



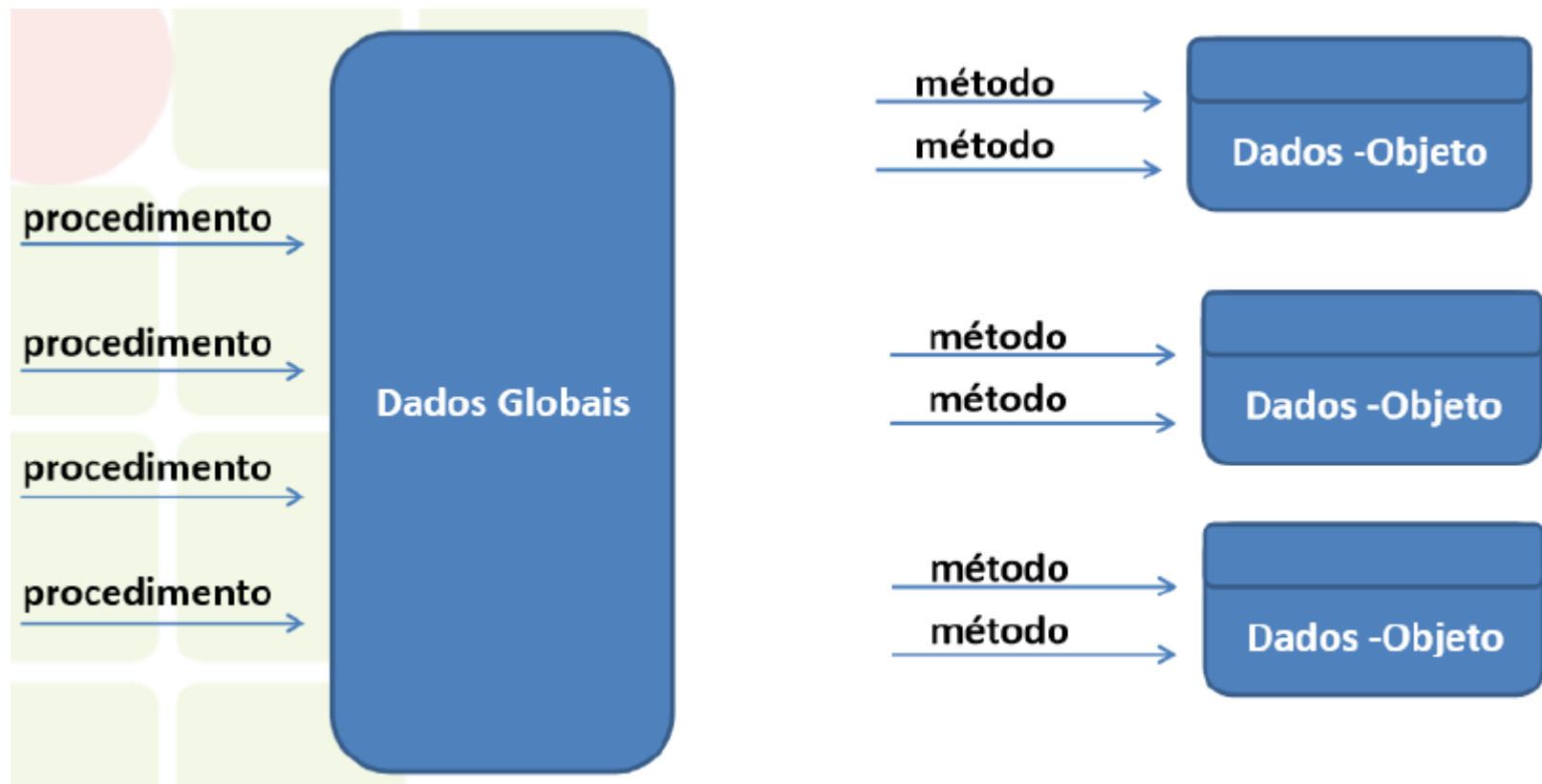
Instituto Federal de Santa Catarina – IFSC
Campus São José

Programação Orientada a Objetos

Prof. Francisco de Assis S. Santos, Dr.

