они (операции) выполняются в определенном порядке. При определении порядка действий учитывается приоритет (старшинство) операций. Они выполняются в следующем порядке:

- операции в скобках;
- возведение в степень, справа налево (!);
- умножение (*) и деление (/), слева направо;
- сложение и вычитание, слева направо.

Таким образом, умножение и деление имеют одинаковый приоритет, более высокий, чем сложение и вычитание. Поэтому в последнем приведенном примере ($sred := (a + b)/2$) запись без скобок привела бы к тому, что сначала выполнилось бы деление значения b на 2 , а затем сложение этого частного и значения переменной a .

Над величинами целого типа, кроме операций сложения, вычитания, умножения, деления и возведения в степень, можно выполнять также еще две операции:

1) операция определения целой части частного («целочисленного частного») от деления одного целого числа на другое (её называют «целочисленное деление», или «деление нацело»);

2) операция определения остатка от целочисленного деления.

Это сделано потому, что задачи, связанные с использованием этих операций, очень часто встречаются в программировании.

Знаки этих двух операций показаны в таблице:

Таблица 8

Язык программирования	Целочисленное деление	Определение остатка	Примеры
Паскаль	div	mod	$ost := a \bmod 2$ $a := b \operatorname{div} (c + 1)$
Python	//	%	$ch = n // d$ $ost = 1000 \% m$

Они имеют такой же приоритет, как и «обычные» умножение и деление.

В школьном алгоритмическом языке для определения целочисленного частного от деления одной величины на другую и остатка от такого деления используются не знаки операций, а функции `div` и `mod`, соответственно:

частн := div(a, b) ост := mod(a, 10)

Как работает оператор присваивания? Объясним на примере:

средн := (a + b)/2

Транслятор (см. главу 1) «посмотрит», что «лежит» в ячейках a и b, сложит соответствующие значения и сумму разделит на 2. Результат «положится» в ячейку средн:

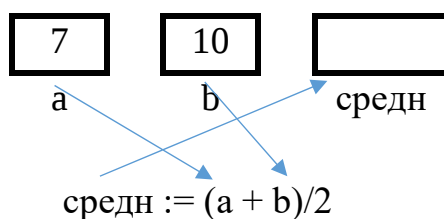


Рис. 2.

Это означает, что в программе может быть запись:

$x := x + 10$

невозможная в математике. Ее смысл в том, что нужно к имеющемуся в данный момент выполнения программы значению переменной x прибавить 10, а результат присвоить той же переменной.

Мы рассмотрели особенности присваивания значения величинам числового типа.

Заключение

Завершая обсуждение переменных величин, заметим, что в программах могут использоваться также так называемые «константы» – величины, которые в ходе выполнения программы не меняют заданное им единожды значение. Константы может создать (описать) программист, и, кроме того, во многих языках программирования имеются стандартные (предусмотренные в системе программирования).

© О.А. Туменбай, 2022