

Técnico em Desenvolvimento de Sistemas

Lógica de Programação II

Classes Math e Random

Alex Helder Cordeiro do Rosário de Oliveira

Instituto Federal de Brasília - *Campus Brasília*

Motivação

- Uma das aplicações mais comuns da informática é a resolução de cálculos ou processamentos matemáticos.
- É muito comum as linguagens de programação (especialmente as estruturadas ou orientadas a objetos) terem bibliotecas para realização de funções matemáticas.
- Em Java, os cálculos matemáticos básicos podem ser encontradas na classe `java.lang.Math`.
- O acesso aos métodos ou atributos da classe `java.lang.Math` são feitos diretamente pela classe, isto é, não é necessário* criar um objeto dela para acessar os seus membros.

*Nem é permitido também; Sua construtora é declarada “privada” para não ser instanciada; e a classe é declarada como “final” para que não seja herdada. Todos os seus membros são estáticos.

Atributos

- A classe Math possui dois atributos que são constantes tipo double:

`Math.PI`: $\pi \cong 3.141592653589793$

`Math.E`: Número de Euler: $e \cong 2.718281828459045$

Métodos

- Dentre outros, serão encontrados os seguintes métodos:
 - double `abs(double a)`: Retorna o valor absoluto (sem o sinal);
 - double `round(double a)`: Arredondamento. Retorna o double com valor inteiro mais próximo de a;
 - double `ceil(double a)`: Arredondamento para cima. Retorna o double com menor valor inteiro maior que a;
 - double `floor(double a)`: Arredondamento para baixo. Retorna o double com maior valor inteiro menor que a;
 - double `max(double a, double b)`: Retorna o maior entre a e b;
 - double `min(double a, double b)`: Retorna o menor entre a e b;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros da classe Math; ex.: `Math.abs(a)`.

Métodos

- Dentre outros, serão encontrados os seguintes métodos:
 - `double sqrt(double a)`: Retorna a raiz quadrada de `a`;
 - `double cbrt(double a)`: Retorna a raiz cúbica de `a`;
 - `double pow(double a, double b)`: Retorna `a` elevado a `b`;
 - `double exp(double a)`: Retorna o número de Euler e elevado a `a`;
 - `double log(double a)`: Retorna o logaritmo de `a` na base e (número de Euler);
 - `double log10(double a)`: Retorna o logaritmo de `a` na base 10;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros da classe `Math`; ex.: `Math.sqrt(a)`.

Métodos

- Dentre outros, serão encontrados os seguintes métodos:
 - double `sin(double a)`: Retorna o seno de a em radianos;
 - double `cos(double a)`: Retorna o cosseno de a em radianos;
 - double `tan(double a)`: Retorna a tangente de a em radianos;
 - double `sinh(double a)`: Retorna o seno hiperbólico de a;
 - double `cosh(double a)`: Retorna o cosseno hiperbólico de a;
 - double `tanh(double a)`: Retorna a tangente hiperbólico de a;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros da classe Math; ex.: `Math.sin(a)`.

Métodos

- Dentre outros, serão encontrados os seguintes métodos:
 - double `asin(double a)`: Retorna o arco-seno de a em radianos;
 - double `acos(double a)`: Retorna o arco-cosseno de a em radianos;
 - double `atan(double a)`: Retorna o arco-tangente de a em radianos;
 - double `toDegrees(double a)`: Converte a de radianos para graus;
 - double `toRadians(double a)`: Converte a de graus para radianos;
 - double `hypot(double a, double b)`: Retorna a hipotenusa de um triângulo retângulo de catetos a e b: $\sqrt{a^2 + b^2}$;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros da classe Math; ex.: `Math.toDegrees(a)`.

Como calcular outras raízes diferentes de 2 ou 3?

- Elevando a números fracionários.

$$\sqrt[x]{b} \equiv b^{\frac{1}{x}}$$

- Exemplo: Realizando a raiz quarta de 81:

```
double raiz = Math.pow(81, 1.0/4);
```


Como calcular logaritmos em outras bases?

- Efetuando a divisão dos logaritmos:

$$\log_x a \equiv \frac{\log_q a}{\log_q x}$$

- Exemplo: Calcular o \log_2 de 512:

```
double resultado = Math.log(512)/Math.log(2);
```

Sua vez:

- 13 Implemente um programa que auxilie os profissionais de química a estimarem a quantidade de substância radioativa que permanece no decorrer do tempo. O usuário deve entrar a quantidade inicial da substância, a taxa de desintegração da substância e o tempo que se passou; e deve apresentar ao usuário a quantidade que ainda sobrou da substância.