São José, 2015.

Programação Estruturada X Orientada a Objetos



Programação Estruturada X Orientada a Objetos

Programação Estrutura:

- Consiste na criação de um conjunto de procedimentos (algoritmos) para resolver o problema;
- Encontrar modos apropriados de armazenar os dados.

Programação Orientada a Objetos:

- Consistem em identificar os objetos e as operações relevantes no mundo real;
- O mapeamento desses em representações abstratas no espaço de soluções.

Introdução ao Java: Definições

- Linguagem desenvolvida pela SUN para manter o poder computacional do C++ e promover **segurança, robustez e portabilidade**;
- Totalmente Orientada a Objetos;
- Não possui ponteiros (Utiliza um gerenciador para alocação de memória - Garbage Collection);
- Possibilita a programação multitarefa, mesmo em sistemas operacionais que não oferecem suporte nativo a Threads;



Introdução ao Java: Arquitetura

- A máquina virtual Java (JVM – *Java Virtual Machine*);
- Gerenciador de alocação/liberação de memória (*Garbage Collection*);
- *Sand box* – módulo de garantia de segurança de código (é impossível criar um vírus em Java);



Introdução ao Java: Ferramentas de Desenvolvimento

- O kit de desenvolvimento Java (*Java Development Kit* - JDK) e distribuído gratuitamente pela Oracle
- Compilador, máquina virtual Java, código fonte, documentação das APIs
- Para executar uma aplicação Java é necessário possuir uma máquina virtual Java
- A Oracle disponibiliza gratuitamente o ambiente de execução Java (*Java Runtime Environment* - JRE) para diversos sistemas operacionais e arquiteturas de máquina

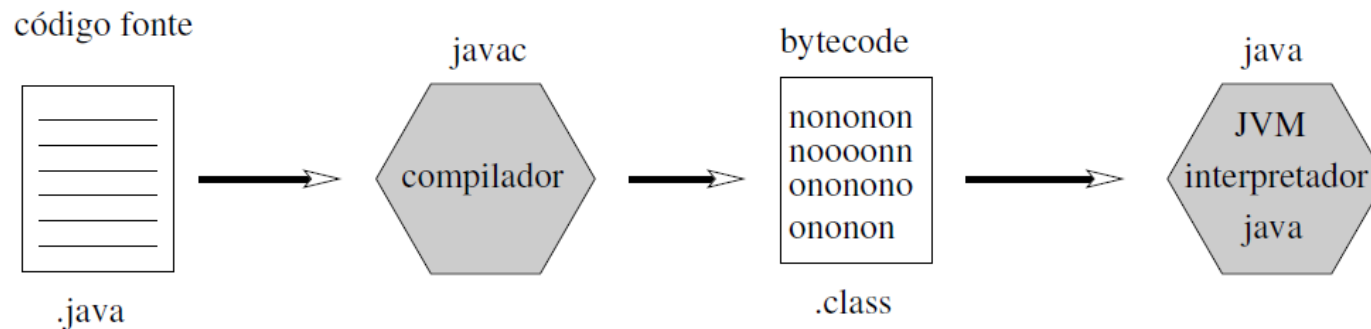
Introdução ao Java: Algumas tecnologias Java

- Java SE - *Standard Edition*
 - Para o desenvolvimento de aplicações desktop
- Java EE - *Enterprise Edition*
 - Para o desenvolvimento de aplicações empresariais
- Java ME - *Mobile Edition*
 - Para o desenvolvimento de aplicações para dispositivos moveis
 - Java FX
 - Para o desenvolvimento de ricos aplicativos clientes (áudio, vídeo);
 - Aplicações JavaFX podem ser para navegador, desktop e dispositivos moveis.

