



ORACLE

Academy



Java Foundations

4-4

A Classe Random

ORACLE
Academy



Objetivos

- Esta lição abrange os seguintes objetivos:
 - Descrever a finalidade e o uso de números aleatórios na programação Java
 - Identificar métodos da classe Random para obter números aleatórios
 - Obter números aleatórios em um intervalo de números
 - Entender a finalidade do seed de números aleatórios



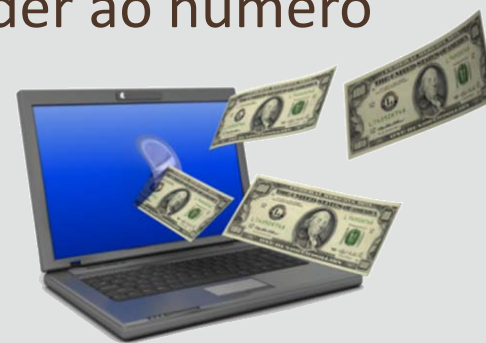
Finalidade da Geração de Números Aleatórios no Java

- Em geral, um aplicativo de software precisa executar uma tarefa com base em um valor obtido aleatoriamente
- Vários aplicativos precisam da geração de números aleatórios
- Vamos observar alguns aplicativos que usam a geração de números aleatórios



Aplicativos Baseados na Geração de Números Aleatórios

- Um aplicativo de jogos de cartas precisa embaralhar as cartas aleatoriamente e, em seguida, distribuí-las para os jogadores
- Um aplicativo de loteria precisa de um número gerado aleatoriamente que se baseie em um algoritmo
 - A pessoa ganhará se o número corresponder ao número gerado aleatoriamente



Gerando Números Aleatórios em Java

- Até o momento nas lições anteriores, você viu que o Java vem com uma variedade de classes que suportam praticamente todos os recursos básicos de desenvolvimento de aplicativos
- Por exemplo:
 - String fornece a capacidade de manipular strings
 - Scanner fornece a capacidade de obter entrada do console
- Outra classe importante no Java é a classe Random, que é usada para obter números aleatórios

O que É a Classe Random em Java?

- Em Java, você usa a classe Random para obter números aleatórios
- A classe está localizada no pacote java.util
- Ela contém vários métodos que retornam um inteiro obtido aleatoriamente e valores dos tipos double, boolean, float e long

Como Você Usa a Classe Random em um Programa Java

- Importe a classe Random do pacote java.util
- Crie uma instância da classe Random, como esta:

```
import java.util.Random;
```

A instrução import importa a classe Random do pacote java.util

```
public class RandomIntNums {  
    public static void main(String[] args) {  
        Random rndNumber = new Random();  
    } //fim do método main  
} //fim da classe RandomIntNums
```

Cria uma instância da classe Random: rndNumber

Métodos Fornecidos pela Classe Random

- Você pode obter valores aleatórios chamando os métodos a seguir fornecidos na classe Random:

Método	Produz
<code>boolean</code> <code>nextBoolean();</code>	Um valor verdadeiro ou falso
<code>int</code> <code>nextInt();</code>	Um valor integral entre <code>Integer.MIN_VALUE</code> e <code>Integer.MAX_VALUE</code>
<code>long</code> <code>nextLong();</code>	Um valor longo integral entre <code>Long.MIN_VALUE</code> e <code>Long.MAX_VALUE</code>
<code>float</code> <code>nextFloat();</code>	Um número decimal entre 0 (inclusive) e 1 (exclusive)
<code>double</code> <code>nextDouble();</code>	Um número decimal entre 0 (inclusive) e 1 (exclusive)

Como Você Obtém um Número Aleatório?

- Você pode obter um número aleatório do tipo inteiro usando o método `nextInt`
- Por exemplo:

```
import java.util.Random;
public class RandomNum {
    public static void main(String[] args) {
        Random rndNum = new Random();
        int randomNum = rndNum.nextInt();
        System.out.println("Número Aleatório: " + randomNum);
    } //fim do método main
} //fim da classe RandomNum
```

- Saída:

Random Number: 1660093261

Como Você Obtém uma Sequência de Números Aleatórios?

- Você pode obter uma sequência de números aleatórios chamando o método `nextInt` várias vezes
- Por exemplo:

```
public class RandomNumSeries {  
    public static void main(String[] args) {  
        Random num = new Random();  
        System.out.println("Número Aleatório 1: " + num.nextInt());  
        System.out.println("Número Aleatório 2: " + num.nextInt());  
        System.out.println("Número Aleatório 3: " + num.nextInt());  
        System.out.println("Número Aleatório 4: " + num.nextInt());  
        System.out.println("Número Aleatório 5: " + num.nextInt());  
    } //fim do método main  
} //fim da classe RandomNumSeries
```

`nextInt()` é chamado cinco vezes e, portanto, são gerados cinco números aleatórios





Gerando Números Aleatórios do Tipo Double

- Você pode obter números aleatórios do tipo double usando o método `nextDouble` como este:

```
public class RandomDouble {  
    public static void main(String[] args) {  
        Random num = new Random();  
        double randomDouble = num.nextDouble();  
        System.out.println("Número Aleatório: " + randomDouble);  
    } //fim do método main  
} //fim da classe RandomDouble
```

- Neste exemplo, o método `nextDouble` retorna números do tipo double no intervalo de 0 a 1

