

# Aula 07 – Condicionais e Laços

Norton Trevisan Roman

5 de abril de 2013

# O operador ?

- O Java possui um atalho para condicionais:

# O operador ?

- O Java possui um atalho para condicionais:
  - ▶ O operador ?

# O operador ?

- O Java possui um atalho para condicionais:
  - ▶ O operador ?
    - ★ `var = condição ? expressão 1 : expressão 2;`

# O operador ?

- O Java possui um atalho para condicionais:

- ▶ O operador ?

- ★ `var = condição ? expressão 1 : expressão 2;`

- ★ Correspondendo a

- ```
if (condição) var = expressão 1;  
else var = expressão 2;
```

# O operador ?

- O Java possui um atalho para condicionais:

- ▶ O operador ?

- ★ `var = condição ? expressão 1 : expressão 2;`

- ★ Correspondendo a

- ```
if (condição) var = expressão 1;  
else var = expressão 2;
```

- ▶ Pode ser usado como substituição a esse tipo de condicional em qualquer parte do código:

# O operador ?

- O Java possui um atalho para condicionais:

- ▶ O operador ?

- ★ `var = condição ? expressão 1 : expressão 2;`

- ★ Correspondendo a

- `if (condição) var = expressão 1;`  
`else var = expressão 2;`

- ▶ Pode ser usado como substituição a esse tipo de condicional em qualquer parte do código:

```
static double areaPiscina(double raio) {  
    if (raio >= 0) return Math.PI *  
                        Math.pow(raio,2);  
    else return(-1);  
}
```

```
static double areaPiscina(double raio) {  
    double resp;  
    resp = (raio >= 0) ? Math.PI *  
                        Math.pow(raio,2) : -1;  
    return(resp);  
}
```

# O operador ?

- O Java possui um atalho para condicionais:

- ▶ O operador ?

- ★ `var = condição ? expressão 1 : expressão 2;`

- ★ Correspondendo a

- `if (condição) var = expressão 1;`  
`else var = expressão 2;`

- ▶ Pode ser usado como substituição a esse tipo de condicional em qualquer parte do código:

```
static double areaPiscina(double raio) {  
    if (raio >= 0) return Math.PI *  
                        Math.pow(raio,2);  
    else return(-1);  
}
```

```
static double areaPiscina(double raio) {  
    double resp;  
    resp = (raio >= 0) ? Math.PI *  
                        Math.pow(raio,2) : -1;  
    return(resp);  
}
```

Ou...



# O operador ?

- O Java possui um atalho para condicionais:

- ▶ O operador ?

- ★ `var = condição ? expressão 1 : expressão 2;`

- ★ Correspondendo a

- `if (condição) var = expressão 1;`  
`else var = expressão 2;`

- ▶ Pode ser usado como substituição a esse tipo de condicional em qualquer parte do código:

```
static double areaPiscina(double raio) {  
    if (raio >= 0) return Math.PI *  
                        Math.pow(raio,2);  
    else return(-1);  
}
```

Ou...

```
static double areaPiscina(double raio) {  
    double resp;  
    resp = (raio >= 0) ? Math.PI *  
                        Math.pow(raio,2) : -1;  
    return(resp);  
}  
  
static double areaPiscina(double raio) {  
    return((raio >= 0) ? Math.PI *  
            Math.pow(raio,2) : -1);  
}
```

# Switch

- Considere o código que calcula o valor da construção

# Switch

- Considere o código que calcula o valor da construção
- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina

# Switch

- Considere o código que calcula o valor da construção
- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina
  - ▶ Teremos que desmembrar o código, criando método apenas para a piscina

# Switch

- Considere o código que calcula o valor da construção
- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina
  - ▶ Teremos que desmembrar o código, criando método apenas para a piscina
  - ▶ Como fazer?

