

Funcionamento do Math.random()

A Definição oficial diz: "*Returns a double value with a positive sign, **greater than or equal to 0.0 and less than 1.0.***" Ou seja, retorna um ponto flutuante, sendo esse um número pseudo-aleatório no intervalo [0, 1), isto é, de 0 (inclusive) até 1, mas não incluindo 1 (exclusivo). Em outras palavras, gera um número aleatório entre 0 até 0.9999999.

Dessa forma, a chamada do método Math.random() geraria a possível sequência de valores aleatórios:

```
0.5824683890332182
0.15599339785628574
0.19135110741912686
0.4996250046384343
0.0008728340690463687
```

Para gerar números aleatórios inteiros, devemos multiplicar o método por um inteiro qualquer. Por exemplo:

```
Math.random() * 10
```

Em seguida, devemos utilizar o casting (int) para pegar apenas a parte inteira da multiplicação:

```
((int)(Math.random() * 10))
```

O código acima irá gerar números aleatórios no seguinte intervalo 0, 1, 2, 3, 4, 5, 6, 7, 8, e 9. **INCLUINDO O 0, MAS EXCLUINDO O 10.**

Caso quiséssemos gerar um número entre 1 e 10, **incluindo 1 e incluindo o 10.** Deveríamos utilizar o seguinte código:

```
((int)(Math.random() * 10) + 1)
```

No código acima, estamos multiplicando `Math.random()` por 10 e transformando o valor double gerado em int e, somente depois, somamos com o valor 1, para garantir que sejam gerados números incluindo o 1 e incluindo o 10. Exemplificando:

1. se o método `Math.random()` sortearse o valor limite mínimo, 0.0, esse valor seria multiplicado por 10, que seria igual a 0.0, transformando em int ficaria apenas 0 e, após somando com 1 seria igual a 1 (limite mínimo do intervalo que se quer gerar).
2. já se o método `Math.random()` sortearse o valor limite máximo, 0.99, esse valor seria multiplicado por 10, que seria igual a 9.9, transformando em int ficaria apenas 9 e, após somando com 1 seria igual a 10 (limite máximo do intervalo que se quer gerar).

Podemos transformar nosso código em uma função onde recebemos como parâmetro o valor máximo aleatório que será incluído na geração, pois temos a soma de 1 ao final que garante isso.

```
public int getRandom(int max) {
    return ((int)(Math.random() * max) + 1)
}
```

Abaixo seguem vários exemplos com a geração de aleatórios dos mais variados padrões:

Gerar Número entre 0 (inclusive) e 1 (exclusivo)

```
public int getRandom() {  
    return ( (int) Math.random() );  
}
```

Gerar Número entre dois valores, **incluindo o mínimo e excluindo o máximo**

```
public int getRandom(int min, int max) {  
    return ( ((int) (Math.random() * (max - min))) + min );  
}
```

Gerar Número entre dois valores, **incluindo o mínimo e incluindo o máximo**

```
public int getRandom(int min, int max) {  
    return ( ((int) (Math.random() * (max - min + 1))) + min );  
}
```

-> Diferença entre Math.random() e `import java.util.Random`

As instâncias da classe Random são objetos geradores de números aleatórios sem escopo de intervalo como o Math.random().

```
import java.util.Random;  
  
public class Random1  
{  
    public static void main(String[] args)  
    {  
        //instância um objeto da classe Random usando o construtor padrão  
        Random gerador = new Random();  
  
        //imprime um número inteiro aleatório  
        System.out.println(gerador.nextInt());  
    }  
}
```

```
        //imprime um número inteiro aleatório entre 0 e 25  
        System.out.println(gerador.nextInt(26));  
    }  
}
```

A **sequência de números retornada será sempre diferente a cada execução do programa**, pois utilizamos o construtor default (sem parâmetros) da classe “Random”. As sequências são diferentes, pois quando se usa o construtor default, o Java escolhe “por conta própria” uma semente diferente a cada execução. Mas também é possível gerar **sequências fixas**, para isso é preciso fornecer a sua própria semente para o construtor:

```
import java.util.Random;  
  
public class Random3  
{  
    public static void main(String[] args)  
    {  
        //instância um objeto da classe Random especificando a semente  
        Random gerador = new Random(19700621);  
    }  
}
```

```
//imprime sequência de 10 números inteiros aleatórios entre 0 e 25
for (int i = 0; i < 10; i++) {
    System.out.println(gerador.nextInt(26));
}
}
```

A classe “Random” também fornece métodos para a geração de números reais: `nextDouble()` e `nextFloat()`. Estes métodos **não** aceitam parâmetros, sempre gerando números entre 0 e 1.

```
import java.util.Random;

public class RandomReal
{
    public static void main(String[] args)
    {
        Random r = new Random();

        System.out.println(r.nextDouble());
        System.out.println(r.nextFloat());
    }
}
```