**Домашнее задание 2**

**Теория (вопросы)**

1. **Что необходимо указать для корректного объявления переменной?**

Для корректного объявления переменной необходимо указать:

* тип данных. В нашем примере указан тип int, то есть переменная содержит целое число (типы данных мы разберём подробнее чуть позже);
* имя переменной. В данном примере имя y;
* начальное значение переменной или, другими словами, инициализация переменной. В нашем примере переменной присвоено значение 1. Но инициализация не является обязательным условием для объявления переменной. Например, можно было бы написать так:

1. **Что такое инициализация переменной?**

Инициализация — это присвоение переменной начального значения.

1. **Какие правила именования переменных вы знаете?**

Для именования переменных существует несколько правил:

* имена переменных должны начинаться с буквы, знака $ или знака подчеркивания \_. Имена не могут начинаться с цифр;
* после первого символа в имени могут идти любые комбинации букв, цифр, знаков подчеркивания и $;
* нет ограничения на количество символов, которое может содержать имя переменной;
* в качестве имен нельзя использовать зарезервированные в Java слова
* имена переменных в Java регистрозависимые — это значит, что переменные с именами hello и Hello — это две различные переменные.

А также есть несколько необязательных правил, но именование переменных согласно им является хорошей практикой в программировании:

* при выборе имени переменных следует использовать полные слова вместо аббревиатур и сокращений. Это сделает код более понятным, удобным для чтения и понимания;
* если имя переменной состоит только из одного слова, его следует записать маленькими буквами. Если оно состоит из более чем одного слова, то отделяйте каждое последующее слово в имени переменной заглавной буквой. Например: myCourse, firstProgram.
* имя переменной должно начинаться с маленькой буквы, так как с заглавной буквы принято писать имена классов.

1. **На какие две группы делятся все типы данных?**

Типы данных в Java делятся на примитивные и ссылочные.

1. **Что такое примитивные типы?**

Примитивные типы — это основа языка. Эти типы являются неделимыми и состоят из байтов. На основе примитивных типов строятся все более сложные типы.

1. **Какие примитивные типы данных вы знаете?**

В Java существует 8 примитивных типов данных:

* byte (целые числа, 1 байт);
* short (целые числа, 2 байта);
* int (целые числа, 4 байта);
* long (целые числа, 8 байтов);
* float (вещественные числа, 4 байта);
* double (вещественные числа, 8 байтов);
* char (символ Unicode, 2 байта);
* boolean (значение истина/ложь, 1 байт).

1. **Что можно хранить в целочисленных типа и типах с плавающей точкой?**

Целочисленные типы — это типы, которые могут хранить в себе только целые числа. Они могут быть как положительными, так и отрицательными.

(В Java существуют четыре целочисленных типа: byte, short, int, long.)

Числа с плавающей точкой — это числа, которые имеют дробные части ( обычно выраженные с десятичной точкой).

Типы – double, float.

1. **Что такое компиляция программы, компилятор?**

Компиляция в программировании — это приведение исходного кода в байт-код для последующего старта программы.

Компилятор — то, что преобразовывает понятный человеку код на языке Java в понятный компьютеру байт-код.

1. **Что можно хранить в переменных типа char, а что в String? В чём отличие?**

Char – это символьный тип данных, используется для хранения в переменных любых символов. Их нужно записывать в одинарные кавычки ' ' , и только по одному.

Инициализировать переменную типа char можно тремя способами:

* по индексу символа в таблице UTF-8;
* по значению символа;
* через шестнадцатеричную форму Unicode.

String – это ссылочный тип данных, предназначен для хранения любых строк с текстом.

Тип String можно проинициализировать несколькими способами:

* используя конструкцию = new String(””);
* записывая значение в двойных кавычках “”.

Отличия:

Тип Char, являясь примитивным типом переменной, хранит в себе значение переменной, состоит из байтов. String, являясь ссылочным, хранит не значение переменной, а адрес ячейки памяти, в которой расположен определённый объект.

Значение переменной Char записывается в одинарных кавычках, а String – в двойных кавычках.

Каждый ссылочный тип является классом, примитивный тип может не быть классом.

1. **Что такое ссылочная переменная? В чём отличие от примитива?**

Ссылочные переменные хранят не значение переменной, как примитивы, а адрес ячейки памяти, в которой расположен определённый объект. То есть это «карточка» с адресом, имея которую мы можем найти наш объект в общей памяти компьютера и выполнять с ним некоторые манипуляции.

Каждый ссылочный тип является классом, примитивный тип может не быть классом.

1. **Что такое обертка?**

Соответствующий примитиву ссылочный тип данных называется оберткой. Все классы-обертки пишутся с заглавной буквы, так как они являются полноценными классами.

1. **Как называется процесс преобразования примитива в его обёртку и обратный процесс?**

Процесс преобразования примитива в ссылочный тип называется упаковкой (boxing). Обратный процесс преобразования ссылочного типа в примитив называется распаковкой (unboxing).

1. **Какие способы преобразования примитива в обёртку вы знаете?**

Для упаковки есть несколько способов:

1. через конструктор.
2. через метод .valueOf(a)
3. автоматическая упаковка.
4. **Какие способы преобразования обёртки в примитив вы знаете?**
   1. через методы .longValue(), .intValue(), byteValue(), shortValue(), floatValue(), doubleValue(), charValue(), booleanValue().
   2. автоматическая распаковка.
5. **Приведите примеры арифметических операций.**

public class Main {

public static void main(String[] args) {

int a = 100;

int b = 50;

System.out.println(a+b);

System.out.println(a-b);

System.out.println(a\*b);

System.out.println(a/b);

}

}

1. **Приведите примеры операций сравнения.**

public class Main {

public static void main(String[] args) {

int a = 33;

int b = 999;

System.out.println(a == b);

System.out.println(a > b);

System.out.println(a < b);

System.out.println(a >= b);

System.out.println(a <= b);

System.out.println(a != b);

System.out.println(a = b);

}

}

1. **Что такое унарные операции? Приведите примеры.**

Понятие унарные появилось от слова «‎uno» — «‎один». Такие операции совершают действия только с одним числом.

Первый вид унарных операций — это унарный минус - . Эта операция позволяет взять число с противоположным знаком.

Пример:

public class Main {

public static void main(String[] args) {

int a = 10;

a++;

System.out.println(a);

a--;

System.out.println(a);

}

}

1. **Что такое комбинированные операции? Приведите примеры.**

Комбинированные операции помогают осуществить два действия сразу — одну из арифметических операций и присваивание:

Пример:

public class Main {

public static void main(String[] args) {

int a = 10;

int b = 5;

a += b;

System.out.println(a);

a = 10;

b = 5;

a -= b;

System.out.println(a);

}

}

1. **Какие логические операции вы знаете?**

Логические операции являются операциями над булевыми (boolean) переменными.

Логическое «и» (&&) возвращает true тогда, когда оба выражения являются истинными, в противном случае — false.

Логическое отрицание (!) помогает взять противоположное значение переменной, то есть было true, вернётся false, и наоборот:

1. **Назовите несколько операций над строками из урока и объясните их функции.**

метод .length() - возвращает в результате количество символов, из которого состоит строка:

метод .equals() – сравнивает две строки.

метод .contains() – позволяет узнать, содержит ли строка определённый символ или целую подстроку.

метод .charAt(int index) – позволяет выяснить, какой символ находится на заданной позиции.