

Урок 2. Арифметические операции

На этом уроке Вы научитесь выполнять арифметические операции с числами, осуществлять расчёты по формулам, а также вводить исходные данные с клавиатуры и выводить результат на экран.

Задание 1.

Найти сумму двух целых чисел и вывести результат на экран.

Решение:

Создайте проект `lesson2_1` (*File*→*New Project...* или **Ctrl+N**), добавьте к нему файл `lesson2_1.c` (нажать **Insert**; в появившемся диалоге в поле *File Name* введите `lesson2_1.c`, нажмите кнопку **Add**, а затем **Close**). Откройте его в редакторе (двойной щелчок на имени файла в окне `lesson2_1.exe`) и наберите текст программы.

```
#include <stdio.h>

void main()
{
    int a, b, c;

    a = 10;
    b = 20;
    c = a + b;
    printf("%d %d %d", a, b, c);

    printf("\n"); while (!kbhit());
}
```

Сохраните текст программы (**CTRL+S**), откомпилируйте программу (*Actions*→*Make all* или **F5**) и запустите её (*Targets*→*Run* или **CTRL+R**). Для того чтобы закрыть появившееся окно с результатом, нажмите ENTER.



Далее сохраните проект (*File*→*Save Project* или **CTRL+S**) и закройте окно интегрированной среды разработки.

Что делать, если программа не компилируется?

Проверьте текст программы – он должен быть набран без ошибок. Если же ошибка не находится, то обратите внимание вот на что:

1. Скобки должны быть парными ((и) , { и } , [и])
2. Кавычки тоже должны идти парами
3. На своих ли местах точки с запятой?
4. Не перепутаны ли строчные и прописные буквы (т.е. не набрано ли Int вместо int или A вместо a)?
5. Нет ли опечаток в ключевых словах?

Компилятор может и сам показать, где именно найдена ошибка. Для этого в окне IDE Log найдите строку с сообщением об ошибке (lesson2_1.c (номер_строки) : Error! Суть ошибки) И сделайте по ней двойной щелчок левой кнопкой мыши. Файл lesson2_1.c откроется в редакторе на том месте, где допущена ошибка.

Ошибки при вводе текста программы и в логике программы – это естественное явление, и к нему нужно быть готовым. Не ошибается тот, кто не работает!

Пояснения к тексту программы

По сравнению с первой программой в ней появилось много нового, поэтому подробно разберём текст программы.

```
#include <stdio.h>
```

Подключение стандартной библиотеки ввода-вывода

```
void main()
```

Объявление функции main. Выполнение программы начинается с вызова этой функции. Тело функции заключено между фигурными (операторными) скобками { }. Ключевое слово void означает, что функция ничего не возвращает.

```
int a, b, c;
```

В этой строке происходит **объявление переменных** a, b, c.

Переменная – это ячейка памяти, у которой есть имя.

Ключевое слово int указывает на то, что переменные a, b, c предназначены для хранения целых чисел (принято говорить, что они – типа int). int – это сокращение от слова Integer (переводится как «целый»).

Имена переменных перечисляются через запятую, объявление оканчивается точкой с запятой.

Результатом объявления переменных являются три выделенные ячейки памяти *a*, *b* и *c*. В них и будут храниться целые числа.

Имя переменной должно начинаться с латинской буквы и дальше могут идти буквы латинского алфавита, цифры и знаки подчёркивания. Пробелы в имени не допускаются. **Строчные и прописные буквы различаются** (т.е. переменные *Num* и *num* – разные).

```
a = 10;
```

«**=**» - это **оператор присваивания**. В результате в переменную (ячейке памяти) *a* будет помещено значение 10. В следующей строке программе

```
b = 20;
```

переменной *b* присваивается значение 20.

```
c = a + b;
```

В этой строке происходит сложение двух чисел. Сначала будет вычислено арифметическое выражение $a + b$, *a* затем оператор присваивания запишет результат в переменную *c*.

Не путайте оператор присваивания с математическим равенством!
a=b это не b=a!

a=b; Взять число из переменной *b* и записать его в переменную *a*

b=a; Взять число из переменной *a* и записать его в переменную *b*

После выполнения этих строк программы у нас есть переменные (ячейки памяти), в которых есть два числа (10 и 20) и результат их сложения. На экран они выводятся в строке:

```
printf("%d %d %d", a, b, c);
```

Эта функция уже встречалась в первой программе. Она выводила на экран строку текста (`printf("Hello, World!");`)

Здесь строка тоже выводится на экран, но вместо трёх спецификаторов `%d` при печати будут выведены три целых числа из переменных *a*, *b*, *c*. Пробелы между `%d` превратятся в пробелы между числами.

```
printf("\n"); while (!kbhit());
```

В этой строке происходит ожидание нажатия клавиши. Подробный разбор этой строки будет дан в следующих уроках.