# Switch

- Considere o código que calcula o valor da construção

- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina

- ▶ Teremos que desmembrar o código, criando método apenas para a piscina
- ▶ Como fazer?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    if (material == ALVENARIA) valor = 1500;
    else
        if (material == VINIL) valor = 1100;
        else
            if (material == FIBRA) valor = 750;
            else
                if (material == PLASTICO)
                    valor = 500;
                else valor = -1;

    return(area*valor);
}
```

# Switch

- Considere o código que calcula o valor da construção
- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina
  - ▶ Teremos que desmembrar o código, criando método apenas para a piscina
  - ▶ Como fazer?
- Constantes também podem ser usadas para definir categorias

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    if (material == ALVENARIA) valor = 1500;
    else
        if (material == VINIL) valor = 1100;
        else
            if (material == FIBRA) valor = 750;
            else
                if (material == PLASTICO)
                    valor = 500;
                else valor = -1;

    return(area*valor);
}
```

# Switch

- Considere o código que calcula o valor da construção

- Suponha agora que haja 4 materiais, com preços diferentes, para a piscina

- ▶ Teremos que desmembrar o código, criando método apenas para a piscina
- ▶ Como fazer?

- Constantes também podem ser usadas para definir categorias
- ▶ Deixam o código bem mais legível

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    if (material == ALVENARIA) valor = 1500;
    else
        if (material == VINIL) valor = 1100;
        else
            if (material == FIBRA) valor = 750;
            else
                if (material == PLASTICO)
                    valor = 500;
                else valor = -1;

    return(area*valor);
}
```



# Switch

- Haveria como evitar todo esse aninhamento?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    if (material == ALVENARIA) valor = 1500;
    else
        if (material == VINIL) valor = 1100;
        else
            if (material == FIBRA) valor = 750;
            else
                if (material == PLASTICO)
                    valor = 500;
                else valor = -1;

    return(area*valor);
}
```

# Switch

- Haveria como evitar todo esse aninhamento?

- ▶ Switch:

```
switch (variavel) {  
    case valor_1: código_1  
        break;  
    case valor_2: código_2  
        break;  
    ...  
    case valor_n: código_n  
        break;  
    default: código_default;  
}
```

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    if (material == ALVENARIA) valor = 1500;  
    else  
        if (material == VINIL) valor = 1100;  
        else  
            if (material == FIBRA) valor = 750;  
            else  
                if (material == PLASTICO)  
                    valor = 500;  
                else valor = -1;  
  
    return(area*valor);  
}
```

# Switch

- Haveria como evitar todo esse aninhamento?

- ▶ Switch:

```
switch (variavel) {  
    case valor_1: código_1  
        break;  
    case valor_2: código_2  
        break;  
    ...  
    case valor_n: código_n  
        break;  
    default: código_default;  
}
```

- ▶ Correspondendo a

```
if (variavel == valor_1) código_1;  
else  
    if (variavel == valor_2) código_2;  
    else  
        ...  
        else  
            if (variavel == valor_n)  
                código_n;  
            else código_default;
```

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    if (material == ALVENARIA) valor = 1500;  
    else  
        if (material == VINIL) valor = 1100;  
        else  
            if (material == FIBRA) valor = 750;  
            else  
                if (material == PLASTICO)  
                    valor = 500;  
                else valor = -1;  
  
    return(area*valor);  
}
```

# Switch

## • Então...

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    if (material == ALVENARIA) valor = 1500;
    else
        if (material == VINIL) valor = 1100;
        else
            if (material == FIBRA) valor = 750;
            else
                if (material == PLASTICO)
                    valor = 500;
                else valor = -1;

