

Переменные. Операторы

Лекция 2

Переменные

- это именованные области компьютерной памяти, в которых можно хранить различные значения.

- В языке Си, прежде чем использовать переменную, ее надо *объявить*.

Объявить нужно до того, как переменная будет использована!

- Объявления выглядит следующим образом:

```
тип имя1, имя2;
```

- Пример:

```
int age;
```

Имя переменной может начинаться с буквы или символа нижнего подчеркивания и состоять из любой комбинации букв, цифр и символа нижнего подчеркивания

Функция scanf_s

- это функция для ввода информации с клавиатуры

Пример:

```
scanf_s("%d", &age);
```

Первый параметр этой функции – форматная строка, которая описывает, что и какого типа нужно считать с клавиатуры.

%d означает, что надо считать одно десятичное число (decimal).

Второй параметр – имя переменной, куда функция должна сохранить введенное с клавиатуры значение, со специальным символом &.

Функция scanf_s. Пример

```
#include <stdio.h>
#include <locale.h>
int main(){
    setlocale(LC_ALL, "Russian");
    int age, my_age;
    printf("Сколько вам лет?\n");
    scanf_s("%d", &age);

    my_age = age + 2;
    printf("Ваш возраст: %d. Мой возраст – %d!\n", age, my_age);
    return 0;
}
```

Константа

- (лат. *constanta* – постоянная, неизмен-ная) – это некоторое конкретное значение (числовое или строковое), которое, будучи определенным в программе, меняться уже не может.
- Константы бывают двух типов: *литеральные* и *символические*.

Литеральные константы

— это конкретные значения, записанные непосредственно в тексте программы.

Типы:

- числовые (целочисленными и дробными), например: 0, -5, 3.14, 103
- строковые (заключенными в двойные кавычки) — "Hello, world!\n", "Input a number: "
- символьные (заключенными в одинарные кавычки) — 'a', 'b', '0'

Системы счисления

- Целочисленные константы в языке Си могут быть представлены в одной из следующих систем счисления:
 - десятичные: последовательность цифр от 0 до 9, которая начинается с цифры, отличной от 0 (исключение составляет лишь сам 0). Пример: 2, -45, 23;
 - восьмеричные: последовательность цифр от 0 до 7, которая всегда начинается с нуля. Пример: 00, 032, 063;
 - шестнадцатеричные: последовательность цифр от 0 до 9 и символов от A до F (без учета регистра), которая всегда начинается с комбинации 0x. Пример: 0x0, 0x4, 0xFFFF, 0x2A4, 0x7fff.

Символические константы

— это своего рода «переменные», значения которых меняться не могут.

- Их также необходимо объявлять в коде программы, но перед типом надо ставить ключевое слово `const`. Например:

```
const double pi = 3.14;
```

Очень важно помнить!!!

Присваивать значение символической константе необходимо *одновременно с ее объявлением*.

```
const double pi;
```

```
pi = 3.14;
```

<- Ошибка

Типы данных

- char – символ,
- int – целое число,
- float – дробное число одинарной точности,
- double – дробное число двойной точности.

Общий вид создания переменной:

тип имя1 [= значение1], ..., имяN [= значениеN];

Для записи в переменную какого-либо значения, используется *оператор присваивания* (=).
Этот оператор работает следующим образом:

куда = что;

Пример. Программа для вычисления объема бассейна

```
#include <stdio.h>
#include <locale.h>
int main(){
    setlocale(LC_ALL, "Russian");
    const double pi = 3.14;
    double V, r, h;
    printf("Введите значения r и h через пробел: ");
    scanf_s("%lg %lg", &r, &h);
    V = pi * r * r * h;
    printf("Объем бассейна равен %lg\n", V);
    return 0;
}
```

Выражения

— это основная рабочая сила языка Си.

Выражения бывают двух типов:

- математические
- логические.

Математические выражения

— это совокупность имен переменных, символических и литеральных констант, скобок и знаков математических операций (+, -, *, /, %) порядке приоритетов операций.

