

## Раздел «Язык Си» . SumProblem :

## Учимся складывать

Разнообразные вычисления – моделирование, решение алгебраических и дифференциальных уравнений – это то, для чего, собственно, и создавались первые компьютеры. Давайте и мы научимся использовать компьютер для вычислений. Начнём со сложения двух чисел.

В нашей программе будет две целочисленные переменные `a` и `b` – две ячейки памяти, в которых могут храниться целые числа из определенного диапазона (в 32 битной архитектуре от  $-2^{31}$  до  $2^{31}-1$  включительно).

Переменные объявляются сразу же после открывающей фигурной скобочкой функции `main`. Объявления начинаются со слова, указывающего тип переменных.

В языке C есть несколько типов данных. Они делятся на две группы: целые типы и типы с плавающей точкой. К первому типу относятся

```
char, short int, int, long int, long long int,
unsigned char, unsigned short int, unsigned int, unsigned long int, unsigned long long int.
```

Ко второму –

```
float, double, long double .
```

```
#include <stdio.h>
int main ()
{
    int a, b;
    printf ("Введите два числа: ");
    scanf ("%d%d", &a, &b);
    printf ("%d\n", a + b);
    return 0;
}
```

Функция `scanf`, также как и `printf`, определена в библиотеке `stdio`. Эта функция считывает данные, которые пользователь (тот, кто запустит вашу программу) вводит с клавиатуры. Слово `scan` означает *считывать данные*, а `print` – *печатать данные*. Буква "f" в конце соответствует первой букве английского слова *formatted*, то есть `scanf` и `printf` есть функции для форматированного ввода и вывода данных.

Первый аргумент у функции `scanf ("%d%d" – выражение в двойных кавычках после открывающей круглой скобки до запятой) есть описание формата входных данных, то есть описание типа данных, которые (как мы ожидаем) введёт пользователь. В этой программе мы ожидаем, что пользователь введет два целых числа.`

Символ `%` служебный, с него начинается описание формата. Обычно, после него идет один или два символа, определяющих тип входных данных.

Подробнее об спецификациях форматах ввода/вывода можно прочитать в документации (для Unix систем):

```
bash$ man 3 printf
bash$ man 3 scanf
```

Первый аргумент команды `man` есть номер раздела документации. Помощь по языкам C/C++ находится в третьем разделе.

Следует отметить, что для каждого типа данных, существует несколько форматов, и наоборот, для разных типов можно использовать один и тот же формат, хотя это редкость.

Приведённая программа умеет складывать только целые числа. Если вы хотите складывать действительные числа, то эту программу нужно несколько модифицировать. Ниже приведена программа, которая считывает два действительных числа и выводит результат четырех арифметических операций: сложения, вычитания, умножения и деления. Причём, программа выводит результаты вычислений два раза – сначала в обычном виде, а потом со

Поиск

Поиск

Раздел «Язык Си»

Главная  
Зачем учить C?  
Определения

Инструменты:

Поиск  
Изменения  
Index  
Статистика

Разделы

Информация  
Алгоритмы  
Язык Си  
Язык Ruby  
Язык  
Ассемблера  
El Judge  
Парадигмы  
Образование  
Сети  
Objective C

Login&gt;&gt;

специальным форматированием. Формат "%10.3lf" соответствует выводу числа типа double, при котором под запись числа выделяется ровно 10 позиций (если это возможно), а после запятой пишется ровно три знака. Равнение происходит по правому краю.

```
/* Программа "Арифметические операции с числами с плавающей точкой" */
#include <stdio.h>
int main ()
{
    double a, b;
    printf ("Введите два числа: ");
    while ( scanf ("%lf%lf", &a, &b) == 2 )
    {
        printf ("%lf %lf %lf %lf\n", a + b, a - b, a * b, a / b );
        printf ("a + b=%10.3lf\n a - b=%10.3lf\n a * b=%10.3lf\n a / b=%10.3lf\n",
                a + b, a - b, a * b, a / b );
    }
    return 0;
}
```

В этой программе мы встречаемся с оператором while. Конструкция

```
while ( A ) B;
```

означает буквально следующее: "Пока выполнено условие A делать B".

В нашем случае A есть

```
scanf ("%lf%lf", &a, &b) == 2.
```

Что соответствует логическому выражению: "пользователь ввёл два действительных числа, и они удачно считаны в переменные a и b"

Таким образом, эта программа будет считывать пары чисел и выводить результаты арифметических операций, пока пользователь не введёт что-нибудь непохожее на число.

Цикл while закончится тогда, когда функция scanf не сможет успешно считать два числа.

Заметьте, что после каждой команды стоит точка с запятой. Одна из самых популярных синтаксических ошибок начинающих программистов – это не ставить точку запятой в конце команды.

(с) Материалы раздела "Язык Си" публикуются под лицензией GNU Free Documentation License.