

### ПРИМЕР-1: Простая программа на Java, вызвать в консоли текст строки.

- Написать простую Java программу, которая показывает в консоли сообщение «Welcome to Java!».
- Вне IDE консольные программы (т.е. без графического интерфейса) запускают в командной строке.
- Исходный код программы:

```
1 public class Welcome {
2     public static void main(String[] args) {
3         // Показать в консоли сообщение Welcome to Java!
4         System.out.println("Welcome to Java!");
5     }
6 }
```

- **Строка #1** – определяет **Класс** и его **Имя**
  - **class** – определяет **Класс**.
  - **Welcome** – название **Класса** **class** – каждый **Класс** имеет **Имя**, и они начинаются с заглавной буквы.
- **Строка #2** – определяет **Метод main()** – это точка входа, где программа начинает выполнение.
  - **Метод** – это конструкция, которая содержит Инструкции.
  - Метод **main()** в этой программе содержит **Инструкцию System.out.println**.
  - **Инструкция** отображает в консоли **Строку «Welcome to Java!»**.
  - **Строка (String)** – это термин в программировании, означающий последовательность символов. Строка должна быть заключена в двойные кавычки.
  - **;** – Каждая **Инструкция** заканчивается точкой с запятой «;», которая служит разделителем Инструкций.
  - **public, static, void** – **Ключевые Слова** – имеют определённое значение для Компилятора, и они не могут использоваться для других целей в программе.
- **Строка #3** – это Строчный Комментарий – кратко документирует действия программы.
- **Строка #4** – вывод на экран текстовой строку «Welcome to Java!», с переходом на новую строку.
  - **println()** – **Встроенный Метод**, которым выполняется вывод текста на экран – отображение переданной ему текстовой строки или других типов данных.
  - **System** – обозначает предопределённый класс, предоставляющий доступ к Системе.
  - **out** – поток вывода, связанный с Консолью.
  - Оператор, в котором вызывается Метод **println()**, завершается точкой с запятой «;».
  - Причина отсутствия точки с запятой в конце остальных строк кода программы состоит в том, что формально они не являются операторами.
- **Строка #5** – Первый символ «}» – завершает **Метод main()**.
- **Строка #6** – Последний символ «}» – завершает определение **Класса Name**.
- Чтобы скомпилировать «Welcome», надо запустить компилятор «**javac**», указав Имя исходного Файла в командной строке следующим образом:

```
C:\>javac Welcome.java
```

- Компилятор **Javac** создаст файл **Welcome.class**, содержащий версию **байт-кода**.
- **Байт-код** Java – промежуточное представлением программы, содержащим инструкции, которые будет выполнять виртуальная машина JVM.
- Компилятор **Javac** выдаёт результат, который не является непосредственно исполняемым кодом.

```
C:\>java Welcome
```

- При запуске «**Welcome**» – программа, выводит в Консоль сообщение:

```
Welcome to Java!
```

- Программу, легко распространить для отображения большего числа сообщений.  
Например, можно перезаписать программу для отображения 3 сообщений:

```
1 public class WelcomewithThreeMessages {
2     public static void main(String[] args) {
3         System.out.println("Программировать весело!");
4         System.out.println("Сначала основы");
5         System.out.println("Problem Driven");
6     }
7 }
```

- Скомпилировать «**WelcomeWithThreeMessages**» – запустить Компилятор «**javac**».
- Указать Имя исходного Файла в командной строке:

```
C:\>javac Welcome.java
```

- Компилятор «**javac**» создаст файл **WelcomeWithThreeMessages.class**, содержащий версию **байт-кода** – промежуточного представления программы, содержащее инструкции, для Виртуальной Машины **JVM**.
- Компилятор «**javac**» выдаёт результат, который не является непосредственно исполняемым кодом.

```
C:\>java WelcomeWithThreeMessages
```

- При запуске «**WelcomeWithThreeMessages**» – программа, выводит в Консоль сообщение:

```
Программировать весело!  
Сначала основы  
Problem Driven
```