Fatec
Mogi Mirim
Arthur de Azevedo

PROGRAMAÇÃO ORIENTADA A OBJETOS [POO]

Material 05 – POO_05

Prof. Mestre Marcos Roberto de Moraes [Maromo]

Sobrecarga, APIs de Java, construtores, variável this

Agenda

- Construtores
- Sobrecarga
- Variável **this**
- Classe **Math**
- Comando **package** / Pacotes da API do Java
- Enumeradores
- Exercícios / Desafio

Construtores

- Quando usamos a palavra reservada **new**, estamos **construindo um objeto.**
- O **construtor é um bloco que possui o mesmo nome da classe.**
- É importante **observar que quando não criamos explicitamente um construtor o compilador o fará automaticamente.**
- Toda classe necessita de pelo menos um construtor (**seja implícito ou não**).
- Pode-se ter mais do que um método construtor (overload), com assinaturas diferentes.

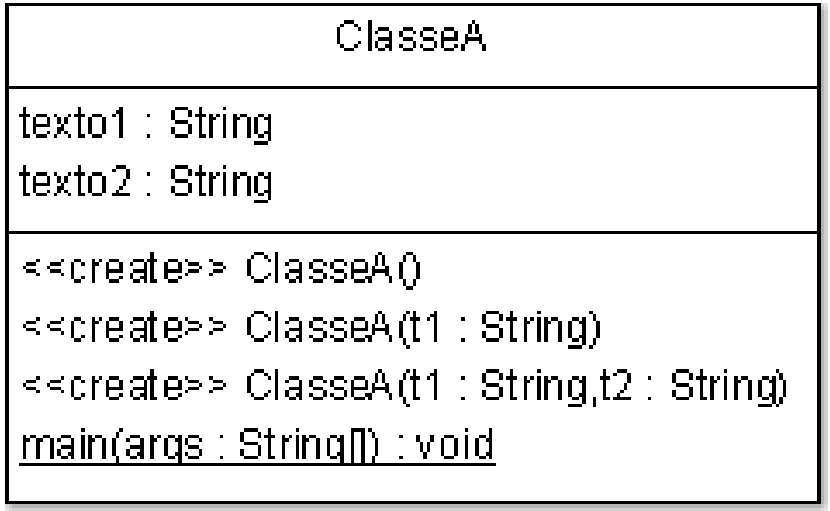
Sobrecarga (overload)

- É a capacidade de definir mais de um método com o mesmo nome. No entanto sua assinatura deve ser diferente. Ex:

```
public class ExemploOverload {
    void imprimir(){
        System.out.println("Método imprimir - void");
    }
    int imprimir(int a){
        int num=a;
        return num;
    }
    int imprimir(int a, int b){
        return a+b;
    }
    public static void main(String[] args) {
        //Exemplo de chamada de métodos sobrecarregados.
        ExemploOverload obj = new ExemploOverload();
        System.out.println("Conceituando Overload (sobrecarga)");
        int a = 3;
        int b = 7;
        obj.imprimir();
        System.out.println("Valor de A: " + obj.imprimir(a));
        System.out.println("Valor da soma de A e B: " + obj.imprimir(a, b));
    }
}
```

Sobrecarga de Métodos Construtores

- Considere a classe ClasseA abaixo:



ClasseA

```java
import java.util.Scanner;
public class ClasseA {
    //Membros públicos
    public String texto1;
    public String texto2;
    //Construtor 1
    public ClasseA(){
        texto1 = "Primeiro Texto\n";
        texto2 = "Segundo Texto\n";
    }
    //Construtor 2 (overload)
    public ClasseA(String t1){
        texto1 = t1;
        texto2 = "";
    }
    //Construtor 3 (overload)
    public ClasseA(String t1, String t2){
        texto1 = t1;
        texto2 = t2;
    }
```

```java
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Digite o número de parâmetros [0,1,2]: ");
    int qtd = sc.nextInt();
    ClasseA obj=null;
    if (qtd==0){
        obj = new ClasseA();
    }else if(qtd==1) {
        String t;
        sc = new Scanner(System.in);
        System.out.printf("Digite um texto: \n");
        t = sc.nextLine();
        obj = new ClasseA(t);
    }else if(qtd==2){
        System.out.println("Digite dois textos: ");
        sc = new Scanner(System.in);
        String t = sc.nextLine();
        String t1 = sc.nextLine();
        obj = new ClasseA(t,t1);
    }
    System.out.println("Texto 1: " + obj.texto1);
    System.out.println("Texto 2: " + obj.texto2);
}
```

Nota sobre o exemplo anterior

- A ClasseA possui três métodos construtores 1 + (2 Overload).
- Quando um método sobrecarregado é chamado, o compilador Java seleciona o método adequado examinando a assinatura do mesmo, na chamada.

