



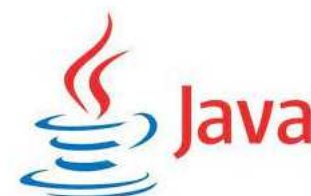
Residência em Software

Residência em Tecnologia da Informação e Comunicação

Estrutura Básica de um programa Java

Professor:

Alvaro Degas Coelho



INSTITUIÇÃO EXECUTORA



COORDENADORA



APOIO



Packages

- Classes podem (e devem) ser organizadas em pacotes
 - packages
- Classes que convivem num mesmo pacotes são “amigas”
 - protected
- Idealmente: cada classe num arquivo com extensão .java
 - E todas dentro de um package
- Podem haver packages dentro de packages
 - Crie um novo projeto chamado RedeSocial
 - Seu projeto tem um package: src
 - Dentro do src crie um package cliente
- Note que packages vão em letras minúsculas e classes em letras maiúsculas
 - convenção

Import de Biblioteca

- Crie uma classe Usuario com os atributos nome, email e nacionalidade
 - Getters e Setters
 - Dica: procure “Insert Getters and Setters” no menu Source – Botão direito)
- Digite cuidadosamente a linha abaixo para criar um novo atributo chamado listaDeEmails
 - Dica: vai usando ctrl + space
- `private ArrayList<String> postagens;`
- Note que o Eclipse está apontando a necessidade de importar bibliotecas
 - ArrayList
 - Aceite a sugestão dele (import class, ou organize imports)
- Similar ao `#include`

Declaração da Classe

- Um exemplo

```
public class Biometrica{  
    public int altura,peso,idade;  
        public Biometrica(int altura,int peso,int idade){  
            this.altura=altura;  
            this.peso=peso;  
            this.idade=idade;  
        }  
}
```

Declaração da Classe

- Ex: `public class media { //Declaração da classe Media.`

Declaração do objeto

- Declaração do Objeto nada mais e do que declarar as variáveis que apontarão para as instâncias .Na declaração os objetos ainda não estão instanciados,mas apenas nomeados.
- Biometrica entrada, saida;

Instanciação do Objeto

- Objetos são estruturas de dados definidas e agrupadas dentro de uma classe. Sempre que utilizamos um objeto ou classe devemos reservar espaço em memória para que aquele objeto seja manipulado sem maiores problemas.
- Além do mais, também podemos utilizar a mesma classe (com todos os seus métodos e atributos) para manipular outros objetos que serão tratados de forma diferente (mesmo se tiverem as mesmas características do objeto anterior), pois serão dois endereços de memória diferentes.

Instanciação do Objeto

- Sem contar que, se um objeto não é mais referenciado dentro do programa, o próprio Java trata de liberar os recursos de memória consumidos pelo objeto usando o Garbage Collector - Coletor de Lixo.
- Quem faz o papel de instanciador em Java é o **new**. New trata de reservar memória o suficiente para o objeto e criar automaticamente uma referência a ele. Para new conseguir determinar o objeto, precisamos usar o [método construtor](#) que será usado como base para instanciar a classe e gerar o objeto.

Instanciação do Objeto

- Tecnicamente, declaramos uma variável qualquer como sendo do tipo da classe (ex.: `TV minhaTV;`), depois instanciamos o objeto atribuindo a variável o resultado obtido por `new` mais o método construtor (ex.: `minhaTV = new TV();`).
- Apesar de parecer muitos detalhes, isso fica mais facilmente visualizado no código abaixo.

Exemplo 1

```
public class TV {  
    int tamanho;  
    int canal;  
    boolean ligada;  
  
    TV() {  
        tamanho = 21;  
        canal = 0;  
        ligada = false;  
    }  
  
    public static void main(String[] args) {  
        TV objeto1 = new TV();  
        TV objeto2;  
        objeto2 = new TV();  
    }  
}
```

No código ao lado, criamos uma classe chamada TV

Declaramos duas variáveis chamadas objeto1 e objeto2 sendo do tipo TV. Depois, instanciamos o objeto usando new e o método construtor.

Exemplo 2

Ex: Class Biometrica

```
Entrada=new Biometrica ();  
Saida=new Biometrica ();
```

Criação de Classe

- Já vimos, mas precisamos documentar
- `class NomeDaClasse{ ...}`
- Crie a classe `Usuario` com os atributos, os getters e os setters.
- Crie um construtor (com parâmetros ou não, você decide) e ponha o seguinte comando
 - `Postagens = new ArrayList<String>()`
 - Isto força a alocação de espaço para armazenar as postagens

Classe main

- Crie seu método main
- Dentro dela crie uma variável do tipo usuário
- Solicite que sejam digitados dados deste usuário
- Crie um método adicionaPostagem(String post) {...} na classe
- Crie uma variável String chamada postagem
- Solicite que sejam digitados algumas postagens e execute o método adicionaPostagem(passando a variável postagem como parâmetro)

NÃO SE PREOCUPE! ARRAYLIST SERÁ VISTO MAIS DETALHADAMENTE
NAS PRÓXIMAS AULAS!!!

Operadores Matemáticos

- Os mesmos de C++
 - +, -, *, /, %
- Crie um atributo quantidadeDePostagens
 - Inicia com 0 no construtor
 - Incrementa em 1 quando executa adicionaPostagem()
- Crie um atributo pontuacao (inteiro)
 - Inicia com zero no contador
- Crie um método alteraPontuacao(int delta) {...}
 - A pontuação é somada com o delta (que pode ser negativo)
- Getters e Setters
 - Teste seus métodos

Estruturas de controle

- if (condição) {... (caminho da verdade)}
- else {... Caminho da mentira}
- Modifique seu método alteraPontuacao(int Delta)
 - A pontuação nunca pode ficar menor que zero.

Repetição

- `Do {...} while (condicao)`
- `While (condicao) {...}`
- Limpe sua classe main
- Crie um ArrayList de Usuario
 - `ArrayList<Usuario> listaUsuarios = new ArrayList<Usuario>;`
- Crie a seguinte estrutura
 - `do {`
 - Solicita os dados de um Usuário
 - Cria uma variável Usuário (chame de usuario – minúsculo)
 - `listaUsuarios.add(usuario);`
 - Solicita se quer inserir mais usuários (uma variável String chamada sn)
 - `while (sn != "n");`

Comando For

- `for (int v=Inicial; v< final; v++) {...}`
- O arraylist tem um atributo `size()` que diz seu tamanho
- E tem um método `get(i)` que retorna o item da posição `i`
- Crie um método `mostrarPostagens()`
 - Este método deve usar um `for` para percorrer o arraylist de postagens, partindo da posição 0 até seu tamanho
 - Recuperar cada postagem e escrever na tela
- **IMPORTANTE:** isto pode ser feito com iterators, mas será visto nas próximas aulas

Exercícios

- 1. Calculadora de Média:
 - Crie uma classe ListaDeNumeros e um ArrayList de float dentro dela
 - Peça ao usuário que insira uma série de números. Quando o usuário inserir um número negativo, pare de pedir números e calcule a média dos números positivos inseridos.
 - Crie o método para mostrar a média

Exercícios

- 2. Tabuada:
 - Crie uma classe Tabuada
 - Peça ao usuário que insira um número e imprima a tabuada desse número de 1 a 10.
 - Crie um método que receba um número como parâmetro e escreva a tabuada dele na tela