**Практическое занятие № 2**

**Примитивные типы данных и их операции**

Для написания программ на языке Java необходимо знать его синтаксис. Начнем знакомство с основами языка. В этой работе мы рассмотрим:

* что такое переменные в java и как их объявлять,
* как правильно именовать переменные,
* какие существуют типы данных в java:
  + примитивные типы данных,
  + объектные типы данных,
  + отдельно рассмотрим тип String.

**Java переменные**

Класс в Java состоит из атрибутов и методов. Переменные могут быть атрибутами класса, параметрами метода или могут использоваться в программе для краткосрочного хранения данных. В языке Java все переменные должны быть объявлены, перед тем, как они будут использоваться.

**Объявление переменных в java**

Пример:

int x = 1; int y = 2;

При объявлении переменной, в следующей последовательности указываются:

* **тип данных** (в данном примере - int - переменная содержит целое число),
* **имя переменной** (в данном примере имена - x и y),
* **начальное значение переменной** или, другими словами, **инициализация переменной**. В данном примере переменным x и y присвоены значения 1 и 2. Однако, это не является обязательным условием при объявлении переменной.

Пример: **объявление переменных без инициализации**:

int x; int y;

После каждой строки при объявлении переменных необходимо ставить точку с запятой «;».

Если нужно **объявить несколько переменных одного типа**, то это также можно сделать одной строкой, указав имена переменных через запятую.

int x,y;

**Правила именования переменных в java**

1. Имя переменной должно **начинаться с буквы (маленькой) и состоять из букв (Unicode) цифр и символа подчеркивания «\_»**. Технически возможно начать имя переменной также с «$» или «\_», однако это запрещено соглашением по оформлению кода в Java (Java Code Conventions). Кроме того, символ доллара «$», по соглашению, никогда не используется вообще. В соответствии с соглашением имя переменной должно начинаться именно с маленькой буквы (с заглавной буквы начинаются имена классов). Пробелы при именовании переменных не допускаются.
2. Имя переменной **не должно бытьключевым или зарезервированным словом языка Java.**
3. Имя переменной **чувствительно к регистру**. newVariable и newvariable - разные имена.
4. При выборе имени переменных, следует **использовать полные слова** вместо загадочных аббревиатур. Это сделает ваш код более удобным для чтения и понимания. Во многих случаях это также сделает ваш код самодокументируемым.
5. Если выбранное вами имя переменной состоит только из одного слова - запишите его маленькими буквами. Если оно состоит из более чем одного слова, то **отделяйте каждое последующее слово в имени переменной заглавной буквой.** Например: superCounter, myDomesticAnimal
6. Если переменная сохраняет постоянное значение, то каждое слово следует писать заглавными буквами и отделять при помощи символа подчеркивания. Пример: static final int NUMBER\_OF\_HOURS\_IN\_A\_DAY = 24

**Типы данных в java**

Каждая переменная и каждое выражение в Java обладает типом и этот тип строго определен.

**Примитивные типы данных**

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)

Эти 8 типов служат основой для всех остальных типов данных. Примитивные типы обладают явным диапазоном допустимых значений.

**byte**- диапазон допустимых значений от -128 до 127

//объявление переменных типа byte.

byte getByte, putByte;

// инициализация переменных

getByte = 0;

putByte = 0;

Переменные типа byte полезны при работе с потоком данных, который поступает из сети или файла.

**short** - диапазон допустимых значений от -32768 до 32767

//объявление и инициализация переменной типа short.

short employeeID = 0;

**int**- диапазон допустимых значений от -2147483648 до 2147483647

//объявление и инициализация переменной типа int.

int max = 2147483647;

Тип int используется чаще при работе с целочисленными данными, нежели byte и short, даже если их диапазона хватает. Это происходит потому, что при указании значений типа byte и short в выражениях, их тип все равно автоматически повышается до int при вычислении.

**long** - диапазон допустимых значений от -9223372036854775808 до 9223372036854775807

//Использование переменных типа long.

long days = getDays();

long seconds;

seconds = days \* 24 \* 60 \* 60;

Тип удобен для работы с большими целыми числами.

**float**- диапазон допустимых значений от ~1,4\*10-45 до ~3,4\*1038

//Объявление и инициализация переменных типа float.

float usd = 61.24f;

float eur = 74.03f;

Удобен для использования, когда не требуется особой точности в дробной части числа.