Utilize a seguinte expressão

$$Q = Q_o \times e^{-rt}$$

, onde Q é a massa final, Q_o é a massa inicial, r é a taxa e t é o tempo em anos.

Métodos

- A classe `Math` também tem um método para calcular números aleatórios:
 - `double random();`
- Este método é considerado limitado;
- A recomendação geral para a geração de *números aleatórios* é utilizar a classe `Random`.

Geração de Número Aleatório

- A geração de números aleatórios ou de sequências de números aleatórios é extremamente importante na informática, sendo usada em:
 - jogos;
 - criptografia e segurança;
 - data mining;
 - estatísticas;
 - teste de software;
 - e muitos outros.

Random

- Para obter números aleatórios em Java, criamos um objeto da classe Random e usamos algum método para gerar o próximo número da sequência.

```
// Criação do objeto  
Random geradorDeAleatorios = new Random();  
// gerando um número inteiro aleatório  
int aleatorio = geradorDeAleatorios.nextInt();
```

Métodos para geração de números aleatórios

- Dentro da classe Random, temos os seguintes métodos que geram números aleatórios:
 - `int nextInt()`: Retorna o próximo valor inteiro da sequência aleatória;
 - `int nextInt(int limite)`: Retorna o próximo valor inteiro menor que o limite da sequência aleatória;
 - `long nextLong()`: Retorna o próximo valor long da sequência aleatória;
 - `boolean nextBoolean()`: Retorna o próximo valor booleano da sequência aleatória;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros do objeto.

Métodos para geração de números aleatórios

- Dentro da classe Random, temos os seguinte métodos que geram números aleatórios:
 - void `nextBytes(byte[] bytes)`: Armazena no array bytes os próximos bytes da sequência aleatória;
 - double `nextDouble()`: Retorna o próximo valor double entre 0.0 e 1.0 da sequência aleatória;
 - float `nextFloat()`: Retorna o próximo valor float entre 0.0 e 1.0 da sequência aleatória;
 - double `nextGaussian()`: Retorna o próximo valor double da sequência aleatória com distribuição gaussiana (normal) com média 0.0 e desvio padrão 1.0;

*Devemos lembrar que para qualquer destes métodos devem ser acessados como membros do objeto.

Números Pseudo-Aleatórios

- O computador é um conjunto de circuitos determinísticos;
- Não é possível gerar um valor (ou sequência de valores) verdadeiramente aleatório;
- É possível entretanto gerar uma sequência de números que, para os usuários pareça com uma sequência de números aleatórios.
- Esta é chamada de uma sequência de números *pseudo-aleatórios**.
- A sequência de números pseudo-aleatórios é obtida pela execução de diversas funções matemáticas sobre um número, chamado de *semente*.

*Pseudo: do grego, o que finge ser ou tenta se passar pelo que não é.

Sementes

- Cada objeto da classe `Random` tem sua própria semente para gerar a sequência pseudo-aleatória.
- Se dois objetos da classe `Random` tiverem a mesma semente, eles devem gerar sequências idênticas de números pseudo-aleatórios.
- “Qual a utilidade disso?”

Sementes

- Cada objeto da classe `Random` tem sua própria semente para gerar a sequência pseudo-aleatória.
- Se dois objetos da classe `Random` tiverem a mesma semente, eles devem gerar sequências idênticas de números pseudo-aleatórios.
- “Qual a utilidade disso?”
- Permitir replicar contextos específicos para comparações (em simulações, por exemplo);
- Auditoria (confirmação de um sorteio aleatório, por exemplo);
- etc.

Sementes

- O construtor padrão (sem argumentos) cria o objeto com semente esperadamente diferente da criação de outros objetos criados pelo mesmo padrão.
- Se for interesse do programador, ele pode usar o construtor passando, como parâmetro, a semente (número inteiro de 64 bits - long) que será usada pelo objeto para gerar a sequência pseudo-aleatória.
- Outra forma de definir a semente é através do método `setSeed(long semente)`.

Sua vez:

- 14 Escreva um programa que gere uma senha aleatória composta por 6 dígitos.