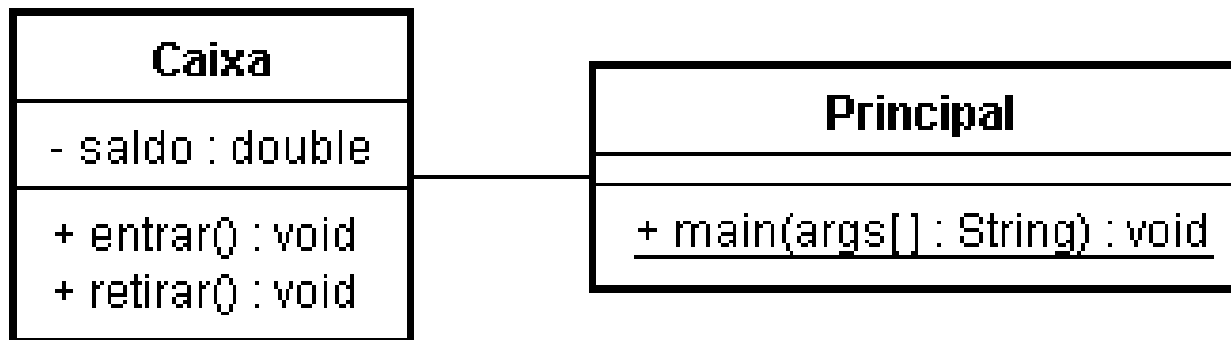


# Análise e desenvolvimento de aplicações orientadas a objeto com Java SE

Aula 4 - Métodos com Classe de Modelagem

# Projeto exemplo: ControleCaixa



# Estrutura básica de um projeto Java

- Projeto (Estrutura de pastas e arquivos que compõem a aplicação)
  - Classe de modelagem (Definição de tipo)
    - Atributo
    - Construtores
    - Getters e setters
    - Métodos específicos da classe
  - Classe Principal
    - Método main()

# Código da estrutura básica da classe Caixa

```
Caixa.java X
public class Caixa {

    // Atributo
    private double saldo;

    // Construtores
    // Inicializando o atributo zerado
    public Caixa() {
        this(0);
    }
    // Inicializando o atributo com o parâmetro
    public Caixa(double saldo) {
        this.saldo = saldo;
    }

    // getter e setter
    public double getSaldo() {
        return saldo;
    }

    public void setSaldo(double saldo) {
        this.saldo = saldo;
    }
}
```

# Código dos métodos específicos da classe Caixa

```
// Métodos específicos da classe

public void entrar() {
    // Lê um valor, converte de String para double e atribui a variável valor
    double valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor da entrada: "));
    /* Soma o conteúdo do atributo saldo com o a variável valor e passa o resultado por parametro
       para o método setSaldo() */
    this.setSaldo(this.saldo + valor);
}

public void retirar() {
    // Lê um valor, converte de String para double e atribui a variável valor
    double valor = Double.parseDouble(JOptionPane.showInputDialog("Digite o valor da retirada: "));
    /* Subtrai o conteúdo do atributo saldo com o a variável valor e passa o resultado por parametro
       para o método setSaldo() */
    this.setSaldo(this.saldo - valor);
}
```

Lembrando de importar a classe swing para poder utilizar a JOptionPane



# Classe Principal


- Instanciar um objeto chamado cx1 do tipo caixa.
- Apresentar um menu com as opções:
  - 1 – Entrada
  - 2 – Retirada
  - 3 – Consultar saldo
  - 0 – Sair
- Realizar as chamadas aos métodos de acordo com a opção do usuário
- Repetir as operações enquanto não for informada a opção zero

# Código da classe Principal

```
Caixa.java Principal.java X
import javax.swing.*; // importação da API swing
public class Principal {
    public static void main(String args[]){
        Caixa cx1 = new Caixa(); // Instanciação do objeto cx1
        int op; // declaração da variável de opções
        do{ // Início do looping do-while
            // Apresentação e leitura do menu de opções
            op = Integer.parseInt(JOptionPane.showInputDialog("Digite: \n1 - Entrada " +
                "\n2 - Retirada \n3 - Consultar saldo \n0 - Sair "));
            switch (op) { // Abertura da estrutura de switch-case
                case 1:
                    cx1.entrar(); // Chamada ao método entrar do objeto cx1
                    break;
                case 2:
                    cx1.retirar(); // Chamada ao método retirar do objeto cx1
                    break;
                case 3:
                    // Apresentação do conteúdo do atributo saldo
                    JOptionPane.showMessageDialog(null, "Saldo atual: " + cx1.getSaldo());
                    break;
                case 0:
                    JOptionPane.showMessageDialog(null, "Finalizando programa!");
                    break;
                default:
                    JOptionPane.showMessageDialog(null, "Opção inválida!");
            }
        }while(op != 0); // Repetirá as operações enquanto a opção for diferente de zero
    }
}
```

# Em tempo de execução

Input


 Digite:

- 1 - Entrada
- 2 - Retirada
- 3 - Consultar saldo
- 0 - Sair

1

OK Cancel


Input

 Digite o valor da entrada:

100

OK Cancel

Input


 Digite:

- 1 - Entrada
- 2 - Retirada
- 3 - Consultar saldo
- 0 - Sair

3

OK Cancel

Input


 Digite:

- 1 - Entrada
- 2 - Retirada
- 3 - Consultar saldo
- 0 - Sair

2

OK Cancel


Input

 Digite o valor da retirada:

70


OK Cancel

Message

 Saldo atual: 100.0

OK

Message

 Saldo atual: 30.0

OK





# Assinatura de métodos

- A primeira linha do método é chamada de assinatura e ela indica:
  - Visibilidade
  - Retorno de valor
  - O nome do método
  - Passagem de parâmetros

# Assinatura de métodos

- Exemplo: Sem retorno e sem parâmetros.

- Na classe de modelagem:

```
public void cadastraProduto() {  
  
}
```

- No método main:

```
cadProd.cadastraProduto();
```

# Assinatura de métodos

- Exemplo: Sem retorno e com parâmetros.

- Na classe de modelagem:

```
public void cadastraProduto(int quantidade) {  
  
}
```

- No método main:

```
cadProd.cadastraProduto(qtde);
```

# Assinatura de métodos

- Exemplo: Com retorno e sem parâmetros.

- Na classe de modelagem:

```
public boolean cadastraProduto() {  
  
    return true;  
}
```

- No método main:

```
boolean resposta = cadProd.cadastraProduto();
```

# Assinatura de métodos

- Exemplo: Com retorno e com parâmetros.

- Na classe de modelagem:

```
public boolean cadastraProduto(int quantidade) {  
  
    return true;  
}
```

- No método main:

```
boolean resposta = cadProd.cadastraProduto(qtde);
```

# Assinatura de métodos

- Exemplo: Passando um objeto por parâmetro.

- Na classe de modelagem:

```
public void cadastraProduto(Produto p) {  
  
}
```

- No método main:

```
cadProd.cadastraProduto(prod) ;
```

# Exercicio 01

## ■ Projeto: CalculadoraBasica