Variável **this**

- É uma referência ao próprio objeto, usa-se em:
  - Dentro de um construtor, podemos executar outro construtor com assinatura diferente.
  - Resolver ambiguidade de nome entre um atributo e um parâmetro ou variável (caso mais comum de uso).
  - Retornar a própria referência da instância em algum método.

Comando package

- Representa um grupo de elementos Java, como classes, interfaces, enumeradores e anotações.
- Quando agrupados permitem uma melhor organização dos inúmeros programas que podemos manipular.
- Regra de nomenclatura: usar letras minúsculas.

Comando import

- Comando import
- Permite que um programa Java use elementos pertencentes a um pacote já existente.
- Usado em conjunto com a palavra reservada static, ele pode ser usado sem o qualificador, ou seja, sem o nome da classe onde foi definido.
- Ex:

Exemplo: comando import

Programa: ExemploImport

```
import static java.lang.Math.*;
import static java.lang.System.*;
/**
 * Uso do método estático sqrt
 * para exibir a raiz quadrada da constante PI
 *
 */
public class Modulo03ExemploImport {

    public static void main(String[] args) {
        out.println(sqrt(PI));
    }
}
```

Classe Math

- Possui vários métodos estáticos (***static)***, que permitem realizar cálculos matemáticos comuns.
- Faz parte do pacote **java.lang**, que é implicitamente importado pelo compilador. Não é necessário importar a **classe Math** para utilizar os seus métodos.

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 )`	co-seno 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
`min( x, y )`	menor valor de *x* e *y*	`min( 2.3, 12.7 )` é 2,3 `min( -2.3, -12.7 )` é -12,7
`pow( x, y )`	*x* elevado à potência de *y* (isto é, x^y)	`pow( 2.0, 7.0 )` é 128,0 `pow( 9.0, 0.5 )` é 3,0
`sin( x )`	seno trigonométrico de *x* (*x* em radianos)	`sin( 0.0 )` é 0,0
`sqrt( x )`	raiz quadrada de *x*	`sqrt( 900.0 )` é 30,0
`tan( x )`	tangente trigonométrica de *x* (*x* em radianos)	`tan( 0.0 )` é 0,0

DEITEL(2009, pg. 167)

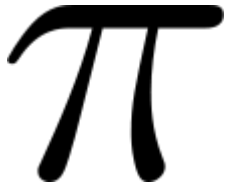
Classe: ClasseMath

```
public class ClasseMath {
    public static void main(String[] args) {
        //Exemplo de Métodos da Classe Math
        System.out.printf("Coseno de 45 (45 radianos): %2.2f\n",  Math.cos(45));
        System.out.printf("Seno de 45    (0 radianos): %2.2f\n", Math.sin(45));
        System.out.printf("Tangente de 0(0 radianos) : %2.2f\n", Math.tan(0));
        System.out.printf("Raiz quadrada de 25       : %2.2f\n", Math.sqrt(900.0));
        System.out.printf("4 elevado ao qudrado      : %2.2f\n", Math.pow(4,2));
        System.out.printf("Menor entre 3 e 7         : %2d\n", Math.min(3,7));
        System.out.printf("Maior entre 3 e 7         : %2d\n", Math.max(3,7));
        System.out.printf("Valor absoluto de -2,1    : %2.2f\n", Math.abs(-2.1));
    }
}
```

```
run:
Coseno de 45 (45 radianos): 0,53
Seno de 45    (0 radianos): 0,85
Tangente de 0(0 radianos) : 0,00
Raiz quadrada de 25       : 30,00
4 elevado ao qudrado      : 16,00
Menor entre 3 e 7         :  3
Maior entre 3 e 7         :  7
Valor absoluto de -2,1    : 2,10
```

Constantes Math.PI e Math.E

- Constantes matemáticas



Math.PI
3,14159...

Math.E
2,718281...

Constante PI é a relação entre a circunferência de um círculo e o seu diâmetro.
Constante E é o valor da base para logaritmos naturais.

