



TECNOLOGIA EM DESENVOLVIMENTO DE SOFTWARE MULTIPLATAFORMA

TÉCNICA DE
PROGRAMAÇÃO I



TÉCNICA DE PROGRAMAÇÃO I
PROFº LUIZ CLÁUDIO

Java – Operadores Aritméticos



Função	Sinal
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Resto da divisão	%
Incremento	++
Decremento	--

Java – Operadores Relacionais



Função	Sinal
Igual	==
Diferente	!=
Maior que	>
Maior ou igual a	>=
Menor que	<
Menor ou igual a	<=

Java – Operadores Lógicos



Função	Sinal
E	&&
OU	
Não	!

Java – Conversão de Tipos



Supondo a variável x	Converter em	y recebe o valor convertido
✓ Entre tipos numéricos		
int x = 10	float	float y = (float) x
int x = 10	double	double y = (double) x
float x = 10.5	int	int y = (int) x
✓ De string para numéricos		
String x = "10"	int	int y = Integer.parseInt(x)
String x = "20.5"	float	float y = Float.parseFloat(x)
String x = "20.5"	double	double y = Double.parseDouble(x)
✓ De numéricos para string		
int x = 10	String	String y = Integer.toString(x) ou String y = String.valueOf(x)
float x = 10.5	String	String y = Float.toString(x) ou String y = String.valueOf(x)
double x = 10.5	String	String y = Double.toString(x) ou String y = String.valueOf(x)

Java – Inserção de Comentários



```
// Comentários em uma única linha
```

```
/* Comentários em  
 * várias linhas  
 */
```

Java – ESTRUTURAS CONDICIONAIS IF / CASE



```
if (num1>=10) {  
    System.out.println("Condição verdadeira!");  
}else{  
    System.out.println("Condição falsa!");  
}
```

```
switch (op) {  
case 1:  
    System.out.println("Caso op igual a 1...");  
    break;  
case 2:  
    System.out.println("Caso op igual a 2...");  
    break;  
case 3:  
    System.out.println("Caso op igual a 3...");  
    break;  
default:  
    System.out.println("Caso op não seja 1, 2 ou 3");  
    break;  
}
```

Java – ESTRUTURAS DE REPETIÇÃO WHILE / DO WHILE / FOR



```
// Teste condicional no inicio  
while(op != 0){  
    // Instruções  
}
```

```
// Teste condicional no fim  
do{  
    // Instruções  
}while(op != 0);
```

```
// Numero pré-definido de interações  
for(int i=0; i<=10; i++){  
    // Instruções  
}
```


Java – Entrada e Saída de dados



■ Classe JOptionPane e a diretiva import.

```
import javax.swing.JOptionPane;
// ou import javax.swing.*;

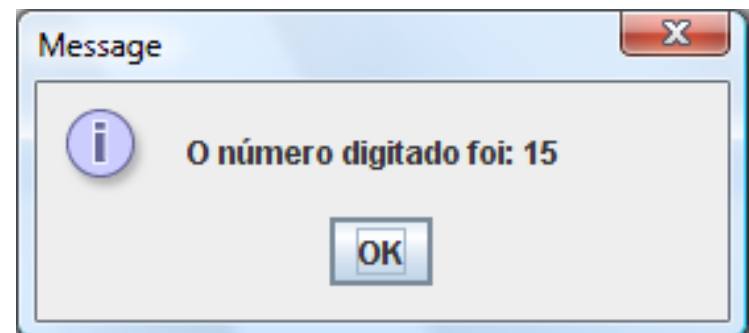
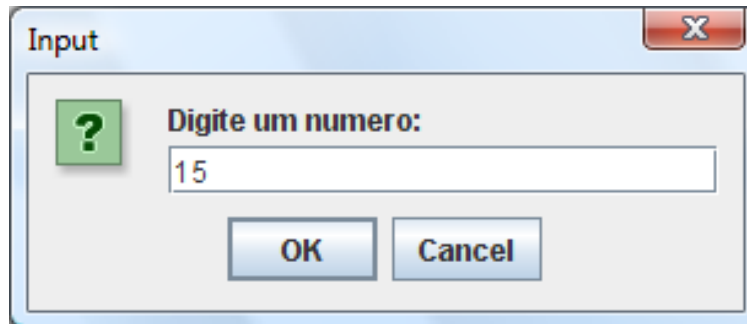
public class TesteEntradaSaida {

    public static void main(String[] args) {

        String numero = JOptionPane.showInputDialog("Digite um numero:");

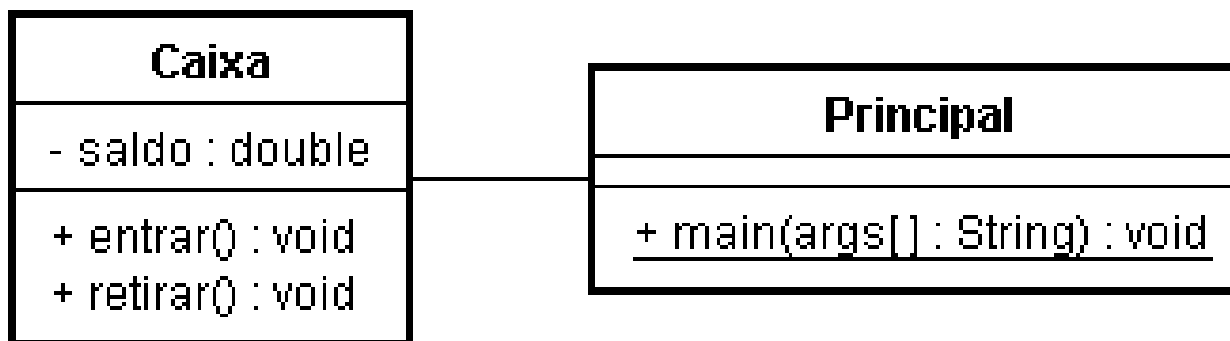
        JOptionPane.showMessageDialog(null, "O número digitado foi: " + numero);

    }
}
```



Java – Projeto Exemplo

Controle Caixa



Java – Projeto Exemplo

Controle Caixa



- **Workspace** (espaço em disco onde serão salvos os arquivos dos projetos)
 - **Projeto** (Estrutura de pastas e arquivos que compõem a aplicação)
 - **Classe de modelagem (Definição de tipo)**
 - Atributo
 - Contrutores
 - Getters e setters
 - Métodos específicos da classe
 - **Classe Principal**
 - Método main()

Java – Projeto Exemplo

Controle Caixa



Código da
estrutura
básica da
classe Caixa

Caixa.java

```
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;  
    }  
}
```

Java – Projeto Exemplo

Controle Caixa



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

Java – Projeto Exemplo

Controle Caixa



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

Projeto Controle Caixa- 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
    }
}
```

Java – Projeto Exemplo

Controle Caixa



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

i

Saldo atual: 100.0

OK

Message

i

Saldo atual: 30.0

OK

Java – 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

Java – Assinatura de métodos



- Exemplo: Sem retorno e sem parâmetros.

- Na classe de modelagem:

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

- No método main:

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

Java – Assinatura de métodos



- Exemplo: Sem retorno e com parâmetros.

- Na classe de modelagem:

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

- No método main:

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

Java – 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();
```

Java – 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);
```