    return(area*valor);
}
```

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- Switch testa a variável contra todos os valores listados

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- Switch testa a variável contra todos os valores listados
  - ▶ Se o valor dela não estiver listado, ativa o código em *default*

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- Switch testa a variável contra todos os valores listados
  - ▶ Se o valor dela não estiver listado, ativa o código em *default*
  - ▶ *default* é opcional – se não houver, simplesmente sai do *switch* e continua o programa

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- Switch testa a variável contra todos os valores listados
  - ▶ Se o valor dela não estiver listado, ativa o código em *default*
  - ▶ *default* é opcional – se não houver, simplesmente sai do *switch* e continua o programa
- E o break?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```



# Switch

- Switch testa a variável contra todos os valores listados
  - ▶ Se o valor dela não estiver listado, ativa o código em *default*
  - ▶ *default* é opcional – se não houver, simplesmente sai do *switch* e continua o programa
- E o break?
  - ▶ Faz com que o switch pare nesse ponto

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- O que acontece tirando o break?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
        case VINIL: valor = 1100;
        case FIBRA: valor = 750;
        case PLASTICO: valor = 500;
        default: valor = -1;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,ALVENARIA));
}
```

# Switch

- O que acontece tirando o break?

```
$ java AreaCasa  
-100.0
```

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    switch (material) {  
        case ALVENARIA: valor = 1500;  
        case VINIL: valor = 1100;  
        case FIBRA: valor = 750;  
        case PLASTICO: valor = 500;  
        default: valor = -1;  
    }  
    return(area*valor);  
}  
...  
public static void main(String[] args) {  
    System.out.println(valorPiscina(100,ALVENARIA));  
}
```

# Switch

- O que acontece tirando o break?

```
$ java AreaCasa  
-100.0
```

- ▶ Quando o switch começou, ALVENARIA fez com que valor=1500

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    switch (material) {  
        case ALVENARIA: valor = 1500;  
        case VINIL: valor = 1100;  
        case FIBRA: valor = 750;  
        case PLASTICO: valor = 500;  
        default: valor = -1;  
    }  
    return(area*valor);  
}  
...  
public static void main(String[] args) {  
    System.out.println(valorPiscina(100,ALVENARIA));  
}
```

# Switch

- O que acontece tirando o break?

```
$ java AreaCasa  
-100.0
```

- ▶ Quando o switch começou, ALVENARIA fez com que valor=1500
- ▶ Como não houve o break, ele continuo, e vez valor=1100, 750, 500 e, finalmente, -1

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    switch (material) {  
        case ALVENARIA: valor = 1500;  
        case VINIL: valor = 1100;  
        case FIBRA: valor = 750;  
        case PLASTICO: valor = 500;  
        default: valor = -1;  
    }  
    return(area*valor);  
}  
...  
public static void main(String[] args) {  
    System.out.println(valorPiscina(100,ALVENARIA));  
}
```

# Switch

- O que acontece tirando o break?

```
$ java AreaCasa  
-100.0
```

- ▶ Quando o switch começou, ALVENARIA fez com que valor=1500
- ▶ Como não houve o break, ele continuo, e vez valor=1100, 750, 500 e, finalmente, -1
- ▶ Então pegou esse valor=-1 e multiplicou por area

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;  
...  
static double valorPiscina(double area,  
                           int material) {  
    double valor;  
  
    switch (material) {  
        case ALVENARIA: valor = 1500;  
        case VINIL: valor = 1100;  
        case FIBRA: valor = 750;  
        case PLASTICO: valor = 500;  
        default: valor = -1;  
    }  
    return(area*valor);  
}  
...  
public static void main(String[] args) {  
    System.out.println(valorPiscina(100,ALVENARIA));  
}
```

# Switch

- E se agora removemos o *default*?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último break, por ser desnecessário

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```



# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último `break`, por ser desnecessário
- Qual a saída?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último break, por ser desnecessário
- Qual a saída?

```
$ javac AreaCasa.java
AreaCasa.java:85: variable valor might not have been
initialized
    return(area*valor);
                ^
1 error
```

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último break, por ser desnecessário
- Qual a saída?

```
$ javac AreaCasa.java
AreaCasa.java:85: variable valor might not have been
initialized
    return(area*valor);
                ^
1 error
```

- Então...