Introdução ao Java: Ambientes Integrados de Desenvolvimento

- Os Ambientes Integrados de Desenvolvimento (AID) tornam mais agradável e produtivo o desenvolvimento de aplicações
 - Netbeans - Apoiado pela Oracle
 - Eclipse – Bastante utilizado pela comunidade

Introdução ao Java: Criando e executando um aplicativo em Java



- Compilando:
javac Arquivo.java
- Executando:
java Arquivo

Introdução ao Java: Estrutura básica de código

1. // Duas barras significam comentário
2. /* comentários também podem seguir o formato de C++ */
- 3.
4. public class *NomeDoPrograma*
5. {
6. // O método ***main sempre deve estar presente para que um código***
7. // Java possa ser executado:
8. static public void main(String[] *args*)
9. {
10. // aqui virão os comandos, que são semelhantes ao C++
11. }
12. }

Introdução ao Java: Comando básicos

1. Declaração de variáveis:

<tipo> + <espaço> + identificador + ; ou

<tipo> + <espaço> + identificador + = + valor + ;

1. Mensagem na tela:

```
System.out.println ("Alô mundo");
```

1. Condicional:

```
if (expressão_booleana)
```

```
{
```

```
// bloco de comandos
```

```
}
```

```
else
```

```
{ // bloco alternativo de comandos }
```

Introdução ao Java: Exemplo

```
package primeiroprograma;
public class PrimeiroPrograma {
public static void main(String[] args) {
    // TODO code application logic here
    int x=5;
    int y=10;
    if (x<y)
    {
        System.out.print("Valor da variável x é menor que o valor da variável y");
    }
    else
    {
        System.out.print("Valor da variável x é maior ou igual que o valor da variável y");
    }
    }
}
```

Introdução ao Java: Comando básicos

4. Laços de repetição:

```
for (int i = 0; i < 10; i++)  
{  
    // bloco de comandos  
}
```

```
While (expressão_booleana)  
{  
    // bloco de comandos executados enquanto a  
    // expressão booleana tiver valor verdade = true  
}
```

Introdução ao Java: Comando básicos

4. Laços de repetição:

```
do  
{  
// bloco de comandos executados pelo menos uma vez  
} While (expressão_booleana);
```

1.Array

```
int[] vetor = new int [10];  
OU
```

```
int[][] matriz = new int [10][10];
```

6. Leitura do teclado:

```
import java.util.Scanner; // antes da classe principal  
Scanner sc = new Scanner (System.in); // dentro da void main  
String Termo = sc.next(); // dentro da void main
```

Introdução ao Java: Alguns tipos de dados

```
System.out.println("Tipos de dados numéricos em Java: \n" +  
    "\nMenor Byte: " + Byte.MIN_VALUE +  
    "\nMaior Byte: " + Byte.MAX_VALUE +  
    "\nMenor Short Int: " + Short.MIN_VALUE +  
    "\nMaior Short Int: " + Short.MAX_VALUE +  
    "\nMenor Int: " + Integer.MIN_VALUE +  
    "\nMaior Int: " + Integer.MAX_VALUE +  
    "\nMenor Long: " + Long.MIN_VALUE +  
    "\nMaior Long:" + Long.MAX_VALUE +  
    "\nMenor Float: " + Float.MIN_VALUE +  
    "\nMaior Float: " + Float.MAX_VALUE +  
    "\nMenor Double: " + Double.MIN_VALUE +  
    "\nMaior Double: " + Double.MAX_VALUE);  
}
```

Introdução ao Java: Funções Matemáticas

```
double d;  
// algumas funções matemáticas  
d = Math.sqrt(25); // obtêm a raiz quadrada  
d = Math.pow(4,2); // 4 elevado a 2  
d = Math.sin(45); // Math.cos(45), Math.tan(45)...  
int valor = (int) (Math.random() * 100); // valor aleatório  
entre 0 e 100
```