Помимо перечисленных основных операций язык Си также имеет следующие операции:

- унарный минус(инвертирует знак своего операнда) $a = -b$;
- унарного плюса(стоит по умолчанию)
- префиксный и постфиксный инкремент «++»: увеличивающий значение аргумента на единицу
- префиксный и постфиксный декремент «--», уменьшающий значение аргумента на единицу.

Постфиксная и префиксная записи

Операция	Поведение	Пример
Постфиксный инкремент	1. Сохраняет значение <code>b</code> в промежуточной временной переменной 2. Увеличивает на единицу значение переменной <code>b</code> 3. Возвращает значение промежуточной временной переменной	<pre>int a, b = 1; a = b++; printf("%d %d", a, b);</pre> Вывод на экран: 1 2
Префиксный инкремент	1. Увеличивает на единицу значение переменной <code>b</code> 2. Возвращает значение переменной <code>b</code>	<pre>int a, b = 1; a = ++b; printf("%d %d", a, b);</pre> Вывод на экран: 2 2

Пример

```
int a, b = 1, c;  
a = b++ * b++;  
printf("%d %d\n", a, b);  
b = 1;  
a = ++b * ++b;  
printf("%d %d", a, b);
```

Вывод на экран:

1 3

9 3

Вместо:

$i = i + 1;$

можно писать

$++i;$

Комбинированные операторы

<code>a += 4;</code>	то же, что и	<code>a = a + 4;</code>
<code>a -= 1;</code>	то же, что и	<code>a = a - 1;</code>
<code>a *= 10;</code>	то же, что и	<code>a = a * 10;</code>
<code>a /= 10;</code>	то же, что и	<code>a = a / 10;</code>
<code>a %= 2;</code>	то же, что и	<code>a = a % 2;</code>

Логическое выражение

– представляет собой некоторую формулу, состоящую из идентификаторов, констант и символов операций.

Основное отличие состоит в том, что в логических выражениях результат может быть лишь одним из двух: 0 – ложь, 1 – истина. Таким образом, логическое выражение представляет собой вопрос, на который можно ответить либо «да», либо «нет».

Операции сравнения

Операция	Название	Пример
==	Равно	I == 0
!=	Не равно	K != 15
>	Больше, чем	Z > 15.2
<	Меньше, чем	Tab < -132
>=	Больше или равно	Z11 >= 0
<=	Меньше или равно	Y2 <= 10

Логические операции

Операция	Название	Тип
	или	Бинарная
&&	и	Бинарная
!	НЕ	Унарная

Таблица истинности

x	y	x && y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x y
0	0	0
0	1	1
1	0	1
1	1	1

x	!x
0	1
1	0

Приоритет операций

Операция	Синтаксис	Приоритет (0 – самый высокий)
Скобки	(a)	0
Префиксный инкремент	++a	1
Постфиксный инкремент	a++	1
Префиксный декремент	--a	1
Постфиксный декремент	a--	1
Унарный минус	-a	1
Логическое «НЕ»	!a	1
Умножение	a * b	2
Деление	a / b	2
Взятие остатка от деления	a % b	2
Сложение	a + b	3
Вычитание	a - b	3
Больше	a > b	4
Меньше	a < b	4
Больше или равно	a >= b	4

Пример 1

- Пусть есть переменная A . Необходимо записать выражение, которое истинно, если значение A четное.
- Решение: $(A \% 2 == 1)$.

Пример 2

- Для двух переменных A и B записать выражение, которое истинно, когда эти переменные положительны.
- Решение: $(A > 0 \ \&\& \ B > 0)$.

Пример 3

- Даны две переменные A и B. Записать выражение, которое истинно, если эти переменные имеют разные значения.
- Решение: $(A \neq B)$.

Пример 4

- Даны три переменные A, B и C. Записать выражение, которое истинно, когда C равно максимальному значению из A и B.
- Решение: $((A \leq B \ \&\& \ C == B) \parallel (A \geq B \ \&\& \ C == A))$.