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor=0;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último `break`, por ser desnecessário
- Qual a saída?

```
$ javac AreaCasa.java
AreaCasa.java:85: variable valor might not have been
initialized
    return(area*valor);
                ^
1 error
```

- Então... E agora?

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor=0;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último break, por ser desnecessário
- Qual a saída?

```
$ javac AreaCasa.java
AreaCasa.java:85: variable valor might not have been
initialized
    return(area*valor);
                ^
1 error
```

- Então... E agora?

```
$ java AreaCasa
0.0
```

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor=0;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- E se agora removemos o *default*?
  - ▶ Note que também removemos o último break, por ser desnecessário
- Qual a saída?

```
$ javac AreaCasa.java
AreaCasa.java:85: variable valor might not have been
initialized
    return(area*valor);
                ^
1 error
```

- Então... E agora?

```
$ java AreaCasa
0.0
```

- Um valor inválido deixou de ser tratado...

```
static final int ALVENARIA = 0;
static final int VINIL = 1;
static final int FIBRA = 2;
static final int PLASTICO = 3;
...
static double valorPiscina(double area,
                           int material) {
    double valor=0;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                    break;
    }
    return(area*valor);
}
...
public static void main(String[] args) {
    System.out.println(valorPiscina(100,5));
}
```

# Switch

- Daria para enxugar o código, dado que há uma escolha e um retorno?

```
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

# Switch

- Daria para enxugar o código, dado que há uma escolha e um retorno?

```
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

```
static double valorPiscina(double area,
                           int material) {
    switch (material) {
        case ALVENARIA: return(area*1500);
        case VINIL: return(area*1100);
        case FIBRA: return(area*750);
        case PLASTICO: return(area*500);
        default: return(-1);
    }
}
```



# Switch

- Daria para enxugar o código, dado que há uma escolha e um retorno?
  - ▶ Ficou mais feio...

```
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

```
static double valorPiscina(double area,
                           int material) {
    switch (material) {
        case ALVENARIA: return(area*1500);
        case VINIL: return(area*1100);
        case FIBRA: return(area*750);
        case PLASTICO: return(area*500);
        default: return(-1);
    }
}
```

# Switch

- Daria para enxugar o código, dado que há uma escolha e um retorno?
  - ▶ Ficou mais feio...
  - ▶ Mas poupa memória, com a ausência de *valor*

```
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                       break;
        default: valor = -1;
    }
    return(area*valor);
}
```

```
static double valorPiscina(double area,
                           int material) {
    switch (material) {
        case ALVENARIA: return(area*1500);
        case VINIL: return(area*1100);
        case FIBRA: return(area*750);
        case PLASTICO: return(area*500);
        default: return(-1);
    }
}
```

# Switch

- Daria para enxugar o código, dado que há uma escolha e um retorno?
  - ▶ Ficou mais feio...
  - ▶ Mas poupa memória, com a ausência de *valor*
  - ▶ E é mais curto, sem os breaks

```
static double valorPiscina(double area,
                           int material) {
    double valor;

    switch (material) {
        case ALVENARIA: valor = 1500;
                        break;
        case VINIL: valor = 1100;
                    break;
        case FIBRA: valor = 750;
                    break;
        case PLASTICO: valor = 500;
                        break;
        default: valor = -1;
    }
    return(area*valor);
}
```

```
static double valorPiscina(double area,
                           int material) {
    switch (material) {
        case ALVENARIA: return(area*1500);
        case VINIL: return(area*1100);
        case FIBRA: return(area*750);
        case PLASTICO: return(area*500);
        default: return(-1);
    }
}
```

# Visão geral do código

