

Módulo 2

A linguagem Java

Programação Orientada a Objetos I

Java

(Rone Ilídio)

Operadores

- Atribuição: =
- Aritméticos:
 - Soma: +
 - Subtração: -
 - Multiplicação: *
 - Divisão: /
 - Modulo: %

Precedência

- $*$, $/$ e $\%$: mesma precedência, a maior
- $+$ e $-$: mesma precedência
- $=$: menor de todas
- Em caso de empate resolve-se da esquerda para a direita
- O uso de parênteses altera a precedência

Operadores

- Igualdade:
 - igual a: $=$
 - diferente: $\neq$
- Relacionais
- maior que: $>$
- menor que: $<$
- maior ou igual: $\geq$
- menor ou igual: $\leq$

Tipos de dados primitivos

Tipo	Bit	Exemplo
boolean	8	true, false
char	16	'a', 'A', '1', '@'
byte	8	-128 a 127
int	32	-32768 a 32767
long	64	-2^{63} a $+2^{63}$
float	32	-3.14, 0
double	64	-3.14, 0
void	0	nenhum valor nenhum valor

Estruturas de controle

- Comando if – sintaxe

```
if (condição) {  
    . . . seqüência de comandos . . .  
}
```

```
if (condição)  
    <comando> ;
```

Estruturas de controle

- Comando if/else – sintaxe

```
if (condição) {  
    . . . seqüência de comandos . . .  
}  
else {  
    . . . seqüência de comandos . . .  
}
```

```
import javax.swing.JOptionPane;

public class testeif{

    public static void main(String args[]){

        String entrada = JOptionPane.showInputDialog("Entre com um
                                                    número");

        int num = Integer.parseInt(entrada);

        if (num<=10){

            JOptionPane.showMessageDialog(null,“Menor que 10 ");

        }

        else if (num<100){

            JOptionPane.showMessageDialog(null,“Menor que 100");

        }

        else JOptionPane.showMessageDialog(null,“Maior que
                                                    100");

        System.exit(0);

    }

};
```

Repetição controlada

- Comando while – sintaxe

```
while (condição){  
    . . . seqüência de comandos . . .  
}
```

Repetição controlada

```
import javax.swing.JOptionPane;
public class testewhile{
    public static void main(String args[]){
        String entrada = JOptionPane.showInputDialog("Entre
                                                    com um número");
        int num = Integer.parseInt(entrada);
        while (num<=10){
            System.out.println("num = " + num);
            num ++;
        }
    }
};
```

Estrutura de repetição “for”

- Sintaxe

```
for (variável ← inicial;condição;incremento){  
    . . . seqüência de comandos . . .  
}
```

Estrutura de repetição “for”

```
public class testefor{  
    public static void main(String args[]){  
  
        for(int u=1; u<=10; u++ ){  
            System.out.println("u = " + u);  
        }  
        System.exit(0);  
    }  
}
```

Repetição controlada do/while

- Tem a mesma função do while.
- A diferença é que o bloco de comandos é executado pelo menos uma vez
- Sintaxe

do{

... seqüência de comandos ...

} while (condição)

Repetição controlada do/while

```
import javax.swing.JOptionPane;
public class testedowhile{
    public static void main(String args[]){
        String entrada = JOptionPane.showInputDialog("Entre com um
                                                    número");
        int num = Integer.parseInt(entrada);
        do{
            JOptionPane.showMessageDialog(null,"O número é" + num);
            num = num/2;
        }while (num>10);
        System.exit(0);
    }
};
```

Estrutura de Controle Switch

- Utilizado quando uma variável pode assumir vários valores
- Este comando só pode ser utilizado com variáveis dos tipos
 - byte
 - short
 - int
 - char

Estrutura de Controle Switch

- Sintaxe

```
switch (variável)
```

```
{
```

```
    case [valor]:
```

```
        . . . seqüência de comandos . . .
```

```
        break;
```

```
    case [valor]:
```

```
        . . . seqüência de comandos . . .
```

```
        break;
```

```
    default:
```

```
        . . . seqüência de comandos . . .
```

```
}
```

```
import javax.swing.JOptionPane;

public class testecase{
    public static void main(String args[]){
        String entrada = JOptionPane.showInputDialog("Entre com um número");
        int num = Integer.parseInt(entrada);
        switch (num)
        {
            case 1:
                JOptionPane.showMessageDialog(null,"1");
                break;
            case 2:
                JOptionPane.showMessageDialog(null,"2");
                break;
            default:
                JOptionPane.showMessageDialog(null,“maior ou igual a 3");
        }
        System.exit(0);
    }
};
```

Operadores Lógicos

- Operador “OU” (or)

A	B	A && B
false	false	false
false	true	false
true	false	false
true	true	true

- Operador “E” (and)

A	B	A B
false	false	false
false	true	true
true	false	true
true	true	true

Operadores Lógicos

- Operador “OU exclusivo” (xor)

A	B	$A \wedge B$
false	false	false
false	true	true
true	false	true
true	true	false

- Operador de negação (not)

A	$\neg A$
false	true
true	false