



Пример 1: Программа "Hello World"

- Цель примера

Продemonстрировать процесс создания простейшей программы, которая выводит в консоль следующий текст:

```
Hello World
```

- Поэтапное создание программы

⚠ Важно: Инструкция не учитывает конкретную среду разработки, а описывает общий подход к решению задачи. Однако, в качестве основного инструмента разработки (IDE) предполагается IntelliJ IDEA, поэтому подробное руководство к использованию можно найти на их [официальном сайте](#).

- Шаг 1.

Необходимо создать Java класс с именем `HelloWorld` со следующим содержанием:

```
public class HelloWorld {  
  
}
```

В рамках данного примера название класса может быть произвольным и не влияет на текст, который будет выведен в консоль.

⚠ Важно: Имя файла с расширением `.java` обязано совпадать с именем класса, который описан внутри. По этой причине, создав класс в IDE, менять название класса желательно через опцию преобразования кода (*refactoring*).

⚠ Важно: Файл с расширением `.java` является специальным файлом, в котором описывается исходный код (*source code*) программы. В исходном коде может присутствовать ключевое слово `class` в заголовке программы. Его не следует путать с результатом компиляции исходного кода, которая имеет расширение файла `.class`.

- Шаг 2.

В классе `HelloWorld` добавить метод `main()` :

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
  
    }  
  
}
```

Метод `main()` является входной точкой в любую программу и все действия начинаются именно в нем.

⚠ Важно: Чтобы метод считался входной точкой в программу, обязательно соблюдать порядок ключевых слов, имя метода и входные параметры. Если одно из слов пропустить или назвать метод иначе, программа не запустится.

- Шаг 3.

Внутри метода `main()` добавить вызов метода, который выводит желаемый текст в консоль. Для данной задачи это текст *"Hello World"*:

```
public class HelloWorld {  
  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
  
}
```

Метод `println()` принимает в качестве параметра любой тип данных и выводит их на экран в консоль.

- Шаг 4.

Запустить программу и проверить консоль. Там должен следующий текст:

```
■ Hello World  
■  
■ Process finished with exit code 0
```

!! Примечание: Несмотря на то, что в исходном коде программы выводится только текст *"Hello World"*, в консоли видно два сообщения. Второе сообщение *"Process finished with exit code 0"* является системным и будет выводиться в консоль после завершения работы любой Java программы. Код 0 обозначает успешное безошибочное завершение работы программы.