

PROGRAMAÇÃO EM JAVA · AULA

Gerando números aleatórios com

`Math.random()`

Da teoria à prática: como usar números aleatórios em jogos, sorteios, senhas e simulações.

AULA 1

Objetivos da aula

Ao final desta aula, o aluno será capaz de:

- Compreender o conceito de números aleatórios
- Utilizar o método **Math.random()**
- Gerar números reais e inteiros aleatórios
- Gerar números dentro de um intervalo específico
- Aplicar números aleatórios em programas simples

CONCEITO

O que é um número aleatório?

Um número escolhido de forma **imprevisível** dentro de um conjunto de possibilidades.

Exemplos do cotidiano

Lançamento de um dado · Sorteio da Mega-Sena · Cara ou coroa · Bingo · Jogos eletrônicos

Por que usamos na programação?

- Criar jogos
- Fazer sorteios
- Gerar senhas
- Simular eventos
- Embaralhar listas

O MÉTODO

Math.random()

A classe **Math** possui diversos métodos matemáticos. Um deles retorna um número do tipo **double**.

Intervalo do valor gerado

`0.0 (inclusive) ≤ número < 1.0 (exclusivo)`

Ele nunca retorna exatamente 1.

EXEMPLO 1

Primeiro programa com Math.random()

```
public class Exemplo1 {  
  
    public static void main(String[] args) {  
  
        double numero = Math.random();  
  
        System.out.println(numero);  
  
    }  
  
}
```

SAÍDAS POSSÍVEIS

0.458712
0.923415
0.001245
0.777891

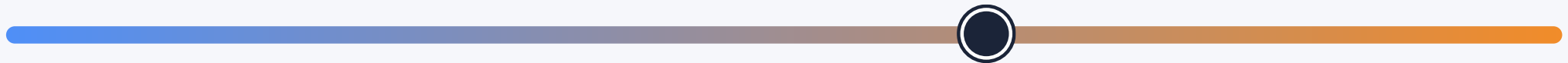
Cada execução gera um valor diferente.

VISUALIZANDO

Como funciona?

Imagine que o intervalo entre 0 e 1 seja uma régua.

0



1

`Math.random()` escolhe um ponto qualquer dessa régua.

EXPANDINDO O INTERVALO

Gerando números de 0 até 9

Podemos multiplicar o resultado:

```
double numero = Math.random() * 10;  
  
System.out.println(numero);
```

SAÍDAS POSSÍVEIS

3.45 · 8.91 · 0.23 · 9.88

Observe que ainda é um **double**.

CONVERTENDO O TIPO

Transformando em inteiro (cast)

Para obter apenas a parte inteira, usamos **cast**:

```
int numero = (int) (Math.random() * 10);  
  
System.out.println(numero);
```

SAÍDAS POSSÍVEIS

0 • 3 • 7 • 9

Agora o computador gera inteiros de **0 até 9**.

ENTENDENDO O TRUNCAMENTO

Por que isso funciona?

O resultado de `Math.random() * 10` fica entre **0.0** e **9.999999...** Ao aplicar **(int)**, a parte decimal é descartada.

Valor original	Após (int)
7.82	7
4.99	4
0.15	0

AJUSTANDO O INTERVALO

Gerando números de 1 até 10

Basta somar 1:

```
int numero = (int) (Math.random() * 10) + 1;  
  
System.out.println(numero);
```

SAÍDAS POSSÍVEIS

1 · 5 · 8 · 10

Agora o intervalo é **1 até 10**.

A FÓRMULA QUE RESOLVE TUDO

Fórmula geral

Para gerar números entre um valor mínimo e um máximo:

```
(int) (Math.random() * (max - min + 1)) + min
```

Essa é a fórmula mais utilizada em programas Java.

Exemplos com intervalos personalizados

Entre 20 e 30

```
(int) (Math.random()  
      * (30-20+1)) + 20;
```

SAÍDAS

20 • 22 • 25 • 29 • 30

Entre 100 e 999

```
(int) (Math.random()  
      * (999-100+1)) + 100;
```

SAÍDAS

152 • 987 • 301 • 745

EXEMPLOS PRÁTICOS

Lançando um dado & Cara ou coroa

Dado de 6 faces

```
int dado = (int)(Math.random() * 6) + 1;  
System.out.println(  
    "Valor do dado: " + dado);
```

SAÍDA

Valor do dado: 4

Cara ou coroa

```
int moeda = (int)(Math.random() * 2);  
  
if (moeda == 0)  
    System.out.println("Cara");  
else  
    System.out.println("Coroa");
```

EXEMPLOS PRÁTICOS

Sorteio de aluno & Senha de 4 dígitos

Sorteio de aluno

```
String[] alunos = {  
    "Ana", "Carlos", "Maria",  
    "Pedro", "Lucas"  
};  
  
int sorteado = (int) (Math.random()  
    * alunos.length);  
System.out.println(alunos[sorteado]);
```

Senha de 4 dígitos

```
int senha = (int) (Math.random()  
    * 9000) + 1000;  
System.out.println(senha);
```

SAÍDAS (sempre 1000-9999)

1258 · 8471 · 6342

ATENÇÃO

Erros comuns dos alunos

× ESQUECER O CAST

```
int numero = Math.random() * 10;
```

✓ CORRETO

```
int numero = (int)(Math.random() * 10);
```

× NOME DO MÉTODO ERRADO

```
Math.random()
```

✓ CORRETO

```
Math.random()
```

ATENÇÃO

Cast no lugar errado

× ERRADO

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

✓ CORRETO

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

Por que o errado sempre dá 0?

`Math.random()` → 0.45

`(int)` → 0

`0 × 10` → **0** (sempre)

Resumo da aula

Objetivo	Código
Número real entre 0 e 1	<code>Math.random()</code>
Inteiro entre 0 e 9	<code>(int) (Math.random() * 10)</code>
Inteiro entre 1 e 10	<code>(int) (Math.random() * 10) + 1</code>
Inteiro entre 20 e 30	<code>(int) (Math.random() * 11) + 20</code>
Inteiro entre min e max	<code>(int) (Math.random() * (max-min+1)) + min</code>

Exercícios (1 a 5)

- 1 Gere um número aleatório entre **1 e 100** e exiba na tela.
- 2 Simule o lançamento de um **dado de 6 faces**.
- 3 Simule um jogo de **cara ou coroa**.
- 4 Sorteie uma **letra entre A e Z**.
- 5 Gere uma **senha aleatória de 4 dígitos**.

Exercícios (6 a 10)

- 1 Armazene 5 nomes de alunos em um vetor e sorteie um deles.
- 2 Simule o sorteio de uma bola da loteria (**1 a 60**).
- 3 Jogo de adivinhação: sorteie um número entre 1 e 50 e informe se o palpite do usuário foi maior, menor ou igual.
- 4 Gere **20 números** entre 1 e 100 em um vetor e exiba com **Arrays.toString()**.
- 5 (**Desafio**) Preencha um vetor com 10 números aleatórios, exiba original e ordenado com **Arrays.sort()**.

FIM DA AULA

Hora de praticar!

Abra sua IDE e resolva os exercícios propostos. Bons estudos!