

ПЕРЕМЕННЫЕ И ТИПЫ ДАННЫХ

НАШИ ПРАВИЛА



Включенная камера



Вопросы по поднятой руке



Не перебиваем друг друга



Все вопросы, не связанные с тематикой курса (орг-вопросы и т. д.), должны быть направлены куратору



Подготовьте свое рабочее окружение для возможной демонстрации экрана (закройте лишние соцсети и прочие приложения)

ЦЕЛЬ

**Изучить способы работы с переменными и
особенности различных типов данных**

ПЛАН ЗАНЯТИЯ

■ Представление информации и переменные

■ int

■ double

■ char

■ Преобразование типов

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

Компьютер хранит информацию в **двоичном (бинарном)** коде

Для того, чтобы правильно интерпретировать эту информацию, необходимо знать ее **тип** (например, целое число, символ и т.д.)

Каждое двоичное значение хранится в одном **бите** и представляет собой либо 1 (истина, true) либо 0 (ложь, false)

7 6 5 4 3 2 1 0

$$10001001_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 137_{10}$$

Бинарный код основан на **двоичной системе счисления**, тогда как мы используем **десятичную систему**.

Группа из **8 бит** образует 1 **байт**. Максимально число, которое можно выразить 1 байтом - 255 (**почему?**)

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

Для того, чтобы представить десятичное число в двоичной системе счисления, можно воспользоваться вычислением остатков от деления на 2 и “прочитать” полученные значения снизу вверх

$$137 / 2 = 68 \quad \text{остаток: } 1$$

$$68 / 2 = 34 \quad \text{остаток: } 0$$

$$34 / 2 = 17 \quad \text{остаток: } 0$$

$$17 / 2 = 8 \quad \text{остаток: } 1$$

$$8 / 2 = 4 \quad \text{остаток: } 0$$

$$4 / 2 = 2 \quad \text{остаток: } 0$$

$$2 / 2 = 1 \quad \text{остаток: } 0$$

$$1 / 2 = 0 \quad \text{остаток: } 1$$

ИТОГ: 10001001_2

ДВОИЧНАЯ АРИФМЕТИКА

Операции сложения и вычитания для двоичной системы счисления аналогичны этим операциям в десятичной системе счисления:

$$0_2 + 0_2 = 0_2 = 0_{10}$$

$$0_2 + 1_2 = 1_2 = 1_{10}$$

$$1_2 + 1_2 = 10_2 = 2_{10}$$

$$1_2 + 1_2 + 1_2 = 11_2 = 3_{10}$$

Сложение двоичных чисел “столбиком”

$$\begin{array}{r} 11 111 \\ + 11100111 \\ \hline 111001110 \end{array}$$

ПРЕДСТАВЛЕНИЕ ИНФОРМАЦИИ

Для хранения отрицательных чисел компьютер использует **дополнительный код**

Представление -5_{10}

$5_{10} = 00000101_2$ (прямой код)

обратный код: 11111010_2
дополнительный код: 11111011_2

Для того, чтобы понять, что байт хранит отрицательное число, старший разряд используется как **знак** (1 - отрицательное число в дополнительном коде, 0 - положительное в прямом коде)

11111011_2

Следовательно, с учетом знака, максимальное число для 1-го байта **127**, минимальное **-128** (почему?)

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

- Компьютер хранит информацию в двоичном коде

- Байт - это набор 8 бит

- Механизмы преобразования чисел из десятичной системы счисления в двоичную и обратно

- Основы двоичной арифметики

- Способ хранения отрицательных чисел

ПЕРЕМЕННЫЕ

Переменная представляет собой участок памяти, которому присвоено определенное **имя**

Чтобы использовать переменную, ее необходимо **объявить**, указав имя и тип

```
int x = 5;  
double y = 2.5;  
char c = 'A'
```

Переменная содержит данные определенного **типа** - целое число, вещественное число, символ и т.д.

Переменной можно **присвоить** какое-либо значение, соответствующее ее типу



int

int - целочисленный тип данных, занимает 4 байта

С переменными целого типа возможно выполнение следующих операций - **+**, **-**, *****, **/**, **%**

```
int a = 13;
```

```
int b = 3;
```

```
int x = a / b; // 4
```

```
int y = a % b; // 1
```

Минимальное значение **-2147483648**, максимальное **2147483647**

Результат деления переменных целого типа всегда будет целочисленным

$$\begin{array}{ccccccc} & & \text{делитель} & & & & \text{остаток} \\ \boxed{13} & = & \boxed{3} & * & \boxed{4} & + & \boxed{1} \\ \text{число} & & & & \text{целая часть} & & \end{array}$$

double

double - вещественный тип данных, занимает 8 байт

```
double a = 7;  
double b = 2;  
double c = a / b; // 3.5
```

Минимальное положительное - $4.9 * 10^{-324}$,
максимальное положительное значение -
 $1.7976931348623157 * 10^{308}$

```
int a = 7;  
int b = 2;  
double c = a / b; // 3.0
```

char

char - символьный тип, занимает в памяти **2 байта**

```
char upper = 'A';  
char lower = 'a';  
char cyrillic = 'Ы';
```

Представляет собой числовое значение в диапазоне от 0 до 65 535, которое ассоциировано с символом из таблицы **Unicode**

```
char d = '8';  
int number = d - '0'; // d = 8
```

ПРИМИТИВНЫЕ ТИПЫ ДАННЫХ

Целочисленные

byte
int
short
long

Символьные

char

Вещественные

float
double

Логические

boolean

СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Примитивные типы данных и их назначения

■ Особенности работы с примитивными типами

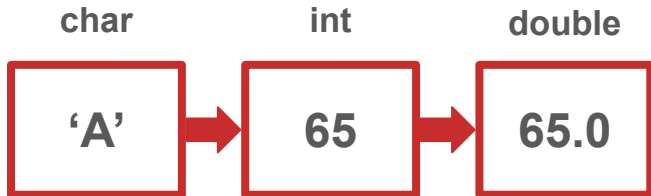
■ Литеральные значения и переполнение

■ Основы двоичной арифметики

■ Способ хранения отрицательных чисел

ПРЕОБРАЗОВАНИЯ ТИПОВ

Неявное преобразование выполняется автоматически из переменных типов с меньшим диапазоном значений в переменные типов с большим диапазоном значений



Такое преобразование выполняется автоматически, поскольку потеря данных невозможна.

ПРЕОБРАЗОВАНИЯ ТИПОВ

Явное преобразование выполняется с явным указанием целевого типа и применяется в обратной ситуации

```
double x = 66.3;  
int y = (int)x;  
char z = (char)y;
```

Такое преобразование выполняется автоматически, поскольку потеря данных невозможна.



СТАВИМ ПЛЮС, ЕСЛИ ПОНЯТНО

■ Неявное преобразование

■ Возможные потери данных при явном преобразовании

ПОИГРАЕМ ;)

■ Переменная

■ Тип данных

■ double

■ Явное преобразование

■ Литерал

ДОМАШНЕЕ ЗАДАНИЕ



Ваша новая IT-профессия – Ваш новый уровень жизни

Программирование с нуля в
немецкой школе AIT TR GmbH