Задание 2.

Ввести с клавиатуры два числа, найти их сумму и вывести результат на экран.

Решение:

Создайте проект `lesson2_2` (**Ctrl+N**), добавьте к нему файл `lesson2_2.c` (Нажать **Insert**; в появившемся диалоге в поле *File Name* введите `lesson2_2.c`, нажмите кнопку **Add**, а затем **Close**). Откройте его в редакторе (двойной щелчок на имени файла в окне `lesson2_1.exe`) и наберите текст программы.

```
#include <stdio.h>

void main()
{
    int a, b, c;

    scanf("%d", &a);
    scanf("%d", &b);
    c = a + b;
    printf("%d %d %d", a, b, c);

    printf("\n"); while (!kbhit());
}
```

Откомпилируйте программу (F5) и запустите её (CTRL+R). Появится окно программы с мигающим курсором.

Введите первое число и нажмите Enter. Введите второе число и нажмите Enter. После появления строки с ответом нажмите любую клавишу.

Пояснения к тексту программы

Теперь складываемые числа не «зашиты» в текст программы, а запрашиваются с клавиатуры. Поэтому строки `a=10;` `b=20;` заменены на:

```
scanf("%d", &a);
scanf("%d", &b);
```

Функция `scanf` обеспечивает ввод с клавиатуры. Строка со спецификатором `%d` означает, что вводится одно целое число. Символ `&` перед именами переменных означает то, что функции `scanf` будут переданы не значения переменных `a` и `b`, а их адреса в памяти компьютера (или как говорят, указатели на переменные).

Чтобы это лучше понять, давайте проведём аналогию с Интернетом: разница между `a` и `&a` примерно такая же, как между содержимым сайта и ссылкой на сайт.

Упражнение: переделайте программу так, чтобы она считала не сумму, а разность, произведение, частное или остаток от деления двух чисел. Это потребует изменение всего лишь одного-двух символов.

Обозначения арифметических операторов:

Операторы языка Си	Что означает
<code>a + b</code>	Сложение двух чисел
<code>a - b</code>	Вычитание двух чисел
<code>a * b</code>	Умножение двух чисел
<code>a / b</code>	Деление двух чисел
<code>a % b</code>	Остаток от деления двух чисел
<code>()</code>	Скобки

Задание 3*

Есть треугольник с основанием a и высотой h . Найдите площадь треугольника (формула - $S = \frac{ah}{2}$). Входные и выходные данные – дробные числа.

Решение:

```
#include <stdio.h>

void main()
{
    double a, h, s;

    printf("Расчёт площади треугольника\n");
    printf("Введите основание a: ");
    scanf("%lf", &a);
    printf("Введите высоту h: ");
    scanf("%lf", &h);

    s = a*h/2.0;
    printf("Площадь s: %lf", s);

    printf("\n"); while (!kbhit());
}
```

Важное новшество в этой программе – вместо `int` для переменных использован тип `double`. Он предназначен для хранения дробных чисел в памяти компьютера. Из-за этого спецификаторы `%d` были заменены на `%lf` (*если вместо `%lf` использовать `%d`, то будет показано не округлённое число, а бессмыслица*).

При вводе десятичных дробей помните, что в языке Си целая и дробная части разделяются точкой, а не запятой!

Также обратите внимание на сочетание символов `\n` в строке

```
printf("Расчёт площади треугольника\n");
```

При выводе на экран вместо него будет подставлен символ новой строки и следующая строка («Введите основание a:») будет печататься на новой строке.

Упражнения:

1. Программа для нахождения периметра прямоугольника.
Формула - $P = 2(a + b)$
Указание: не забывайте знак умножения * между скобкой и числом/именем переменной. На языке Си строка для вычисления по формуле запишется как $p=2 * (a+b)$;
2. Напишите программу для пересчёта из одной валюты в другую по заданному курсу. Пусть программа запрашивает сумму в иностранной валюте, текущий курс, а на выходе выдаёт сумму в рублях (можно и наоборот – на входе рубли, на выходе – иностранная).
Указание: не забудьте применить тип `double` для всех переменных.
3. Напишите программу для перевода дюймов в сантиметры (1 дюйм = 2,54 см.). Ею, например, можно пользоваться для пересчёта размеров диагоналей мониторов в более привычные сантиметры.