Билет 14.

1.

В Java есть 8 примитивных типов, которые делят на 4 группы, вот они:

1. Целые числа - byte, short, int, long
2. Числа с плавающей точкой (иначе вещественные) - float, double
3. Логический - boolean
4. Символьный – char

**Целочисленные типы**

Целочисленные типы различаются между собой только диапазонами возможных значений, например, для хранения номера элемента в таблице Менделеева пока хватит переменной типа byte.

|  |  |  |
| --- | --- | --- |
| **Тип** | **Размер (бит)** | **Диапазон** |
| byte | 8 бит | от -128 до 127 |
| short | 16 бит | от -32768 до 32767 |
| char | 16 бит | беззнаковое целое число, представляющее собой символ UTF-16 (буквы и цифры) |
| int | 32 бит | от -2147483648 до 2147483647 |
| long | 64 бит | от -9223372036854775808L до 9223372036854775807L |

Пример использования целочисленных типов:

**public** **class** **IntegralTypes** {

**public** **static** void main(String[] args) {

byte b = 216; *// Вот тут будет ошибка, т.к. у нас диапазон от -128 до 127!*

short s = 1123;

int i = 64536;

long l = 2147483648L; *// Постфикс l или L обозначает литералы типа long*

System.out.println(i);

System.out.println(b);

System.out.println(s);

System.out.println(l);

}

}

Символы тоже относят к целочисленным типам из-за особенностей представления в памяти и традиций.

**public** **class** **Characters** {

**public** **static** void main(String[] args) {

char a = 'a', b, c = 'c';

b = (char) ((a + c) / 2); *// Можно складывать, вычитать, делить и умножать*

*// Но из-за особенностей арифметики Java результат приходится приводить к типу char явно*

System.out.println(b); *// Выведет символ 'b'*

}

}

**Типы с плавающей точкой**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Размер (бит)** | **Диапазон** |
| float | 32 | от 1.4e-45f до 3.4e+38f |
| double | 64 | от 4.9e-324 до 1.7e+308 |

Пример использования:

**public** **class** **FloatingPointTypes** {

**public** **static** void main(String[] args) {

double a, b = 4.12;

a = 22.1 + b;

float pi = 3.14f; *// При использовании типа float требуется указывать суффикс f или F*

*// так как без них типом литерала будет считаться double*

float anotherPi = (float) 3.14; *// Можно привести явно*

double c = 27;

double d = pi \* c;

System.out.println(d);

}

}

**Логический тип**

|  |  |  |
| --- | --- | --- |
| **Тип** | **Размер (бит)** | **Значение** |
| boolean | 8 (в массивах), 32 (не в массивах используется int) | true (истина) или false (ложь) |

Преобразование примитивных типов возможно, но с нюансами. Если переменную типа, требующий меньше памяти, «положить» в тип, требующий больше памяти, то такое преобразование пройдёт без проблем. А вот если наоборот, то необходимо явно указывать, к какому типу преобразуется переменная. К тому же, в новый тип данных будет перемещено только то количество первых бит, которое допускает для хранения конечный тип данных. Этот процесс называется **сужением**.

Пример.

**public** **static** **void** main(String[] args) {

**int** bigNumber = 10000000;

**short** littleNumber = 1000;

littleNumber = (**short**) bigNumber;

System.out.println(littleNumber);

}

Для дробных типов (float и double) сужение происходит по-своему. Если попытаться привести такое число к целочисленному типу, у него будет отброшена дробная часть.

Пример.

**public** **static** **void** main(String[] args) {

**double** d = 2.7;

**long** x = (**int**) d;

System.out.println(x);

}

Вывод в консоль: *2*

Логический тип не приводится ни к одному типу данных.