INSTITUTO FEDERAL
Sudeste de Minas Gerais

Construtores

Curso: Técnico em Informática Integrado

Prof.: Jean Henrique de Sousa Câmara

Contato: jean.camara@ifsudestemg.edu.br

Inicialização de objetos com construtores

- Como visto em aulas anteriores, um atributo de um objeto **é inicializado com o valor null**
- Caso queira fornecer o valor de um atributo no **momento da criação do objeto** deve-se usar os construtores
- Um construtor **deve ter o mesmo nome que a classe**

Inicialização de objetos com construtores

- Como exemplo vamos especificar um nome para um objeto Conta quando ele é criado

```
//Cria um objeto do tipo Conta
Conta minhaConta = new Conta(nomePessoa: "Jean Henrique");
```

- Para que isso funcione, precisamos especificar o construtor na classe Conta

```
//O construtor inicializa nome com nomePessoa
public Conta(String nomePessoa){
    nome = nomePessoa;
}
```

Inicialização de objetos com construtores

- Instanciando dois objetos do tipo Conta

ContaTeste.java > ...

```java
public class ContaTeste{
    Run | Debug
    public static void main(String[] args) {
        //Cria dois objetos do tipo Conta
        Conta conta1 = new Conta(nomePessoa: "Jean Henrique");
        Conta conta2 = new Conta(nomePessoa: "Isaac Luiz");

        //Exibe o nome das contas
        System.out.printf(format: "O nome da conta 1 é %s%n", conta1.getNome());
        System.out.printf(format: "O nome da conta 2 é %s%n", conta2.getNome());
    }
}
```

Inicialização de objetos com construtores

- Uma diferença importante entre construtores e métodos é que os **construtores não podem retornar valores**
- Em qualquer classe que não declare explicitamente um construtor, o compilador fornece um tipo padrão (que sempre não tem parâmetros)
  - Chamado de **construtor padrão**
- Se você declarar um construtor para uma classe, o compilador não criará um construtor padrão para ela

Inicialização de objetos com construtores

- Quando uma classe tem somente o construtor padrão, as variáveis de instância da classe são inicializadas de acordo com seus valores padrões

Modificando a classe Conta

- Baixe os arquivos Conta.java e ContaTeste.java do SIGAA e implemente o seguinte construtor na classe Conta

```java
//Atributos
private String nome;
private float saldo;

//O construtor inicializa nome com nomePessoa e saldo com saldoInicial
public Conta(String nomePessoa, float saldoInicial){
    this.nome = nomePessoa;//armazena o nome

    if (saldoInicial > 0) {
        this.saldo = saldoInicial;//armazena o saldo
    }
}
```

Modificando a classe Conta

- Crie um método para exibir o nome e saldo de uma conta (exibeInformacoes)
- Depois chame esse método na classe ContaTeste no lugar desse código

```
//Exibe as informações da conta
System.out.printf(format: "Saldo da conta do %s: %.2f %n", conta1.getNome(), conta1.getSaldo());
System.out.printf(format: "Saldo da conta do %s: %.2f %n%n", conta2.getNome(), conta2.getSaldo());
```

Diagrama de classe UML

- Define a estrutura das classes do sistema
  - Apresenta uma visão estática de como as classes estão organizadas
- Estabelece como as classes se relacionam
- Uma classe é representada por um retângulo com três divisões:
  - Nome da Classe
  - Atributos da Classe
  - Métodos da Classe

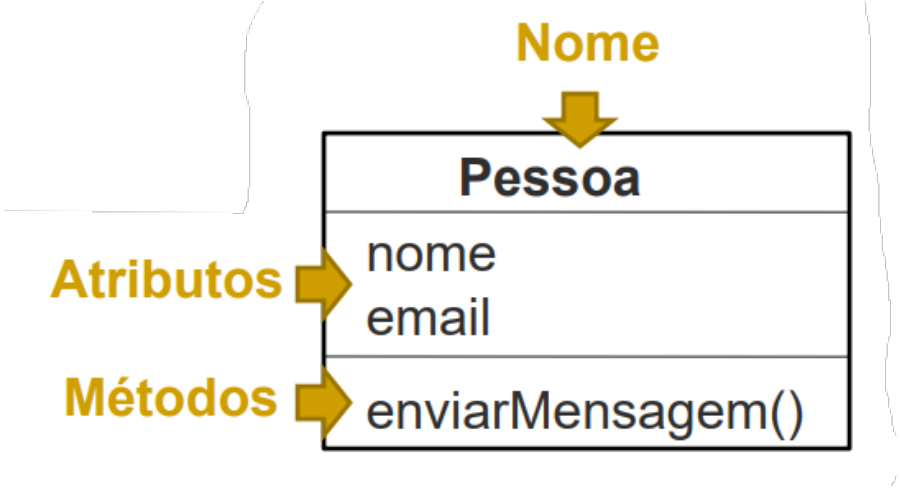


Diagrama de classe UML - Visibilidade

- **Pública (+) - public**
  - O atributo ou método pode ser utilizado por qualquer classe
- **Protegida (#) - protected**
  - Somente a classe, sub-classes e classes amigas (ex. pacote) têm acesso
- **Privada (-) - private**
  - Somente a própria classe terá acesso