```
class AreaCasa {
    /* valor do metro quadrado */
    static double valorM2 = 1500;
    /* materiais da piscina */
    static final int ALVENARIA = 0;
    static final int VINIL = 1;
    static final int FIBRA = 2;
    static final int PLASTICO = 3;

    static void areaCasa(float lateral,
                        float cquarto) {
        float areaq; // área do quarto
        float areas; // área da sala
        float areat; // área total

        if (lateral>=0 && cquarto>=0) {
            areas = lateral*lateral;
            System.out.println("Área da sala: "
                            +areas);
            areaq = cquarto*(lateral/2);
            System.out.println("Área do banheiro: "
                            +areaq);
            System.out.println("Área do quarto: "
                            +areaq);
            areat = areas + 2*areaq;
            System.out.println("Área total: "
                            +areat);
        }
        else
            System.out.println("Erro: parâmetro < 0");
    }

    static double areaPiscina(double raio) {
        return((raio >= 0) ? Math.PI * Math.pow(raio,2)
        )

    static double valorPiscina(double area,
                                int material) {
        double valor;

        switch (material) {
            case ALVENARIA: return(area*1500);
            case VINIL: return(area*1100);
            case FIBRA: return(area*750);
            case PLASTICO: return(area*500);
            default: return(-1);
        }
    }

    static double valorCasa(double area) {
        if (area >= 0) return(valorM2*area);
        return(-1);
    }

    public static void main(String[] args) {
        boolean valorOK = false;

        System.out.println(valorPiscina(100,PLASTICO));
    }
}
```

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.
  - ▶ Como faríamos?

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.
  - ▶ Como faríamos?
  - ▶ Primeira tentativa:

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?
- ▶ Primeira tentativa:

★ \t?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```



# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?
- ▶ Primeira tentativa:
  - ★ \t? – tabulação

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
                        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
                        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
                        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
                        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?
- ▶ Primeira tentativa:
  - ★ \t? – tabulação

- E qual a saída?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?
- ▶ Primeira tentativa:
  - ★ \t? – tabulação

- E qual a saída?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

Material	Valor
0	150000.0
1	110000.0
2	75000.0
3	50000.0

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?

- ▶ Primeira tentativa:

- ★ \t? – tabulação

- E qual a saída?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

Material	Valor
0	150000.0
1	110000.0
2	75000.0
3	50000.0

- Com 4 tipos foi fácil...

# Laços

- Queremos agora ter uma comparação entre o custo da piscina, para cada material.

- ▶ Como faríamos?
- ▶ Primeira tentativa:
  - ★ \t? – tabulação

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
                        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
                        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
                        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
                        valorPiscina(area,PLASTICO));  
}
```

- E qual a saída?

Material	Valor
0	150000.0
1	110000.0
2	75000.0
3	50000.0

- Com 4 tipos foi fácil...

- ▶ E se tivéssemos 20 tipos diferentes de material?

# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.
- Tem como melhorar?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.
- Tem como melhorar?
  - ▶ Repare nos valores dos tipos de material para a piscina:

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;
```

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```



# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.
- Tem como melhorar?
  - ▶ Repare nos valores dos tipos de material para a piscina:

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;
```

- ▶ São inteiros seqüenciais...

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.
- Tem como melhorar?
  - ▶ Repare nos valores dos tipos de material para a piscina:

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;
```

- ▶ São inteiros seqüenciais...
- ▶ E como usar isso?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

# Laços

- Fazer dessa forma, com 20 itens, ficaria trabalhoso.
- Tem como melhorar?
  - ▶ Repare nos valores dos tipos de material para a piscina:

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;
```

- ▶ São inteiros seqüenciais...
- ▶ E como usar isso?

```
public static void main(String[] args) {  
    double area = 100;  
    System.out.println("Material\tValor");  
    System.out.println(ALVENARIA+"\t\t"+  
        valorPiscina(area,ALVENARIA));  
    System.out.println(VINIL+"\t\t"+  
        valorPiscina(area,VINIL));  
    System.out.println(FIBRA+"\t\t"+  
        valorPiscina(area,FIBRA));  
    System.out.println(PLASTICO+"\t\t"+  
        valorPiscina(area,PLASTICO));  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
            valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- Como

```
static final int ALVENARIA = 0;  
static final int VINIL = 1;  
static final int FIBRA = 2;  
static final int PLASTICO = 3;
```

O código acima corresponde a

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = 0;  
  
    System.out.println("Material\tValor");  
    while (tipo <= 3) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

```
• while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

```
• while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}  
  
• enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- ```
while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

- ```
enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.

