Aula 04 Introdução à Programação

Antes de iniciar...

Dúvidas ???



Modularização Antes de POO...

P00

Programação Orientada a Objetos



Como organizar?

emerson@paduan.pro.br



Características comuns

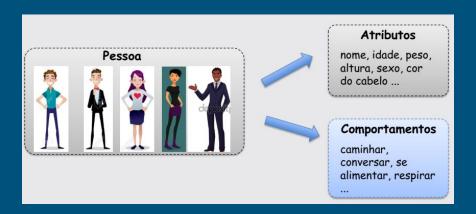
Classe

Uma classe é um modelo que define, especifica um objeto. É uma abstração (representação) dos objetos.

Ela define os dados (ATRIBUTOS) e os comportamentos (MÉTODOS) do Objeto.

emerson@paduan.pro.bi

Exemplo



ATENÇÃO

Uma classe é um MODELO!

Não se coloca dados ou se utiliza diretamente uma classe.

É necessário criar um objeto a partir da classe



emerson@paduan.pro.br

Exemplo

TODOS os objetos criados a partir da classe, possuem os mesmos atributos e métodos, mas com valores diferentes.











Exemplo

```
class Pessoa {
    String nome;

    void apresentar(){
        System.out.println("Olá! Eu sou " + nome);
    }
}
```

```
class Exemplo{
    main () {
    Pessoa p = new Pessoa();

    p.nome = "Emerson";
    p.apresentar();
}
```

emerson@paduan.pro.b

Construtores

São métodos especiais utilizados para inicialização dos atributos de um objeto no "momento" da criação do objeto.

Detalhes:

Os construtores podem ter ou não parâmetros O nome do construtor DEVE ser o mesmo da classe Construtores NÃO possuem valor de retorno

Exemplo

```
public class Pessoa {
   String nome;
   float salario;

public Pessoa(String nome, float salario){
   this.nome = nome;
   this.salario = salario;
}

public void exibir(){
   System.out.println("Pessoa: " + nome + ": R$ " + salario);
}

class Exemplo{
   main () {
    Pessoa p = new Pessoa("Marcos", 5000 );
   p.exibir();
}

public void exibir(){
   System.out.println("Pessoa: " + nome + ": R$ " + salario);
}
```

Exercício 4.1



Crie a classe veículo, com os atributos modelo, marca e consumo (quantos km/l).

Faça um construtor para inicializar os atributos da classe.

Escreva um método para exibir os dados do carro (modelo, marca) e outro para retornar o valor do consumo.

Faça o main para testar a classe criada.

Encapsulamento



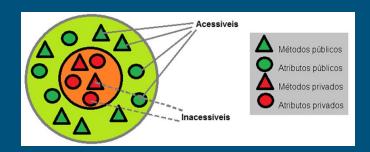
"Esconder / Proteger"

emerson@paduan.pro.bi

Conceito

É necessário garantir que os dados (atributos) dos objetos não sejam modificados de forma indevida.

A modificação de forma indevida interfere no funcionamento do objeto e pode gerar erros.



Modificadores

Modificador	Funcionalidade
public	permite que qualquer outra parte da aplicação tenha acesso ao membro
Padrão (default)	membros que não foram marcados com nenhum modificador explicitamente. Só podem ser acessados por outras classes dentro do mesmo pacote
protected	os membros são acessíveis por classes dentro do mesmo pacote e por classes derivadas (mesmo em pacotes diferente)
private	só é acessível dentro da própria classe em que foi declarado

emerson@paduan.pro.bi

get / set

Em Java utilizamos nomes de métodos iniciando com get ou set para indicar métodos que alteram ou obtém valores de atributos privados.

Exemplo:

getHora: obtém o valor do atributo hora

setHora: altera o valor do atributo hora

Exercício 4.2 - Relógio

Escreva a classe Relógio, com os atributos hora, minuto e segundo, e com um construtor que recebe horas, minutos e segundos para inicializar o relógio. Faça um método da classe para exibir a hora atual.

Faça um programa (main) que crie um objeto do tipo Relógio e exiba a hora atual do relógio.