Introdução ao Java: Formata Número

```
import java.text.DecimalFormat;
```

```
double x=3.36985465456;
```

```
float y= 3.3696546f;
```

```
DecimalFormat formata = new DecimalFormat("0.##");
```

```
System.out.print(" Número formatado:
```

```
" +formata.format(x));
```

Métodos: Sem retorno

- Pode, ou não, receber parâmetros.
- Não retorna um resultado.
- Exemplos:
 - Gravar dados em arquivo no disco;
 - Enviar documento para a impressora;
 - Gerar Série Numérica;
 - Ordenar que o computador desligue.

Método sem retorno: Exemplo 01

```
public static void Imprime ( )  
{  
  
    System.out.println("Teste de Impressão");  
    System.out.println("IFSC – Instituto Federal de Santa Catarina");  
  
}
```

Método sem retorno: Exemplo 02

```
public static void GeraFibonacci (int num)
{
    String SerieFibonacci="0";
    int anterior=0;
    int atual=1;
    int i=0;
    while (i<(num-1))
    { SerieFibonacci = SerieFibonacci+" "+String.valueOf(atual);
      atual = atual+anterior;
      anterior = atual-anterior;
      i++;
    } JOptionPane.showMessageDialog(null, SerieFibonacci,"Série
      Fibonacci com "+num+" termos:",DEFAULT_OPTION);
}
```