Pacotes da API do JAVA

- Java contém muitas classes que são agrupadas em categorias de classe relacionadas denominadas pacotes.
- Esses pacotes são chamados de Java API(Java Application Programming Interface) ou bibliotecas de classes Java.
- Como vimos, durante os exercícios para especificar as classes exigidas para compilar um programa usamos o comando import.
- Ex:
  - **import java.util.Scanner;**

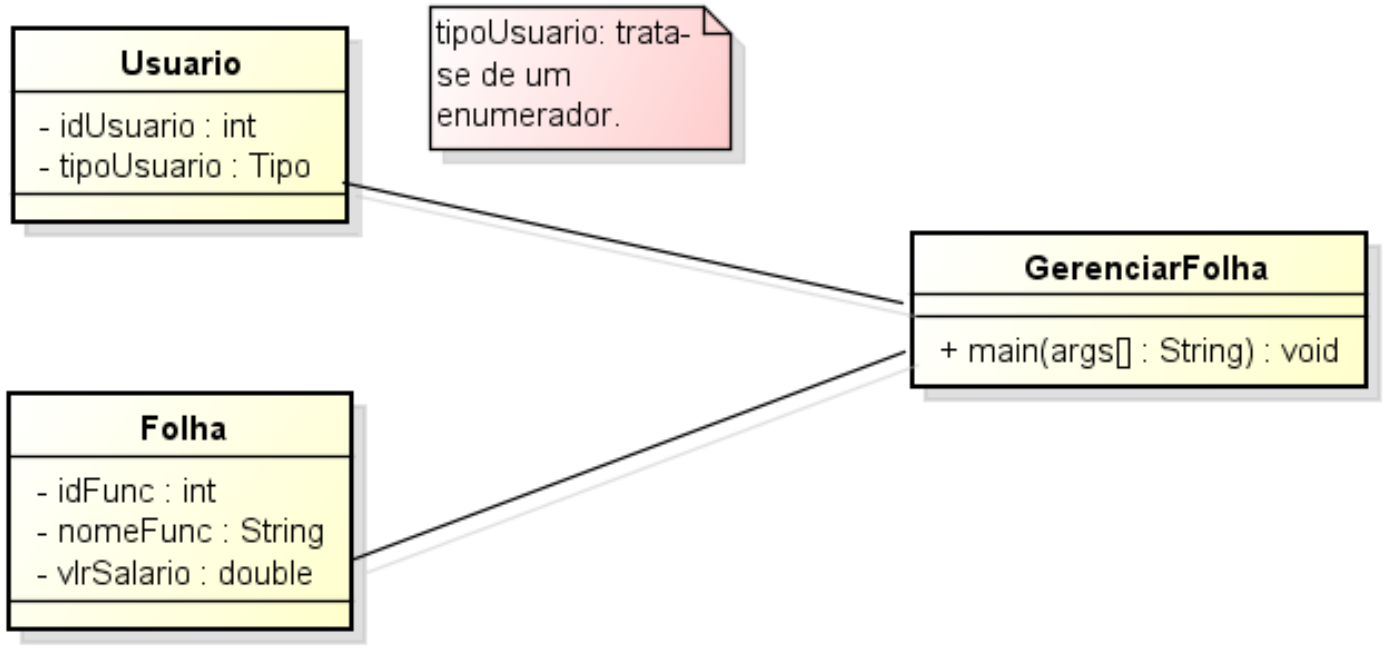
Pacotes da API do Java

- Para ter uma visão geral dos pacotes da API, acesse:
- https://docs.oracle.com/en/java/javase/11/docs/api/index.html

Enumeradores (enum)

- Java nos possibilita criar uma estrutura de dados enumarada.
- Essas estruturas de dados enumeradas são conjuntos de constantes organizados em ordem de declaração, ou seja, o que é declarado primeiro virá primeiro.
- A funcionalidade principal de ***enum*** é agrupar valores com o mesmo sentido dentro de uma única estrutura.
- No próximo exemplo, criou-se uma estrutura de enumerador para representar os tipos de usuários possíveis.

