SENAI CIMATEC

SISTEMA FIEB

TECNOLOGIA E INOVAÇÃO EM PROL DA INDÚSTRIA

SENAI CIMATEC

SISTEMA FIEB

Desenvolvimento de sistemas I - 72 h

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Tipos, Operadores, Estruturas e I/O

Prof: Diego Corrêa

Estrutura de um programa em JAVA

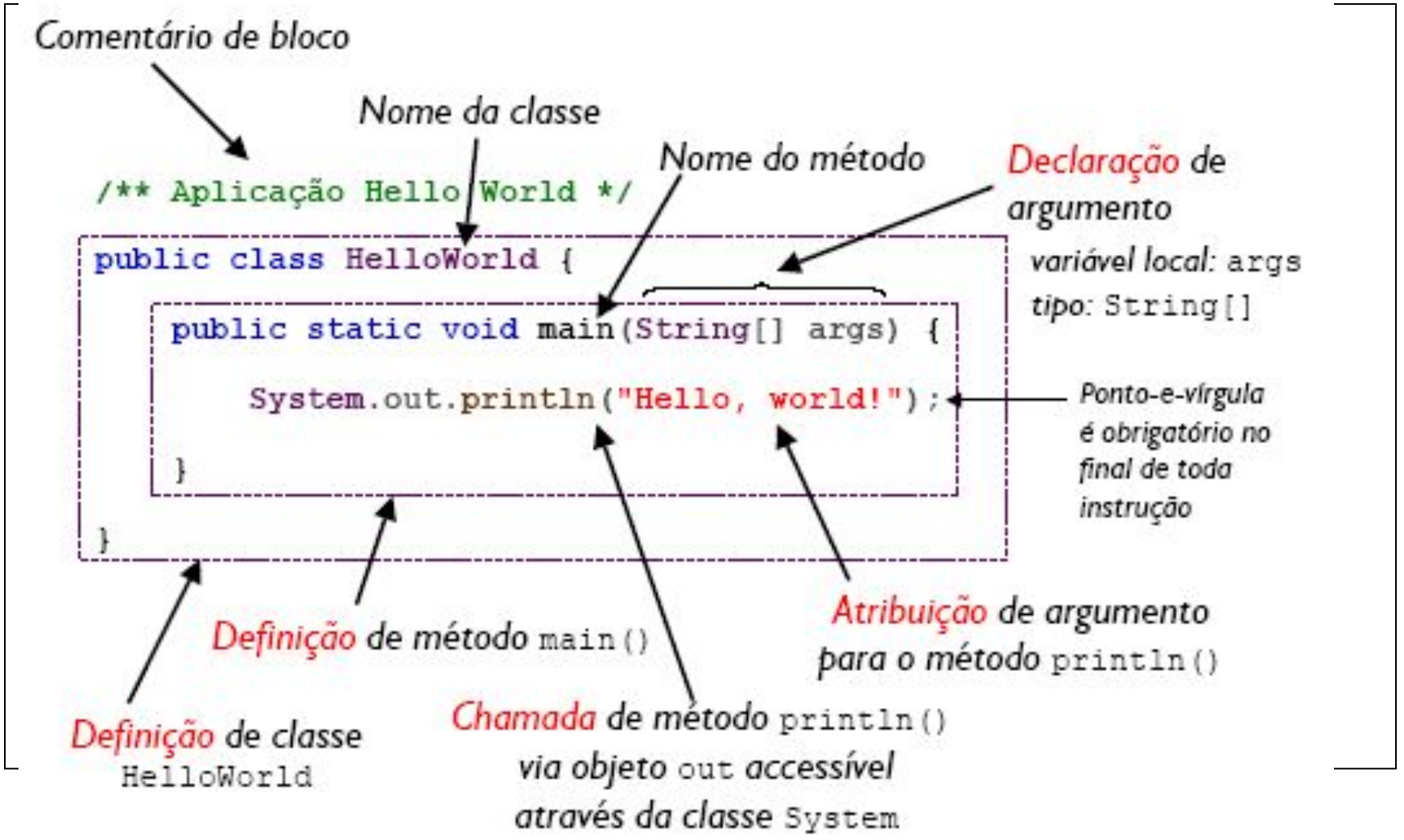
- Todo programa Java, independente do tipo, é uma classe
- Classes são organizadas em arquivos texto com a extensão .java
- Cada arquivo pode conter **apenas uma** classe **pública**