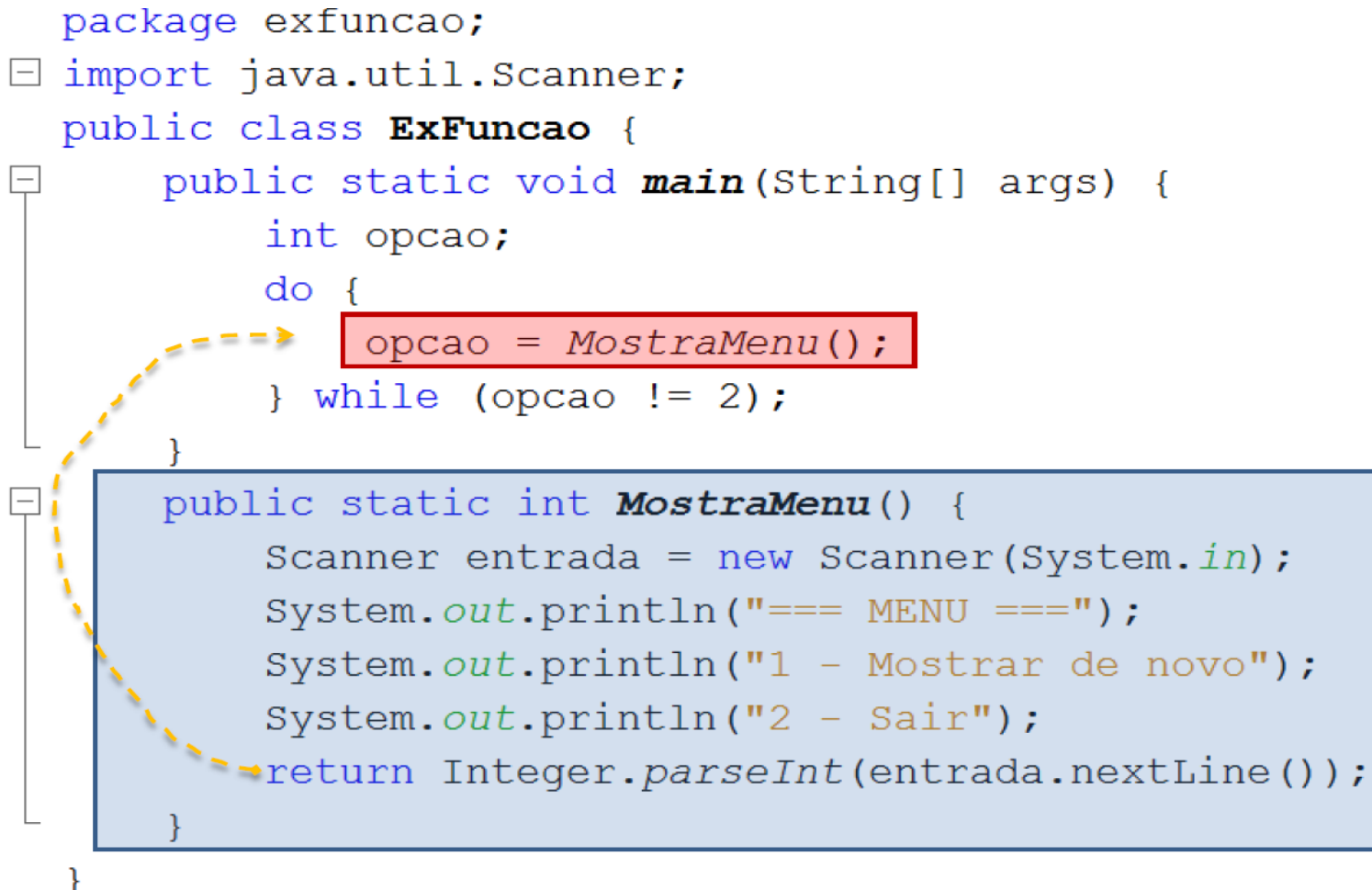
Métodos: Com Retorno

Algo que deve ser feito uma ou várias vezes, sempre que for necessário para se obter um resultado.

- Pode, ou não, receber parâmetros;
- Sempre retorna um resultado;
- Exemplos:
 - Mostrar um menu e retornar a opção;
 - Solicitar uma entrada ao usuário;
 - Efetuar uma operação matemática;
 - Verificar uma condição lógica;

Método com retorno: Exemplo 01

```
package exfuncao;
import java.util.Scanner;
public class ExFuncao {
    public static void main(String[] args) {
        int opcao;
        do {
            opcao = MostraMenu();
        } while (opcao != 2);
    }
    public static int MostraMenu() {
        Scanner entrada = new Scanner(System.in);
        System.out.println("=== MENU ===");
        System.out.println("1 - Mostrar de novo");
        System.out.println("2 - Sair");
        return Integer.parseInt(entrada.nextLine());
    }
}
```



Método com retorno: Exemplo 02

Calcular descontos percentual em valores de produtos.

– Precisa de quais dados?

- Precisa do valor atual do produto –**double**;
- Precisa do percentual de desconto –**double**;

– Resulta em algum novo dado?

- Sim, é uma função! O valor com desconto –**double**;

```
public static double calcDesc(double va, double pc)
```

```
{
```

```
double vd= va* (pc/ 100);
```

```
return va-vd;
```

```
}
```

Método com retorno: Exemplo 03

```
public static boolean isosceles (int num1, int num2, int num3)
{
    return ((num1==num2)||num2==num3)||num1==num3);
}
```


Exercício

Desenvolver uma rotina que permita receber todos os proventos de funcionários (salário base, horas extras e adicionais), considere a contribuição à previdência social (INSS) e calcule o IRRF conforme tributação vigente (incluindo dependentes). Utilizando métodos em Java deve-se obter o salário líquido dos funcionários.

Como calcular o IRRF:

Some todos os vencimentos (salário base, gratificações, horas extras). Subtraia a contribuição à previdência. Subtraia a dedução por dependente, se houver. Aplique a alíquota sobre esse resultado e subtraia a parcela a deduzir correspondente (conforme tabela vigente).

Bibliografia

- Notas de aula do Prof. Emerson Ribeiro de Mello (IFSC)
- Notas de aula do Prof. Vilson Heck Junior (IFSC)
- Notas de aula do Prof. Jailton Carlos (IFRN)