Aplicação: PrjFolhaPagamento



Enumerador
Tipo

```
package prjfolhapagamento;

public enum Tipo {
        operador,
        supervisor,
        admin
}
```

Classe: Usuario

Métodos de acesso com modificadores públicos. Acesso aos campos encapsulados

```
package prjfolhapagamento;

public class Usuario {

    private int idUsuario;
    private Tipo tipoUsuario;

    public int getIdUsuario() {
        return idUsuario;
    }

    public void setIdUsuario(int idUsuario) {
        this.idUsuario = idUsuario;
    }

    public Tipo getTipoUsuario() {
        return tipoUsuario;
    }

    public void setTipoUsuario(Tipo tipoUsuario) {
        this.tipoUsuario = tipoUsuario;
    }

}
```

Classe: Folha

Regra de acesso a leitura do campo. É passado o tipo de usuário que está acessando o campo para leitura. Caso seja tipo.admin, poderá ler o valor do Salário

Só pode definir o salário (atribuir valor) se o usuário for administrador.

```
package prjfolhapagamento;
public class Folha {
    private int idFunc;
    private String nomeFunc;
    private double vlrSalario;

    public int getIdFunc() {
        return idFunc;
    }
    public void setIdFunc(int idFunc) {
        this.idFunc = idFunc;
    }
    public String getNomeFunc() {
        return nomeFunc;
    }
    public void setNomeFunc(String nomeFunc) {
        this.nomeFunc = nomeFunc;
    }
    public double getVlrSalario(Usuario usuario) {
        if (usuario.getTipoUsuario() == Tipo.admin)
            return vlrSalario;
        else
            throw new IllegalArgumentException("Acesso não permitido");
    }
    public void setVlrSalario(double vlrSalario, Usuario usuario) {
        if(usuario.getTipoUsuario()==Tipo.admin)
            this.vlrSalario = vlrSalario;
        else
            throw new IllegalArgumentException("Sem permissão de alteração");
    }
}
```

Laboratório

Prática de Conceitos OO – Estudo de Caso e Exercício

Estudo de Caso:
Jogo Tabuleiro da Sorte

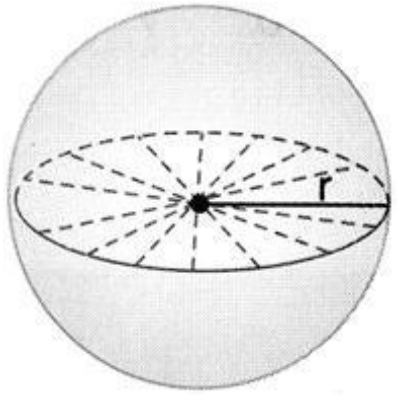
Tempo estimado:
1 hora.

- A loteca Boa Sorte para entreter os seus clientes necessita de um programa em Java que enquanto os clientes aguardam na fila possam se distrair jogando no tabuleiro da sorte.
  - O Tabuleiro da sorte é um jogo simples, que consistem em uma matriz de 10 x 10 [Bi dimensional] que cada elemento é um número inteiro entre 0 e 100.
  - Esses elementos são gerados aleatoriamente e o usuário deve digitar dois números também entre 0 e 100.
  - A regra para a vitória e prêmio:
    - Caso um dos números digitados pelo usuário tenha aparecido 3 ou mais vezes no tabuleiro o usuário é um feliz ganhador.
    - Prêmio: o valor é de R$ 1000,00 por número encontrado, supondo que no tabuleiro seja encontrado quatro números iguais a um dos que o usuário tenha digitado, o seu prêmio é de R$ 4.0000,00.
  - Sua missão: desenvolver um programa em Java (Orientado a Objetos) que solucione este cenário.

Tempo estimado:
20 minutos.

Exercício prático

- Você deve criar um programa orientado a objetos que dado o valor de um raio pelo usuário, seja calculado e devolvido o valor do volume de uma esfera.
- Realize as abstrações necessárias e crie a classe de modelagem com atributos e métodos que ache pertinente.



O volume de uma esfera de raio r é igual a $\frac{4}{3}\pi r^3$.