# While

- ```
while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```
- ```
enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.
  - 1 Inicialmente, testa a condição



# While

- ```
while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```
- ```
enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.
  - 1 Inicialmente, testa a condição
  - 2 Se verdadeira, executa o corpo

# While

```
• while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}  
  
• enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.
  - 1 Inicialmente, testa a condição
  - 2 Se verdadeira, executa o corpo
  - 3 Se falsa, sai do while

# While

```
• while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}  
  
• enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.
  - 1 Inicialmente, testa a condição
  - 2 Se verdadeira, executa o corpo
  - 3 Se falsa, sai do while
  - 4 Ao final do corpo, testa novamente a condição (voltando ao passo 1)

# While

```
• while (CONDIÇÃO) {  
    //condição verdadeira  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}  
  
• enquanto (CONDIÇÃO) {  
    COMANDO-1;  
    COMANDO-2;  
    ...  
    COMANDO-n;  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

- While diz que enquanto a condição for verdadeira, os comandos em seu corpo serão executados.
  - 1 Inicialmente, testa a condição
  - 2 Se verdadeira, executa o corpo
  - 3 Se falsa, sai do while
  - 4 Ao final do corpo, testa novamente a condição (voltando ao passo 1)
- Cada vez que o corpo é rodado chama-se iteração

# While

- Qual seria a condição para que o laço continue a executar?

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?
  - ▶ Que tipo <= PLASTICO

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?

- ▶ Que tipo <= PLASTICO
- ▶ Que tipo <= 3

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?
  - ▶ Que tipo <= PLASTICO
  - ▶ Que tipo <= 3
- E o que aconteceria se removêssemos a seguinte linha:

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```



# While

- Qual seria a condição para que o laço continue a executar?

- ▶ Que `tipo <= PLASTICO`
- ▶ Que `tipo <= 3`

- E o que aconteceria se removêssemos a seguinte linha:

- ▶ A condição sempre seria verdadeira, pois *tipo* sempre seria  $\leq 3$

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?

- ▶ Que `tipo <= PLASTICO`
- ▶ Que `tipo <= 3`

- E o que aconteceria se removêssemos a seguinte linha:

- ▶ A condição sempre seria verdadeira, pois *tipo* sempre seria  $\leq 3$
- ▶ Rodaria eternamente – Laço (ou loop) infinito

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?

- ▶ Que tipo  $\leq$  PLASTICO
- ▶ Que tipo  $\leq$  3

- E o que aconteceria se removêssemos a seguinte linha:

- ▶ A condição sempre seria verdadeira, pois *tipo* sempre seria  $\leq$  3
- ▶ Rodaria eternamente – Laço (ou loop) infinito

- Sabendo que *PLASTICO* = 3, há diferença entre o código acima e esse?

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= 3) {  
        System.out.println(tipo+"\t\t"+  
                             valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

# While

- Qual seria a condição para que o laço continue a executar?

- ▶ Que tipo  $\leq$  PLASTICO
- ▶ Que tipo  $\leq$  3

- E o que aconteceria se removêssemos a seguinte linha:

- ▶ A condição sempre seria verdadeira, pois *tipo* sempre seria  $\leq$  3
- ▶ Rodaria eternamente – Laço (ou loop) infinito

- Sabendo que *PLASTICO* = 3, há diferença entre o código acima e esse?

- ▶ Somente o acesso extra a *PLASTICO* na memória. Contudo, aumenta a legibilidade.

