

Ambientação com a Sintaxe de Java: parte 2

Prof. Gustavo Wagner
Slides Originais: Prof. Tiago Massoni

Desenvolvimento de Sistemas

FATEC-PB

Escopo de blocos

- Um bloco em Java é um conjunto de comandos delimitados por { }.
- Cada bloco define o escopo para suas variáveis.

```
{  
    int n;  
    int k;  
    k = 0;  
}
```

```
{  
    int n;  
    {  
        int k;  
    }  
    k = 0;  
}
```

```
{  
    int n;  
    {  
        int k;  
        int n;  
    }  
}
```

Comando condicional (if-else)

```
if (condicao) {  
    comando1;  
}
```

```
if (condicao)  
{  
    comando1;  
} else {  
    comando2;  
}
```

condicao deve ser obrigatoriamente uma expressão booleana

```
if (media > 7) {  
    resultado = "Aprovado";  
} else {  
    resultado = "Reprovado";  
}
```

Comandos de laço (while)

```
while (condicao) {  
    bloco  
}
```

**Teste é feito
no início**

```
int contador = 0;  
while (contador < 10) {  
    System.out.println(contador);  
    contador++;  
}
```

```
while (true) {  
    fazerAlgumaCoisa;  
}
```

Laço infinito

Comandos de laço (do-while)


```
do {  
    bloco  
}while (condicao);
```

**Teste é feito
no final**



```
int contador = 0;  
do {  
    System.out.println(contador);  
    contador++;  
}while (contador < 10);
```

**Laço é
executado
pelo menos
uma vez**



Comando de laço (for)

```
int soma = 0;
int valor = 10;
for (int i = 0; i < valor ; i = i+1) {
    soma = soma + i;
}
System.out.println(soma);
```

É executado
quando o `for`
termina

Se `valor`
for menor
ou igual a
`0`, não
executa-
se o
corpo do
`for`

O comando de escolha (switch)

```
switch (expressao)
{
    case rotulo1:
        Comandos1
        break;
    case rotulo2:
        Comandos2
        break;
    ...
    default:
        Comandos
}
```

Para executar um `switch`

- Avalia-se `expressao`
- Executa-se os comandos do `case` cujo rótulo é igual ao valor resultante da expressão
- Executa-se os comandos de `default` caso o valor resultante não seja igual a nenhum rótulo

Usando o comando switch

```
switch (resposta) {  
    case 's': case 'S':  
        retorno = true;  
        break;  
    case 'n': case 'N':  
        retorno = false;  
        break;  
    default:  
        retorno = false;  
  
    console.println("Erro!");  
}
```

**Tipo integral
apenas:
int,char
constantes**

**Pula para o
final do
comando**

**Se nenhum
caso for
verdade...**

Comandos de desvio

- break: muda fluxo para depois do laço completo
- continue: muda fluxo para próxima execução do laço

Comandos de desvio

Imprime apenas os pares

```
int i = 0;
while (true) {
    i++;
    if (i % 2 != 0)
        continue;
    System.out.println(i);
    if (i == 10)
        break;
}
```

Volta para o INÍCIO do laço, para novo teste

Pula para o próximo comando DEPOIS do laço

Funções (subprogramas)

- Além da função main, podemos definir outras funções dentro de uma classe
 - Como funções em C
- Funções podem receber zero ou mais parâmetros
- Podem retornar valor ou não

Funções

**Obrigatório
para funções**

- Declaração

```
public static tipo nomeFuncao (parms)  
{..}
```

- Função que não retorna: `void`

- Exemplos

```
public static void main (String[] args){..}  
public static int retornaMaior (int a, int b){..}  
public static int retornaCodASCII(char c){..}
```

Funções

- Se funções retornam algum valor, corpo deve conter pelo menos um *return*
- Os parâmetros são passados por cópia para o corpo da função

```
public static int retornaMaior(int a,  
    int b) {  
    int resultado=0;  
    ... (compara quem é o maior)  
    return resultado;  
}
```

Funções

- Funções são chamadas a partir do nome da classe

```
int a = Programa.retornaMaior (2, 3);  
int c =  
    Programa.retornaCodigoASCII ( 'a' );
```

- Função main não pode ser chamada por você (Máquina virtual “chama” esta função)

Comentários

- Comentário em linha simples

```
// Exemplo de comentário em linha
```

- Comentário em bloco (/*..*/)

```
/*
```

```
Exemplo de
```

```
comentário em bloco
```

```
*/
```

Ler dados do teclado

- A leitura de dados do teclado será feita utilizando funções da classe Scanner.
 - **Scanner sc = new Scanner(System.in);**
 - **int i = sc.nextInt();**
 - **double d = sc.nextDouble();**
 - **String s = sc.next();**
- Para ler dados de um arquivo, basta trocar a chamada a new Scanner por **new Scanner(new File("nome_arquivo"))**.

Exercícios em Sala: sintaxe de Java

Exercício em sala 1

- Criar uma classe programa chamada Multiplicacao e crie a função main.
- Realizar uma multiplicação entre 2 e 3, imprimindo o resultado na tela
- Utilizar variáveis x,y para multiplicar (inicializá-las com 2,3)¹⁸

Exercício em sala 1

1. Multiplicar x e y usando apenas o operador de adição (+) – comando de laço
3. Transformar o cálculo da multiplicação em uma função multi (recebendo dois parâmetros e retornando o resultado). A função main apenas faz a chamada

Exercício em sala 1

1. Receber os dois números como leitura do usuário (Usar classe Scanner)
3. Proibir a entrada de números negativos do usuário (evitar que a entrada seja um número negativo)

Exercício em sala 2

- Crie uma classe-programa chamada GeraPrimos e crie o método main.
- Na função main implemente um programa que lê um número inteiro do teclado e imprime todos os números primos menores que ele (divisíveis por 1 ou por ele mesmo).

Exercício em sala 3

- Crie uma classe-programa chamada Fatorial e crie a função main.
- Na função main implemente um programa que lê um número do teclado e em seguida imprime seu fatorial. (procure implementar isso utilizando alguma estrutura de laço)
- Pode usar uma função auxiliar