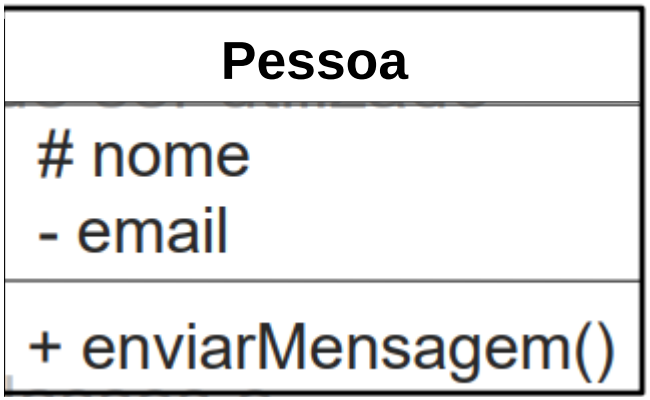
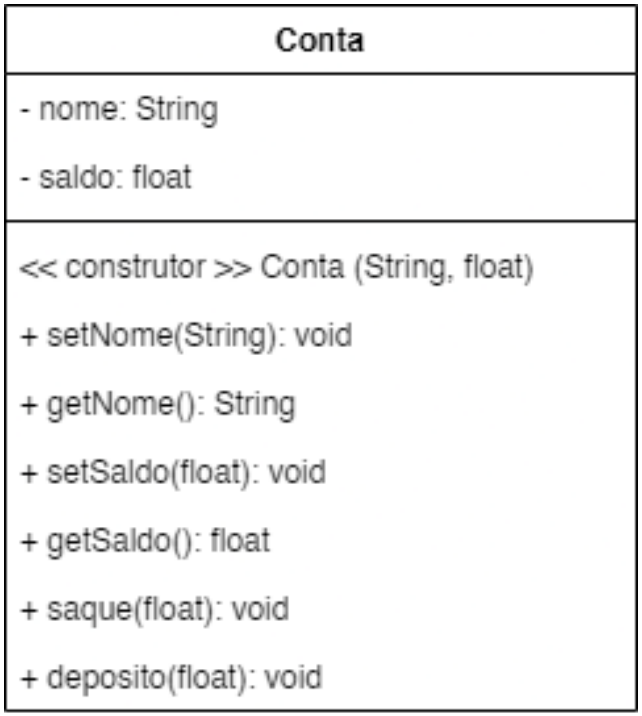


Diagrama de classe UML

- Editor online UML: https://app.diagrams.net/
- Veja como fica o diagrama UML para nossa classe Conta



Utilizando caixas de diálogo

- Os programas apresentados até agora exibem a saída na janela de comando
- Muitos aplicativos utilizam **janelas ou caixas de diálogo** para exibir a saída
- A classe **JOptionPane** fornece caixas de diálogo pré-constituídas que permitem aos programas exibir janelas que contêm mensagens

Utilizando caixas de diálogo

- Hora de praticar!

CaixaMensagem.java > ...

```java
// Usando JOptionPane para exibir múltiplas linhas em uma caixa de diálogo.
import javax.swing.JOptionPane;

public class CaixaMensagem {
    Run | Debug
    public static void main(String[] args) {
        // exibe um diálogo com uma mensagem
        JOptionPane.showMessageDialog(parentComponent: null, message: "Bem vindo ao Java");
    }
}
```

Utilizando caixas de diálogo

- A linha 2 indica que o programa utiliza a classe **JOptionPane** do pacote **javax.swing**
  - Esse pacote contém muitas classes que ajudam a criar interface gráfica (GUIs) para aplicativos
- A linha 7 chama o método **JOptionPane** **showMessageDialog** para exibir uma caixa de diálogo que contém uma mensagem

Utilizando caixas de diálogo

- Esse método requer dois argumentos:
  - O primeiro ajuda o aplicativo Java a determinar onde posicionar a caixa de diálogo
    - Se for null, a caixa de diálogo será exibida no centro da tela
  - O segundo argumento é a String a ser exibida na caixa de diálogo

Utilizando caixas de diálogo

- O método JOptionPane showMessageDialog é o chamado **método static**
- Métodos estáticos são usados para definir tarefas que sempre possuem o mesmo código
- Um método static é chamado utilizando seu nome de classe seguido por um ponto (.) e o nome de método