Estrutura de um programa em JAVA

- O nome do arquivo da classe Java deve possuir exatamente o mesmo nome da classe

- Quando uma classe precisar utilizar outras classes, estas devem ser importadas através do comando **import**

Estrutura de um programa java



Aplicações em Java

Uma "application" em Java nada mais é que uma **classe** Java possuindo um **método específico** que o caracteriza como uma aplicação para a **Máquina Virtual Java**

Aplicações em Java

Toda classe Java que possuir o método com a seguinte **assinatura**

public static void main (String[] ***argumentos***){ }

Será um application

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Comentários

//	Usado para indicar comentários em uma única linha;
/* */	Usados para indicar comentários que se estendem por uma ou mais linhas.

Palavras Reservadas

abstract	double	int	strictfp
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	package	throw
char	for	private	throws
class	goto	protected	transient
const	if	public	try
continue	implements	return	void
default	import	short	volatile
do	instanceof	static	while

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Olá Mundo!

A Maldição

Reza a lenda que que o programador que não fizer no seu primeiro código com as instruções que escreva na tela o simples **Hello World** ou **Olá Mundo**... Não conseguirá entender ou mesmo não será um bom profissional na área da programação….

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Olá Mundo

```java
public class HelloWorld {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }

}
```

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Entrada e Saída

Input e Output

- O Java possui uma biblioteca padrão para Saída de dados: **System.out**

- Para imprimir dados na tela, basta chamar a **System.out.println(“Texto a ser impresso.”);**

Input e Output

- Uma das formas de ler os dados é utilizar a classe **Scanner()**

- Para esta classe necessita ser importa via: **import java.util.Scanner**

- A utilização segue:

```
Scanner input = new Scanner(System.in);
String inputString = input.nextLine();
```

Input e Output

Construir um programa que requisite o nome do usuário e em seguida imprima na tela.

Input e Output

```java
package entrada_saida;
import java.util.Scanner;

/**
 *
 * @author lowe
 */
public class Entrada_Saida {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Digite o nome: ");
        Scanner input = new Scanner(System.in);
        String inputString = input.nextLine();
        System.out.println(inputString);
    }

}
```

Input e Output

- O Java possui uma classe chamada **JOptionPane** que facilita as seguintes tarefas:
  - exibição de mensagens de alerta, confirmação ou informativas
  - captura de informações digitadas pelo usuário

Input e Output

- Para exibir alguma informação/alerta usamos o seguinte método:
  - JOptionPane.showMessageDialog(null,...)
- Para pedir uma confirmação ao usuário usamos:
  - JOptionPane.showConfirmDialog(null,...)

Input e Output

- Para capturar alguma informação digitada pelo usuário declaramos inicialmente uma variável qualquer do tipo String
- String x;
- Atribuímos posteriormente o retorno do método à variável:
- **x**= JOptionPane.showInputDialog(...)

Input e Output

Construa um programa que avisa esta sendo iniciado e requisite o primeiro nome em uma tela e o sobrenome em outra tela. Depois peça confirmação para o nome completo.

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Input e Output

```
package entrada_saida_jpane;
import javax.swing.JOptionPane;
/**
 *
 * @author lowe
 */
public class Entrada_Saida_JPane {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, "Estamos iniciando o programa!");
        String primeiro_nome = JOptionPane.showInputDialog("Digite o Primeiro nome: ");
        String sobrenome = JOptionPane.showInputDialog("Digite o Sobrenome: ");
        JOptionPane.showConfirmDialog(null,"Olá, " + primeiro_nome + " " + sobrenome + "!");

    }

}
```

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Tipos Primitivos no Java

Tipos de dados

- Java é uma linguagem de programação orientada a objetos e, com **exceção** dos tipos primitivos, qualquer coisa em Java é uma classe/objeto

- Além disso, Java é uma linguagem **fortemente *tipada***

Tipos de dados

- Possui 8 (oito) tipos primitivos.
- 4 tipos de **inteiros**
- 2 tipos de **ponto flutuante**
- 1 tipo **caracter**(UNICODE)
- 1 tipo **lógico**