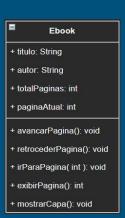


emerson@paduan.pro.bı

Exercício 4.3 - Ebook

Considere que você está desenvolvendo um aplicativo de livro eletrônico. Os atributos e métodos estão no diagrama. As funcionalidades incluem exibir em que página está agora, para que página foi, etc.

Escreva uma classe para implementar estas funcionalidades e faça o app (main) para testar essa navegação entre as páginas.



Exercício 4.4 - Robô

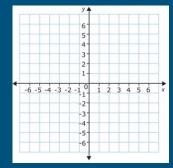


Criar a classe Robô que permite um robô ser navegado pelas direções: 'N', 'S', 'L' ou 'O'. A classe possui coordenadas x, y que indicam a posição atual do robô no plano cartesiano, e um valor máximo no plano até o qual ele pode ser movido.

A cada chamada do método para mover o robô, ele deve se deslocar uma unidade.

Deve-se verificar se o movimento do robô é válido.

OBS: crie os métodos necessários!



emerson@paduan.pro.b

Coleções



Classes para ED em Java





emerson@paduan.pro.br

ArrayList

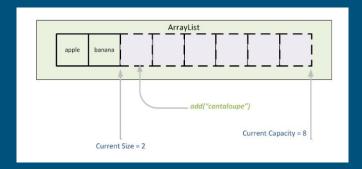
ArrayList<T> (pacote java.util) pode alterar <u>dinamicamente</u> seu tamanho para acomodar mais elementos.

- Tindica o tipo de elemento armazenado na coleção
- Isso é semelhante a especificar o tipo ao declarar um array, exceto que apenas tipos <u>não-primitivos</u> podem ser utilizados com essas classes de coleção.

Classes com essa espécie de marcador de lugar são chamadas *classes genéricas*.

ArrayList

Coleção em Java que permite armazenar elementos de forma DINÂMICA.



emerson@paduan.pro.br

ArrayList

Exemplo:

```
import java.util.ArrayList;
public class Exemplo {
    public static void main(String args[])
    {
        ArrayList<String> nomes = new ArrayList<>>();
        nomes.add("Huguinho");
        nomes.add("Zezinho");
        nomes.add("Luizinho");
        System.out.println(nomes);
    }
}
```

Principais métodos

		
Método	Descrição	
add(Object o)	Adiciona um elemento ao fim do ArrayList	
add(int index, Object o)	Adiciona um elemento no índice especificado do ArrayList	
clear()	Remove todos os elementos do ArrayList	
get(int index)	Retorna o elemento do índice especificado	
indexOf(Object o)	Retorna o índice da primeira ocorrência do elemento especificado no ArrayList	
remove(Object o)	Remove a primeira ocorrência do valor especificado	
remove(int index)	Remove o elemento do índice especificado	
size()	Retorna o número de elementos armazenados no ArrayList	
isEmpty()	Retorna true se não existem elementos no ArrayList	

emerson@paduan.pro.b

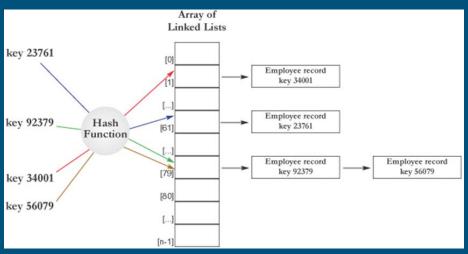
foreach

```
foreach – (para cada)
iterar sobre coleções de maneira simples e direta
Sintaxe:

for( tipo variavel : nomeArray ){

//corpo do for
```

Hash



emerson@paduan.pro.b

Em Java

```
public static void main(String[] args) {
    HashMap<Integer, String> mapa = new HashMap<>();

mapa.put(1, "um");
    mapa.put(2, "dois");
    mapa.put(3, "três");
    mapa.put(4, "quatro");

System.out.println(" 3 = " + mapa.get(3));
    System.out.println(" 3 = " + mapa.get(32));
}
```