Utilizando caixas de diálogo

- Solicitando dados para o usuário em uma caixa de diálogo

CaixaMensagem.java > ...

```java
// Usando JOptionPane para exibir múltiplas linhas em uma caixa de diálogo.
import javax.swing.JOptionPane;

public class CaixaMensagem {
    Run | Debug
    public static void main(String[] args) {
        // pede para o usuário inserir seu nome
        String nome = JOptionPane.showInputDialog(message: "Qual é o seu nome?");

        // cria a mensagem
        String mensagem = String.format(format: "Bem vindo, %s, à programação Java!", nome);

        // exibe a mensagem para cumprimentar o usuário pelo nome
        JOptionPane.showMessageDialog(parentComponent: null, mensagem);
    }
}
```

Utilizando caixas de diálogo

- A linha 7 usa o método **showInputDialog** de JOptionPane para exibir uma caixa de diálogo de entrada que contém um prompt e um campo (conhecido como campo de texto) no qual o usuário pode inserir o texto
  - Retorna uma String contendo os caracteres digitados pelo usuário

Utilizando caixas de diálogo

- A linha 10 utiliza o método **static String format** para retornar uma String que contém uma saudação com o nome do usuário
  - O método format funciona como System.out.printf, exceto que format retorna a String formatada em vez de exibi-la em uma janela de comando

Utilizando caixas de diálogo

- Hora de praticar!
  - Modifique o arquivo ContaTeste.java para que toda entrada e saída de dados sejam baseadas em caixas de diálogo com os métodos da classe JOptionPane
    - Uma vez que o método showInputDialog retorna uma String, você deve converter a String que o usuário insere em um float para utilização em cálculos
      - O método static parseFloat da classe Float (pacote java.lang) recebe um argumento String que representa um número e retorna o valor como um float
      - Se a String não contiver um número válido, o programa terminará com um erro

Exercício

- Crie uma classe chamada Fatura que possa ser utilizado por uma loja de suprimentos de informática para representar uma fatura de um item vendido na loja
- Uma fatura deve incluir as seguintes informações como atributos:
  - o número do item faturado
  - a descrição do item
  - a quantidade comprada do item
  - o preço unitário do item

Exercício

- Sua classe deve ter um construtor que inicialize os quatro atributos
  - Se a quantidade não for positiva, ela deve ser configurada como 0
  - Se o preço por item não for positivo ele deve ser configurado como 0.0.
- Forneça um método set e um método get para cada variável de instância
- Além disso, forneça um método chamado getTotalFatura que calcula o valor da fatura (isso é, multiplica a quantidade pelo preço por item) e
- depois retorna o valor como um double
- Escreva um aplicativo de teste que demonstra as capacidades da classe Fatura

Trabalho

- Individual
- Compactar os arquivos e entregar pelo SIGAA
- Valor: 3.0 pontos
- Data de entrega: até dia 14/08

Trabalho

- Uma questão relacionada à assistência médica discutida ultimamente nos veículos de comunicação é a computadorização dos registros de saúde
- A computadorização dos registros de saúde pode facilitar que pacientes compartilhem seus perfis e históricos de saúde entre vários profissionais de saúde
- Isso talvez aprimore a qualidade da assistência médica, ajude a evitar conflitos e prescrições erradas de medicamentos, reduza custos em ambulatórios e salve vidas

Trabalho

- Neste exercício, você **projetará uma classe**

  Paciente para uma pessoa

  - Os atributos da classe devem incluir
    - Nome
    - Sexo
    - Data de nascimento (consistindo em atributos separados para mês, dia e ano de nascimento)
    - Altura (em metros)
    - Peso (em quilogramas)

Trabalho

- Sua classe deve ter um construtor que receba esses dados.
- Para cada atributo, forneça métodos set e get
- A classe também deve incluir métodos que calculem e retornem
  - a idade do usuário em anos
  - o intervalo de frequência cardíaca máxima
  - a frequência cardíaca alvo
  - o índice de massa corporal (IMC)

Trabalho

- Segundo a American Heart Association (AHA), a fórmula para calcular a frequência cardíaca máxima por minuto é 220 menos a idade em anos
  - Sua frequência cardíaca alvo é um intervalo entre 50-85% da sua frequência cardíaca máxima

$$\text{IMC} = \frac{\text{Peso}}{\text{Altura} \times \text{Altura}}$$

IMC	Classificações
Menor do que 18,5	Abaixo do peso normal
18,5 - 24,9	Peso normal
25,0 - 29,9	Excesso de peso
30,0 - 34,9	Obesidade classe I
35,0 - 39,9	Obesidade classe II
Maior ou igual a 40,0	Obesidade classe III

Classificação segundo a OMS a partir do IMC

Trabalho

- Escreva também um aplicativo Java que solicite as informações da pessoa, instancie um objeto da classe **Paciente** para ela e imprima as informações a partir desse objeto — incluindo nome, sexo, data de nascimento, altura e peso da pessoa —, e então calcule e imprima o IMC, intervalo de frequência cardíaca máxima e frequência cardíaca alvo.
- Ele também deve exibir o gráfico de valores IMC.

Dúvidas?

jean.camara@ifsudestemg.edu.br