Tipos de dados

- Tipos Numéricos Inteiros

Tipo	Tamanho(bits)	Faixa
byte	8	-128 até +127
short	16	-32,768 até +32,767
int	32	-2,147,483,648 até +2,147,483,647
long	64	-9,223,372,036,854,775,808 até +9,223,372,036,854,775,807

- Tipos Numéricos Reais

Tipo	Tamanho(bits)	Faixa
float	32	-3.40292347E+38 até +3.40292347E+38
double	64	-1.79769313486231570E+308 até +1.79769313486231570E+308

Tipos de dados

- Tipo caracter

Tipo	Tamanho em bits	Faixa
char	16	UNICODE - 65536 caracteres possíveis

- Um caracter é delimitado por **apóstrofos**, ex: ‘c’.
- As Strings são delimitadas por **aspas duplas**, ex: “String”.
- A linguagem Java **não possui o tipo primitivo string**, bastante conhecido em várias outras linguagens. Para a manipulação de texto são utilizadas as **classes String** e **String Buffer**

Tipos de dados

Tipo Lógico

Tipo	Faixa
boolean	`true` ou `false`

Variáveis

- Declaração:
  - **Tipo** nomeDaVariável [= *ValorInicial*];
- Exemplo: **String** nome;
  ou **String** nome = “SENAI”;

- **Observação**: Os identificadores Java são **case-sensitive**, ou seja, java diferencia as minúsculas das maiúsculas. Portanto, a variável **nome** é diferente da variável **Nome**

Variáveis

- Declaração:
  - **Final tipo** NOME_DA_CONSTANTE = *ValorConstante*;
- Exemplo: **final String** NOME = “SENAI”;

Operadores

Aritméticos

Operador	Uso	Descrição
+	op1+op2	Adiciona op1 e op2
-	op1-op2	Subtrai op2 de op1
*	op1*op2	Multiplica op1 por op2
/	op1/op2	Divide op1 por op2
%	op1%op2	Calcula o resto da divisão de op1 por op2

Operadores

Incremento/Decremento

Operador	Uso	Descrição
++	var++	Usa a variável, depois incrementa
	++var	Incrementa a variável, depois usa
--	var--	Usa a variável, depois decrementa
	--var	Decrementa a variável, depois usa

Operadores

Relacionais

Operador	Descrição
==	Equivalência de valor
!=	Diferente
> <	“Maior que” e “menor que” respectivamente
>= <=	“Maior ou igual a” e “menor ou igual a” respectivamente

Operadores

Atribuição

Operador	Uso	Descrição
=	Op1=valor	Atribui valor a Op1
+=	Op1+=Op2	Soma op2 a op1 e guarda o resultado em op1
-=	Op1-=Op2	Subtrai op2 de op1 e guarda o resultado em op1
*=	Op1*=Op2	Multiplica op1 por op2 e guarda o resultado em op1
/=	Op1/=Op2	Divide op1 a op2 e guarda o resultado em op1

Operadores

Lógicos

Operador	Descrição
!	Not (Nega qualquer valor/expressão booleana que venha após este símbolo)
&&	And
||	Or

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Estruturas de Seleção

Seleção

- if – else
  - permite analisar uma expressão lógica e direcionar a execução do programa
  - Caso a expressão lógica seja verdadeira (true) a sequência do **if** será executada
  - Caso a expressão lógica seja falsa (false), e exista a cláusula else, a sequência do else será executada

Seleção

- **Estrutura:**

```
if( expressão_lógica ){
    Sequência;
}
if( expressão_lógica ){
    Sequência_1; // Caso a expressão lógica seja verdadeira
}else {
    Sequência_2; // Caso a expressão lógica seja falsa
}
```

Seleção

```java
package pkgif;

import javax.swing.JOptionPane;

/**
 *
 * @author lowe
 */
public class If {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int resposta = JOptionPane.showConfirmDialog(null,"Você é maior de idade?");
        if (resposta == JOptionPane.YES_OPTION){
            JOptionPane.showMessageDialog(null, "É maior de idade!");
        } else {
            JOptionPane.showMessageDialog(null, "Ainda é menor!");
        }
    }

}
```