```
public static void main(String[] args) {
    double area = 100;
    int tipo = ALVENARIA;

    System.out.println("Material\tValor");
    while (tipo <= PLASTICO) {
        System.out.println(tipo+"\t\t"+
                           valorPiscina(area,tipo));
        tipo = tipo+1;
    }
}
```

```
public static void main(String[] args) {
    double area = 100;
    int tipo = ALVENARIA;

    System.out.println("Material\tValor");
    while (tipo <= 3) {
        System.out.println(tipo+"\t\t"+
                           valorPiscina(area,tipo));
        tipo = tipo+1;
    }
}
```

# While

- Já temos o orçamento para um determinado tamanho

# While

- Já temos o orçamento para um determinado tamanho
  - ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).

# While

- Já temos o orçamento para um determinado tamanho
  - ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).
    - ★ Digamos,  $50\text{m}^2$ ,  $100\text{m}^2$ ,  $150\text{m}^2$  e  $200\text{m}^2$

# While

- Já temos o orçamento para um determinado tamanho
  - ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).
    - ★ Digamos,  $50\text{m}^2$ ,  $100\text{m}^2$ ,  $150\text{m}^2$  e  $200\text{m}^2$
  - ▶ Como fazer?



# While

- Já temos o orçamento para um determinado tamanho
  - ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).
    - ★ Digamos,  $50\text{m}^2$ ,  $100\text{m}^2$ ,  $150\text{m}^2$  e  $200\text{m}^2$
  - ▶ Como fazer?

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
            valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

# While

- Já temos o orçamento para um determinado tamanho
  - ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).
    - ★ Digamos,  $50\text{m}^2$ ,  $100\text{m}^2$ ,  $150\text{m}^2$  e  $200\text{m}^2$
  - ▶ Como fazer?

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
            valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

- E a saída será:

# While

- Já temos o orçamento para um determinado tamanho

- ▶ Agora queremos poder comparar vários tamanhos de piscina de alvenaria (ou seja, fixando o material).

- ★ Digamos,  $50\text{m}^2$ ,  $100\text{m}^2$ ,  $150\text{m}^2$  e  $200\text{m}^2$

- ▶ Como fazer?

- E a saída será:

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
            valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

Área	Valor
50.0	75000.0
100.0	150000.0
150.0	225000.0
200.0	300000.0

# Laços

- Temos então, os seguintes códigos:

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
                           valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

# Laços

- Temos então, os seguintes códigos:

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
                           valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

- Como faríamos para criar uma tabela que desse o orçamento para piscina de várias áreas e materiais?

- Temos então, os seguintes códigos:

```
public static void main(String[] args) {  
    double area = 100;  
    int tipo = ALVENARIA;  
  
    System.out.println("Material\tValor");  
    while (tipo <= PLASTICO) {  
        System.out.println(tipo+"\t\t"+  
                           valorPiscina(area,tipo));  
        tipo = tipo+1;  
    }  
}
```

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = ALVENARIA;  
  
    System.out.println("Área\tValor");  
    while (area <= 200) {  
        System.out.println(area+"\t"+  
                           valorPiscina(area,tipo));  
        area = area+50;  
    }  
}
```

- Como faríamos para criar uma tabela que desse o orçamento para piscina de várias áreas e materiais?

para cada uma das áreas:

para cada um dos materiais:

    diga o valor da piscina com aquela  
    área, feita com aquele material

- Temos então, os seguintes códigos:

```
public static void main(String[] args) {
    double area = 100;
    int tipo = ALVENARIA;

    System.out.println("Material\tValor");
    while (tipo <= PLASTICO) {
        System.out.println(tipo+"\t\t"+
                           valorPiscina(area,tipo));
        tipo = tipo+1;
    }
}
```

```
public static void main(String[] args) {
    double area = 50;
    int tipo = ALVENARIA;

    System.out.println("Área\tValor");
    while (area <= 200) {
        System.out.println(area+"\t"+
                           valorPiscina(area,tipo));
        area = area+50;
    }
}
```