**double**- диапазон допустимых значений от ~4,9\*10-324 до ~1,8\*10308

//Объявление и инициализация переменных типа double.

double pi = 3.14159;

Математические функции такие как sin(), cos(), sqrt() возвращают значение double

**char** - символьный тип данных представляет собой один 16-битный Unicode символ. Он имеет минимальное значение ‘\ u0000′ (или 0), и максимальное значение ‘\ uffff’ (или 65535 включительно). Символы char можно задавать также при помощи соответствующих чисел. Например символ ‘Ы’ соответствует числу 1067. Рассмотрим на примере:

public static void main(String[] args) {

char symb1=1067;

char symb2 ='Ы';

System.out.println("symb1 contains "+ symb1);

System.out.println("symb2 contains "+ symb2);

}

Вывод этой программы будет:

symb1 contains Ы

symb2 contains Ы

Небольшой пример того, как узнать, какому числу соответствует символ. Основан на претипировании данных.

public static void main(String[] args) {

char ch = 'J';

int intCh = (int) ch;

System.out.println("J corresponds with "+ intCh);

}

На выводе программа показывает, что символу 'J' соответствует число 74.

J corresponds with 74

**boolean** - предназначен для хранения логических значений. Переменные этого типа могут принимать только одно из 2х возможных значений true или false.

//Объявление и инициализация переменной типа boolean.

boolean b = true;

**Тип String**

Тип String не является примитивным типом данных, однако это один из наиболее используемых типов в Java. String предназначен для хранения строк текста. Несколько примеров использования String

//Создание строки с помощью конструктора

String myString = new String("The weather was fine");

//Можно также создать строку используя кавычки ""

String myString = "The weather was fine";

Для строк определен оператор «+»

public static void main(String[] args) {

String s1 = "I have ";

String s2 = " apples ";

int num = 3;

String s = s1 + num + s2;

System.out.println(s);

}

Вывод программы.

I have 3 apples

**Объектные типы**

В объектные типы входят все классы, интерфейсы, массивы. Описанный выше тип String также относится к объектным типам. Этот тип из стандартной библиотеки Java. Стандартные библиотеки содержат множество типов.

Также существуют классы-оболочки:

* Byte
* Short
* Integer
* Long
* Float
* Double
* Character
* Boolean

В отличие от примитивных типов, они пишутся с заглавной буквы. Эти типы соответствуют примитивным типам, однако являются объектовыми. Их классы cодержат методы для преобразования типов, а также другие константы и методы полезные при работе с примитивными типами данных.

**Преобразование типов в Java**

Иногда возникают ситуации, когда имея величину какого-либо определенного типа, необходимо присвоить ее переменной другого типа. С переменными и их типами мы познакомились в предыдущей лабораторной работе, в этом лабораторной работе мы рассмотрим наиболее популярные **преобразования типов в Java:**

**- Преобразование строки в число**

- String to byte

- String to short

- String to int

- String to long

- String to float

- String to double

- String to boolean

**- Преобразование числа в строку**

- int to String

- double to String

- long to String

- float to String

**- Преобразование символа char**

- char to String

- char to int

**- Преобразование чисел**

- int to long

- int to float

- long to int

- double to int

**Java преобразование числа в строку (NUMBER to STRING)**

int to String

int i = 35;

String str = Integer.toString(i);

System.out.println(str);

double to String

double i = 32.4e10;

String str = Double.toString(i);

System.out.println(str);

long to String

long i = 3422222;

String str = Long.toString(i);

System.out.println(str);

float to String

float i = 3.46f;

String str = Float.toString(i);

System.out.println(str);

**Преобразования символа char**

char to String

char ch = 'S';

// c использованием класса Character

String charToString = Character.toString(ch);

System.out.println(charToString);

// с использованием операции добавления класса String

String str = "" + ch;

System.out.println(str);

//с использованием массива

String fromChar = new String(new char[] { ch });

System.out.println(fromChar);

// с использованием метода valueOf класса String

String valueOfchar = String.valueOf(ch);

System.out.println(valueOfchar);

char to int

char ch = '9';

// c использованием метода getNumericValue

// класса Character

int i1 = Character.getNumericValue(ch);

System.out.println(i1);

// c использованием метода digit класса Character

int i2 = Character.digit(ch,10);

System.out.println(i2);

**Преобразования чисел**

int to long

int i = 2015;

long l = (long) (i);

System.out.println(l);

int to float

int i = 2015;

float f = (float) (i);

System.out.println(f);

long to int

long l = 214748364;

int i = (int) l;

System.out.println(i);

double to int

double d = 3.14;

int i = (int) d;

System.out.println(i);

В данной практической работе нужно выполнить фрагменты программного кода приведенного в данном тексте.

Также выполнить фрагменты программного приведенного в тексте лекционного занятия № 2. Типы данных и операции.

Посмотреть на результаты компиляции.

Также нужно проверить свои навыки загрузки кода программ на хостинге проектов программного обеспечения github.com

В отчете по данной практической работе привести скриншоты выполнения программного кода.

**Контрольные вопросы к защите:**

1. Перечислите правила именования переменных.
2. Назовите основные типы данных