Seleção

- **Switch**
  - Compara (igualdade) uma variável (**byte, char, short ou int**) a uma relação de valores, permitindo associar comandos a cada valor da relação

Seleção

- **Estrutura:**

```
switch(variável) {
    case valor1 : sequência1;
    case valor2 : sequência2;
    case valor3 : sequência3;
    ...
    case valorN : sequênciaN;
}
```

Seleção

```
public class Switch_case {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int numero = Integer.parseInt(JOptionPane.showInputDialog("Digite um número de 1 a 4"));
        switch(numero) {
            case 1:
                JOptionPane.showMessageDialog(null, "O primeiro número!");
            break;
            case 2:
                JOptionPane.showMessageDialog(null, "O segundo número!");
            break;
            case 3:
                JOptionPane.showMessageDialog(null, "O terceiro número!");
            break;
            case 4:
                JOptionPane.showMessageDialog(null, "O quarto número!");
            break;
            default:
                JOptionPane.showMessageDialog(null, "Número não permitido");

        }
    }
```

SENAI CIMATEC SISTEMA FIEB
Federação das Indústrias do Estado da Bahia

Estrutura de Repetição

Repetição

- Em java, como em todas linguagens de programação, há estruturas de controle que permitem repetir instruções, são elas:
  - **while**
  - **for**
  - **do..while**

Repetição

- **while**
  - Executa a seqüência do comando enquanto a expressão lógica for verdadeira;
  - A expressão lógica é analisada antes de cada execução e caso não seja verdadeiro o programa nem entra no *laço*;

```
while ( expressão_lógica ){
    Sequência;
}
```

Repetição

```
public class While {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int contador = 10;
        while(contador > 0) {
            contador--;
            System.out.println(contador);
        }
    }

}
```

Repetição

- do..while
  - Executa a sequência do comando enquanto a expressão lógica for verdadeira.
  - Após a execução da sequência a expressão lógica é analisada.

Repetição

- **Obs:**A sequência sempre será executada **pelo menos uma vez** com a instrução do..while.

```
do
    Sequência;
while ( expressão_lógica );
```

Repetição

```
public class Do_While {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int contador = 10;
        do {
            System.out.println(contador);
            contador--;
        }while(contador > 0);
    }

}
```

Repetição

- for
  - Executa a sequência do comando enquanto a expressão lógica for verdadeira
  - Sendo que permite inicializar a variável, na entrada do comando e incrementar variável a cada repetição

```
for (expressão_inicializacao;
   expressão_lógica; incremento){
    Sequência;
}
```

Repetição

```java
public class For {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        for(int contador = 0; contador < 10; contador++){
            System.out.println(contador);
        }
    }

}
```

Break..Continue

- **break**
  - Força a saída de um comando de repetição ou do comando switch
- **continue**
  - Força o início da próxima iteração de um comando de repetição

Exercício

- 1 -Elaborar um programa em Java que:
  - –Solicite que o usuário **digite um número**.
  - – Calcule o **fatorial** deste número. (Multiplicação dos números que vão de 1 até o próprio número). Ex:
    - 4! = 1x2x3x4 = 24
    - 5! = 1x2x3x4x5 = 120
    - 6! = 1x2x3x4x5x6 = 720
    - 7! = 1x2x3x4x5x6x7 = 5040
    - Exiba o resultado na tela.

Exercício

2 - Elaborar um programa em Java que:

- Solicite que o usuário **digite 2 números**.
- Decida qual deles é o maior e exiba na tela o resultado da seguinte maneira:
  - “Maior = [número] / Menor = [número]”
- Caso os dois sejam iguais, deverá exibir como resultado:
  - “Os números são iguais!”.

Exercício

3 -Elaborar um programa em Java que leia 2 valores numéricos e conte quantos números ímpares múltiplos de 3 existem entre esses dois números (inclusive), ex:

- O usuário digitou 3 e 22, os números ímpares múltiplos de 3 entre 3 e 22 são: 3,9,15 e 21, portanto o programa responderá que existem 4 números que satisfazem a condição do enunciado.

Exercício

- 4 - Elaborar um programa em Java que leia um número inteiro e exiba uma mensagem identificando se ele é um número primo ou não.
- Obs: um número primo é um número que a divisão só é exata por ele mesmo e por 1.