



INSTITUTO FEDERAL
Sudeste de Minas Gerais



Métodos static, campos static e classe Math

Curso: Técnico em Informática Integrado

Prof.: Jean Henrique de Sousa Câmara

Contato: jean.camara@ifsudestemg.edu.br

Programação Orientada a Objetos

Métodos static

2

- Às vezes, um método realiza uma tarefa que não depende de um objeto
 - Esse método é chamado de static
 - Para declarar um método como static, coloque a palavra-chave static antes do tipo de retorno na declaração do método
 - `public static int potencia(int base, int expoente)`
 - Para chamar métodos static da classe especifica-se o nome da classe na qual o método é declarado, seguido por um ponto (.) e nome do método
 - `Matematica.potencia(5, 2)`

Os métodos da classe Math

3

- A classe Math fornece uma coleção de métodos que permite realizar cálculos matemáticos comuns
- Por exemplo, você pode calcular a raiz quadrada de 900.0 com a chamada do método static

```
Math.sqrt(900.0)
```

- Não houve necessidade de criar um objeto Math antes de chamar o método sqrt
 - Todos os métodos da classe Math são static

Os métodos da classe Math

4

A classe Math faz parte do pacote java.lang, que é implicitamente importado pelo compilador, assim não é necessário importar a classe Math para utilizar seus métodos.

Método	Descrição	Exemplo
abs(x)	valor absoluto de x	abs(23.7) é 23.7 abs(0.0) é 0.0 abs(-23.7) é 23.7
ceil(x)	arredonda x para o menor inteiro não menor que x	ceil(9.2) é 10.0 ceil(-9.8) é -9.0
cos(x)	cosseno trigonométrico de x (x em radianos)	cos(0.0) é 1.0
exp(x)	método exponencial e^x	exp(1.0) é 2.71828 exp(2.0) é 7.38906
floor(x)	arredonda x para o maior inteiro não maior que x	floor(9.2) é 9.0 floor(-9.8) é -10.0
log(x)	logaritmo natural de x (base e)	log(Math.E) é 1.0 log(Math.E * Math.E) é 2.0
max(x, y)	maior valor de x e y	max(2.3, 12.7) é 12.7 max(-2.3, -12.7) é -2.3

continua

Os métodos da classe Math

5

continuação

Método	Descrição	Exemplo
<code>min(x,y)</code>	menor valor de x e y	<code>min(2.3, 12.7)</code> é 2.3 <code>min(-2.3, -12.7)</code> é -12.7
<code>pow(x,y)</code>	x elevado à potência de y (isto é, x^y)	<code>pow(2.0, 7.0)</code> é 128.0 <code>pow(9.0, 0.5)</code> é 3.0
<code>sin(x)</code>	seno trigonométrico de x (x em radianos)	<code>sin(0.0)</code> é 0.0
<code>sqrt(x)</code>	raiz quadrada de x	<code>sqrt(900.0)</code> é 30.0
<code>tan(x)</code>	tangente trigonométrica de x (x em radianos)	<code>tan(0.0)</code> é 0.0

Variáveis static

6

- Vimos que cada objeto de uma classe mantém sua própria cópia de cada variável de instância da classe
 - Existem variáveis para as quais cada objeto de uma classe não precisa de sua própria cópia separada
 - Essas variáveis são declaradas static, também conhecidas como **variáveis de classe**
 - Todos os objetos dessa classe compartilham uma cópia dessas variáveis

Constantes PI e E da classe Math static

7

- A classe Math declara duas constantes
 - Math.PI
 - Relação entre a circunferência de um círculo e seu diâmetro
 - Math.E
 - Valor da base para logaritmos naturais
 - Representam aproximações de alta precisão a constantes matemáticas comumente usadas
- Qualquer campo declarado com a palavra-chave **final** é constante
 - Seu valor não pode ser alterado depois que o campo é inicializado

Concatenação de strings

8

- O Java permite montar objetos String em strings maiores utilizando os operadores + ou +=
 - Por exemplo, a expressão "Olá " + "mundo!" cria a String "Olá mundo!"
- Se um dos operandos do operador + for uma String, o outro é convertido em uma String e então os dois são concatenados
- Um boolean concatenado com uma String é convertido na String "true" ou "false"