Exercício 1

- Crie um novo projeto e adicione o arquivo `FlipCoin.java`
- Examine `FlipCoin.java`:
 - Execute o programa a seguir e observe o número aleatório que pode ser gerado
 - Se a chance for $< 0,5$, grave o resultado como "superior"; caso contrário, grave-o como "inferior"
 - Repita isso muitas vezes



Gerando Números Aleatórios em um Intervalo de Números

- Até o momento, você gerou um número aleatório dentro do intervalo de um tipo de dados inteiro
- Às vezes, você pode restringir o intervalo de números que pode ser gerado
- Para implementar isso, você pode usar outra versão do método `nextInt`:
 - `nextInt(int maxValue);`
 - O argumento determina o maior número inteiro que pode ser obtido pelo método `nextInt()`
 - Você pode obter números positivos aleatórios 0 (inclusive) até um máximo (exclusive) de sua opção

Gerando Números Aleatórios em um Intervalo de Números: Exemplo

- Veja um exemplo que obtém números aleatórios no intervalo de 0 a 19:

```
public class RandomNumRange {  
    public static void main(String[] args) {  
        Random num = new Random();  
        int randomnum = num.nextInt(20);  
        System.out.println("Número Aleatório: " + randomnum);  
    } //fim do método main  
} //fim da classe RandomNumRange
```

Gerando um Intervalo a Partir de 1

- Para especificar um intervalo que comece com 1, some 1 ao resultado do método `nextInt()`
- Por exemplo, para escolher um número entre 1 e 40 inclusive, some 1 ao resultado:

```
Random rand = new Random();  
int randomnum = rand.nextInt(40) + 1;
```

Gerando um Intervalo a Partir de um Número Maior que 1

- Se o intervalo começar com um número maior que 1:
 - Subtraia o número inicial do número limite superior e some 1
 - Some o número inicial ao resultado do método `nextInt()`
- Por exemplo, para escolher um número entre 5 e 35 inclusive:
 - O número limite superior será $35 - 5 + 1 = 31$ e 5 precisa ser somado ao resultado:

```
Random rand = new Random();  
int randomnum = rand.nextInt(31) + 5;
```

Programa para Aplicativo de Loteria



```
public class Lottery {  
  
    public static void main(String[] args) {  
  
        Scanner numberScanner = new Scanner(System.in);  
        System.out.print("Informe um número entre 1 e 10: ");  
        int userNum = numberScanner.nextInt();  
        Random rnd = new Random();  
        int winningNum = rnd.nextInt(10) + 1;  
        System.out.println("Seu Número: " + userNum);  
        System.out.println("O número vencedor é: " + winningNum);  
    } // fim do método main  
  
} // fim da classe RandomNumRange
```

Exercício 2

- Crie um novo projeto e adicione o arquivo `RockPaperScissor.java` a ele
- Examine `RockPaperScissor.java`
 - Faça o seguinte:
 - Simule o jogo `RockPaperScissor` gerando um número inteiro aleatório no intervalo de 0 a 3
 - Compare o número gerado com os seguintes números:
 - se número = 0: "pedra"
 - se número = 1: "papel"
 - se número = 2: "tesoura"
 - Grave o resultado e repita muitas vezes

O Número Aleatório Gerado é o Mesmo Sempre?

- Quando você executou os exemplos anteriores várias vezes, observe que a sequência de números aleatórios é diferente a cada vez
- Às vezes, pode ser que você precise gerar sempre a mesma sequência de números aleatórios



O que É um Seed de um Número Aleatório?

- Você pode conseguir isso usando um valor constante denominado seed
- Quando você cria uma instância da classe Random, passe um inteiro constante para especificar o seed

```
Random rndNumbers = new Random(20L);
```



- Você pode alterar o seed chamando o método `setSeed()`
- Toda vez que você passa o mesmo seed, é retornada a mesma sequência aleatória



Obtendo um Sequência Aleatória Usando um Seed: Exemplo

```
public static void main(String[] args) {  
    Random rand = new Random(20L);  
    System.out.println("Número Aleatório 1: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 2: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 3: " + rand.nextInt(100));  
  
    System.out.println("Alterando o seed para alterar a sequência");  
    rand.setSeed(5L);  
    System.out.println("Número Aleatório 4: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 5: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 6: " + rand.nextInt(100));  
  
    System.out.println("Definir o seed 20 produz a sequência anterior");  
    rand.setSeed(20L);  
    System.out.println("Número Aleatório 7: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 8: " + rand.nextInt(100));  
    System.out.println("Número Aleatório 9: " + rand.nextInt(100));  
} //fim do método main
```

Alternativa para a classe Random: Math.random()

- O método `java.lang.Math.random()` retorna um número pseudoaleatório (tipo duplo) maior ou igual a 0,0 e menor que 1,0
- Quando esse método é chamado, ele cria um novo gerador de números pseudoaleatórios, o mesmo que aconteceria se a classe `Random` fosse usada
- Insira o código da nota do slide abaixo
- Execute o código várias vezes para ver como os resultados variam

Resumo

- Nesta lição, você deverá ter aprendido a:
 - Descrever a finalidade e o uso de números aleatórios na programação Java
 - Identificar métodos da classe Random para obter números aleatórios
 - Obter números aleatórios em um intervalo de números
 - Entender a finalidade do seed de números aleatórios





ORACLE

Academy

