



Пример 3: Программа "Random Demo"

- Цель примера

Продемонстрировать способ для генерации случайных чисел в Java.

- Поэтапное создание программы

В Java доступны несколько способов для генерации случайного числа, каждый из которых выбирается в зависимости от решаемой задачи, например требуемой степени уникальности или безопасности. В рамках данного примера будет рассмотрен только один, самый простой и часто используемый на практике метод: при помощи класса (библиотеки) `Random`.

- Шаг 1.

Необходимо создать Java класс с именем `RandomDemo` и методом `main()`:

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

- Шаг 2.

Согласно парадигме объектно-ориентированного программирования, в Java практически все является объектом. Помимо свойств, каждый объект имеет перечень действий (методов), который он может выполнять. Класс `Random` предоставляет возможность для работы со случайными числами. Чтобы начать работу с этим объектом, его необходимо проинициализировать (создать):

```
public class RandomDemo {  
  
    public static void main(String[] args) {  
  
        Random randomGenerator = new Random();  
  
    }  
  
}
```

Данная строка означает, что мы создаем объект типа `Random` с именем переменной `randomGenerator`.

⚠ Важно: Понятие классов и объектов, а также синтаксис создания объектов будет рассмотрен в дальнейшем.

– Шаг 3.

Все классы, которые входят в стандартный инструментальный JDK, расположены в различных пакетах (package). Для того, чтобы использовать класс из другого пакета – его необходимо импортировать в заголовке файла. Класс `Random` находится в пакете `java.util`, следовательно директива будет выглядеть следующим образом:

```
import java.util.Random;

public class RandomDemo {

    public static void main(String[] args) {

        Random randomGenerator = new Random();

    }

}
```

Достаточно указать полный путь всех пакетов до импортируемого класса **включительно**.

⚠ Важно: Отсутствие директивы импортирования класса является ошибкой компиляции из-за которой программа не запустится.

⚠ Важно: В IDE неимпортированные классы, как правило, подсвечиваются красным цветом.

– Шаг 4.

Сгенерировать случайное число можно с указанием диапазона или без. В случае, если указанный диапазон равен `N`, то случайное число будет сгенерировано в диапазоне от `0 ... N-1`. Например, если `N = 15`, то число будет в диапазоне от `0 ... 14` включительно. Если диапазон не указывать, то метод вернет случайное число в диапазоне типа `int`.

Для создания случайного числа необходимо вызвать метод `nextInt()` у объекта `randomGenerator`:

```
import java.util.Random;

public class RandomDemo {

    public static void main(String[] args) {

        Random randomGenerator = new Random();
        int unboundRandom = randomGenerator.nextInt(); //A
        int boundRandom = randomGenerator.nextInt(15); //B

    }

}
```

В строке **A** вызывается метод без указания диапазона. В строке **B** диапазон равен 15. В данном случае диапазон является **аргументом** метода и передается в скобках.

Результат работы каждого вызова метода присваивается в переменные `unboundRandom` и `boundRandom` соответственно. Далее их можно использовать в произвольных целях, например для расчета суммы.

- Шаг 5.

Выведем полученные числа в консоль:

```
import java.util.Random;

public class RandomDemo {

    public static void main(String[] args) {

        Random randomGenerator = new Random();
        int unboundRandom = randomGenerator.nextInt();
        int boundRandom = randomGenerator.nextInt(15);

        System.out.println("unboundRandom = " + unboundRandom);
        System.out.println("boundRandom = " + boundRandom);

    }

}
```

Результат работы программы:

```
■ unboundRandom = 1531186102
■
■ boundRandom = 10
```

!! Примечание: После запуска сгенерированные числа могут отличаться.