Estudo de caso: geração segura de números aleatórios

9

- O elemento de acaso pode ser introduzido em um programa por meio de um objeto da classe `SecureRandom` (pacote `java.security`)
 - Esses objetos podem produzir valores aleatórios boolean, byte, float, double, int, long e gaussianos
- Antigamente usava-se a classe `Random` para obter valores “aleatórios”
 - Essa classe produzia valores determinísticos que poderiam ser previstos por programadores mal-intencionados
- Objetos `SecureRandom` produzem números aleatórios não determinísticos que não podem ser previstos

Estudo de caso: geração segura de números aleatórios

10

- Para obter um número inteiro aleatório primeiro temos que instanciar um objeto gerador seguro

```
SecureRandom randomNumbers = new SecureRandom();
```

- Agora deve se usar o método nextInt()

```
int randomValue = randomNumbers.nextInt();
```

- Esse método obtém um número inteiro aleatório

Estudo de caso: geração segura de números aleatórios

11

- Para alterar o intervalo de valores produzido por `nextInt` basta informar um valor `int` como argumento

```
int randomValue = randomNumbers.nextInt(2);
```

- Gera aleatoriamente 0 ou 1
- Caso queira simular o lançamento de um dado basta fazer

```
int face = randomNumbers.nextInt(6);
```

Hora de praticar!

12

```
3 import java.security.SecureRandom; // o programa usa a classe SecureRandom
4
5 public class RandomIntegers
6 {
7     public static void main(String[] args)
8     {
9         // o objeto randomNumbers produzirá números aleatórios seguros
10        SecureRandom randomNumbers = new SecureRandom();
11
12        // faz o loop 20 vezes
13        for (int counter = 1; counter <= 20; counter++)
14        {
15            // seleciona o inteiro aleatório entre 1 e 6
16            int face = 1 + randomNumbers.nextInt(6);
17
18            System.out.printf("%d ", face); // exibe o valor gerado
19
20            // se o contador for divisível por 5, inicia uma nova linha de saída
21            if (counter % 5 == 0)
22                System.out.println();
23        }
24    }
25 } // fim da classe RandomIntegers
```

Exercício

13

- Faça o programa para simular o lançamento de um dado de 6 faces 6000000 de vezes. No final seu programa deve exibir quantas vezes cada face apareceu. Ex:

Face	Frequência
1	999647
2	999557
3	999571
4	1000376
5	1000701
6	1000148

Exercício (2 pontos)

14

- Um jogo popular de azar é um jogo de dados conhecido como craps, que é jogado em cassinos e nas ruas de todo o mundo. As regras do jogo são simples e diretas:
 - Você lança dois dados. Cada dado tem seis faces que contêm um, dois, três, quatro, cinco e seis pontos, respectivamente. Depois que os dados param de rolar, a soma dos pontos nas faces viradas para cima é calculada. Se a soma for 7 ou 11 no primeiro lance, você ganha. Se a soma for 2, 3 ou 12 no primeiro lance (chamado “craps”), você perde (isto é, a “casa” ganha). Se a soma for 4, 5, 6, 8, 9 ou 10 no primeiro lance, essa soma torna-se sua “pontuação”. Para ganhar, você deve continuar a rolar os dados até “fazer sua pontuação” (isto é, obter um valor igual à sua pontuação). Você perde se obtiver um 7 antes de fazer sua pontuação.

Exercício (2 pontos)

15

- Faça um programa que simule o jogo de dados craps. Veja o exemplo de saídas

```
Player rolled 5 + 6 = 11  
Player wins
```

```
Player rolled 5 + 4 = 9  
Point is 9  
Player rolled 4 + 2 = 6  
Player rolled 3 + 6 = 9  
Player wins
```

```
Player rolled 1 + 2 = 3  
Player loses
```

```
Player rolled 2 + 6 = 8  
Point is 8  
Player rolled 5 + 1 = 6  
Player rolled 2 + 1 = 3  
Player rolled 1 + 6 = 7  
Player loses
```

Dúvidas?

16



jean.camara@ifsudestemg.edu.br