- Como faríamos para criar uma tabela que desse o orçamento para piscina de várias áreas e materiais?

para cada uma das áreas:

para cada um dos materiais:

digamos o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {
    double area = 50;
    int tipo;
    System.out.println("Área\tMaterial\tValor");
    while (area <= 200) {
        tipo = ALVENARIA;
        while (tipo <= PLASTICO) {
            System.out.println(area+"\t"+tipo+"\t\t"+
                               valorPiscina(area,tipo));
            tipo = tipo+1;
        }
        area = area+50;
    }
}
```

# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:  
  para cada um dos materiais:  
    diga o valor da piscina com aquela  
    área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```



# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:

para cada um dos materiais:

digamos o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:

para cada um dos materiais:

digite o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                                valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:

para cada um dos materiais:

digamos o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:

para cada um dos materiais:

digamos o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

- Laço while aninhado

# While

- Com mais calma... que código corresponde a que linha?

para cada uma das áreas:

para cada um dos materiais:

digamos o valor da piscina com aquela  
área, feita com aquele material

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

- Laço while aninhado
- Note que a cada iteração do laço externo (area), temos que reiniciar o laço interno (tipo)

# While

- Com mais calma... que código corresponde a que linha?

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                                valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

para cada uma das áreas:  
para cada um dos materiais:  
diga o valor da piscina com aquela  
área, feita com aquele material

- Laço while aninhado
- Note que a cada iteração do laço externo (area), temos que reiniciar o laço interno (tipo)
  - ▶ Fazendo tipo = ALVENARIA;

# While

- A saída será

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = 0;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

Área	Material	Valor
50.0	0	75000.0
50.0	1	55000.0
50.0	2	37500.0
50.0	3	25000.0
100.0	0	150000.0
100.0	1	110000.0
100.0	2	75000.0
100.0	3	50000.0
150.0	0	225000.0
150.0	1	165000.0
150.0	2	112500.0
150.0	3	75000.0
200.0	0	300000.0
200.0	1	220000.0
200.0	2	150000.0
200.0	3	100000.0

# While

- A saída será

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = 0;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

Área	Material	Valor
50.0	0	75000.0
50.0	1	55000.0
50.0	2	37500.0
50.0	3	25000.0
100.0	0	150000.0
100.0	1	110000.0
100.0	2	75000.0
100.0	3	50000.0
150.0	0	225000.0
150.0	1	165000.0
150.0	2	112500.0
150.0	3	75000.0
200.0	0	300000.0
200.0	1	220000.0
200.0	2	150000.0
200.0	3	100000.0

- E se removermos *tipo = ALVENARIA*;



# While

- A saída será

```
public static void main(String[] args) {  
    double area = 50;  
    int tipo = 0;  
    System.out.println("Área\tMaterial\tValor");  
    while (area <= 200) {  
        tipo = ALVENARIA;  
        while (tipo <= PLASTICO) {  
            System.out.println(area+"\t"+tipo+"\t\t"+  
                               valorPiscina(area,tipo));  
            tipo = tipo+1;  
        }  
        area = area+50;  
    }  
}
```

Área	Material	Valor
50.0	0	75000.0
50.0	1	55000.0
50.0	2	37500.0
50.0	3	25000.0
100.0	0	150000.0
100.0	1	110000.0
100.0	2	75000.0
100.0	3	50000.0
150.0	0	225000.0
150.0	1	165000.0
150.0	2	112500.0
150.0	3	75000.0
200.0	0	300000.0
200.0	1	220000.0
200.0	2	150000.0
200.0	3	100000.0

- E se removermos *tipo = ALVENARIA*;

Área	Material	Valor
50.0	0	75000.0
50.0	1	55000.0
50.0	2	37500.0
50.0	3	25000.0