$$V = \frac{4}{3}\pi r^3$$

Desafio com Enumeradores e List<>

Prática de Conceitos OO – Estudo de Caso e Exercício

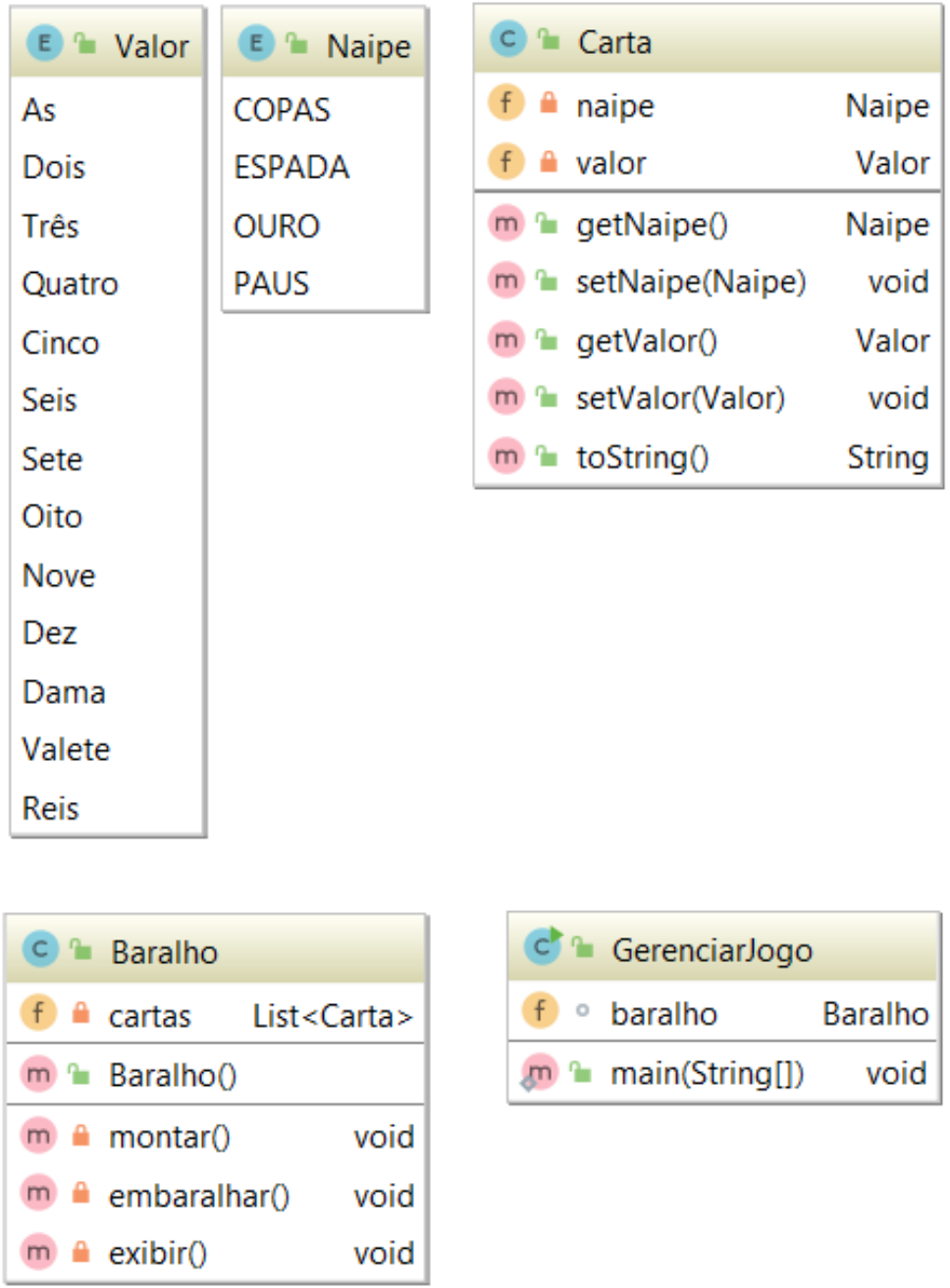
Observe a figura

Crie um novo Projeto Chamado **ProjetoBaralho**

1) Modele conforme figura ao lado. Nela temos dois enumeradores Valor e Naipe, Uma Classe carta que é composta de Naipe e Valor, e um Classe Baralho que contém uma Lista de Cartas.

2) Seu desafio:
   1) Monte um baralho com 52 cartas;
   2) Embaralhe essas cartas;
   3) Exiba o baralho já embaralhado.

3) //////suba no git hub com comentários para pontuação extra.



Referências

- Bibliográficas:
  - Mendes – Java com Ênfase em Orientação a Objetos [Exercícios do Capítulo 1]
  - Deitel – Java, como programar – 6º edição.
  - Arnold, Gosling, Holmes – A linguagem de programação Java – 4º edição.
  - Apostilas Caelum
  - Material do Curso de Capacitação Java do CPS
- Internet
  - http://java.sun.com
  - http://www.guj.com.br
  - http://www.portaljava.com

FIM

Maromo