Введение в язык С

**Функцией** называется правило (отображение, соответствие, закон) f, ставящее каждому элементу x из некоторого множества X в соответствие определённый элемент y из множества Y, который обозначают как f(x).

Функция — это базовая модульная единица языка C. Каждая программа C имеет основную функцию, которая должна быть названа main . Функция main **служит отправной точкой для выполнения программы**. Она обычно управляет выполнением программы, вызывая другие ее функции.

int main()

{

return 0;

}

Данную функцию можно представить как main(x), где ∀x main(x)=0.

int main()

{

return 1+1;

}

Данная функция всегда будет возвращать 2.

int main()

{

return 4 + 4\*4 + 4\*4\*4 + 4\*4\*4\*4;

}

Данная функция будет возвращать сумму ряда , где x = 4.

Однако, так далеко не уедешь. Если мы хотим изменить значение x на 5, то придется изменять его 10 раз. Поэтому введем переменную!

int main()

{

int x;

x=4;

return x + x\*x + x\*x\*x + x\*x\*x\*x;

}

Все переменные в С имеют свой тип.

Типов в С несколько.

Их можно разделить на 3 класса:

Целые числа:

* short - предназначен для хранения целых чисел. Диапазон чисел от -32 768 до 32 767;
* unsigned short - предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до 65 535;
* int - предназначен для хранения целых чисел. Диапазон чисел от -2 147 483 648 до 2 147 483 647;
* unsigned int - предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до 4 294 967 295;
* long - предназначен для хранения целых чисел. Диапазон чисел от –9 223 372 036 854 775 808 до 9 223 372 036 854 775 807;
* unsigned long - предназначен для хранения целых положительных чисел. Диапазон чисел от 0 до 18 446 744 073 709 551 615.

Дробные числа

* float - для создания чисел с плавающей точкой. Диапазон чисел от от -3.4\*1038 до 3.4\*1038;
* double - для создания чисел с плавающей точкой. Диапазон чисел от от ±4.9\*10-324 до ±1.8\*10308.

Символьный тип

* char - тип данных для работы с символами. Позволяет поместить в одинарных кавычках какой-либо символ.

В данном примере мы ввели переменную x типа int. Т.е. внутри себя она может хранить только целые числа.

Вывод на экран

Для того, чтобы информация выводилась на экран, необходимо использовать функции из **стандартной библиотеки**.

Для вывода информации используется функция *printf* (от **print** **f**ormatted)*.*

На вход функция принимает **несколько** аргументов. Первый аргумент – спецификатор.

Спецификатор начинается с символа % и содержит в себе информацию о **типе** выводимой переменной, его формате и некоторые флаги.

|  |  |  |
| --- | --- | --- |
| Спецификатор | Тип | Комментарий |
| %d | Int | 10 система счисления |
| %x | unsigned int | 16 система счисления |
| %f | double |  |
| %c | Int | вывод символа с кодом |
| %s | char\* | вывод строки с нулевым завершающим байтом |

Возможно добавлять модификаторы размера. Например:

|  |  |  |
| --- | --- | --- |
| Спецификатор | Тип | Комментарий |
| %ld | long int |  |

Рассмотрим такой пример

#include <stdio.h>

int

main ()

{

char a = 'A';

printf ("A = %c \n", a);

printf ("A = %d \n", a);

return 0;

}

В данном случае символ А выведется как символ и как его код в формате ASCII.

Теперь придумаем что-то более интересное:

#include <stdio.h>

int main()

{

int r = 3;

double pi = 3.14;

double S = pi\*r\*r;

printf("%f", S);

return 0;

}

Первая строчка – подключение заголовочного файла, в котором описаны библиотечная функция *printf*.

В данном примере происходит вычисление площади круга.

Обратим внимание, что в строчке вычисления происходит перемножения разных типов. Рассмотрим это подробнее.

Преобразование типов

Правила!

* Если один из операндов имеет тип double, то второй операнд тоже будет преобразован к типу double.
* Если один из операндов имеет тип float, то второй операнд тоже будет преобразован к типу float

Чтобы привести один тип к другому необходимо перед ним в скобках указать обозначение приводимого типа.

#include <stdio.h>

int

main ()

{

int a = 10;

int b = 4;

int c = a / b; // 2

double d = a / b; // 2.00000

double e = (double) a / (double) b; // 2.50000

double f = a / (double) b; // 2.50000

double g = (double) a / b; // 2.50000

printf ("c = %d \n", c);

printf ("d = %f \n", d);

printf ("e = %f \n", e);

printf ("f = %f \n", f);

printf ("g = %f \n", g);

return 0;

}

Ввод данных.

Для ввода данных в программу используется стандартная функция *scanf. (scan f*ormatted*)*

Ее аргументы похожи на аргументы функции *printf.* Однако перед переменной, в которую будет производиться запись значения необходимо поставить знак &.

Зачем это делается и в каких случаях его не нужно указывать мы рассмотрим подробнее позже.

Ниже приведен пример простой программы, которая выводит на экран введенное значение.

#include <stdio.h>

int

main ()

{

int x;

scanf("%d", &x);

printf ("X = %d \n", x);

return 0;

}