

# **Данные в языке C**

**Практическое занятие**

# Задание 1. Типы данных

- Выберите подходящий тип данных для переменных в каждой из следующих ситуаций. Будьте как можно более конкретными. Если ответом является целочисленный тип данных, то используйте соответствующий тип. Если переменная должна быть константной, то оформите её как константу.
  - **Возраст пользователя.**
  - **Нравится ли определенный цвет пользователю?**
  - **Число Пи.**
  - **Количество страниц в учебнике.**
  - **Цена акций в долларах (дробная часть присутствует).**
  - **Сколько раз вы моргнули за всю свою жизнь? (*Примечание: Ответ исчисляется в миллионах*)**
  - **Пользователь выбирает опцию с помощью ввода определенной буквы.**

# Какие константы(литералы) здесь записаны

1. 0.25f,
2. .25L,
3. 1.25E10,
4. 255U,
5. 0x100,
6. 012,
7. 3LL,
8. 3i64,
9. 'A'
10. 67uL

Тип константы распознается компилятором по её написанию в тексте программы: по умолчанию или по использованным программистом префиксам и суффиксам

# Задание 2. Неявное преобразование

```
#include <stdio.h>
int main()
{
    double a =3.0;
    short b=2;
    printf("%f\n", a+b );
return 0;
}
```

Ожидается, что результатом выражения  $5u - 10$  будет -5, поскольку  $5 - 10 = -5$ , но результат:

4294967291

Здесь значение signed int (10) подвергается расширению в unsigned int (которое имеет более высокий приоритет), и выражение вычисляется как unsigned int. А поскольку unsigned — это только положительные числа, то происходит переполнение, и мы имеем то, что имеем.

Это одна из тех многих веских причин избегать использования типа unsigned int вообще.

```
#include <stdio.h>
int main()
{
    int x = 10;
    printf("%d\n", 5u - x );
    // 5u означает значение 5 типа unsigned int
return 0;
}
```

# Задание 3. Явные преобразования

Оформите в виде программ:

```
int i1 = 11;  
int i2 = 3;  
float x = (float)i1 / i2;
```

```
char c = 97;  
printf("c = %c/n", c);    //выведется 'a'  
printf("c = %f/n", (int)c); // в результате выведется 97, а не 'a'
```

# Задание 4. Побитовые операции

- Таблица истинности:

X	Y	X & Y	X   Y	X ^ Y	~X
0	0	0	0	0	1
1	0	0	1	1	0
0	1	0	1	1	1
1	1	1	1	0	0

- Напишите программу с выводом в терминал результата каждой из операций.

```
x = 7      // 00000111 (7)
x = x >> 1  // 00000011 (3)
x = x << 1  // 00000110 (6)
x = x << 5  // 11000000 (-64)
x = x >> 2  // 11110000 (-16)
```

```
a = 7
b = 9
c = a & b
c = a | b
c = a ^ b
c = ~a
```

# Домашняя работа

Домашняя работа  
оформляется под названием  
HW\_3, грузим на GitHub.

- Загрузить предыдущую домашнюю работу HW\_2 на GitHub, выслать ссылку мне на почту или в Telegram.
- Выяснить что делает компилятор преобразовывая к «старшему» знаковому и беззнаковому типу знаковые и беззнаковые переменные «младшего» типа.
- Даны два числа  $x$ ,  $y$ . Вывести первое число, если оно больше второго, и оба числа, если это не так.
- Написать программу демонстрирующую переполнение по типу `char` и вывести на экран и число и символ.
- Написать программу с переполнением по типу `double`.
- Дано действительное число  $a$ . Пользуясь только умножением получить  $a^4$  и  $a^{10}$  за четыре операции.