

Aula 06 – Condicionais e Operadores Lógicos

Norton Trevisan Roman

4 de abril de 2013

Visão geral do código

- Considere o teste em valor()

```
class AreaCasa {
    static double valorM2 = 1500;

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

        System.out.println("Cálculo da área da casa");
        // cálculo da área da sala
        areas = lateral*lateral;
        System.out.println("A área da sala é "+areas);
        // cálculo da área do banheiro
        areaq = cquarto*(lateral/2);
        System.out.println("A área do banheiro é "
                        +areaq);
        // cálculo da área do quarto
        System.out.println("A área do quarto é "
                        +areaq);
        // cálculo da área total
        areat = areas + 2*areaq;
        System.out.println("A área total é " + areat);
    }

    static double areaPiscina(double raio) {
        return Math.PI * Math.pow(raio,2);
    }
}
```

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

public static void main(String[] args) {
    double preco; // área da piscina
    boolean valorOK = false;

    preco = valor(20);
    valorOK = preco >= 0;

    if (valorOK) System.out.println("O valor da
                                construção é "+preco);
    else System.out.println("Valor de área
                                negativo");
}
```

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um
 - ★ Se der problema, acuse o erro

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


Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um
 - ★ Se der problema, acuse o erro
 - ★ Senão, teste o segundo

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um
 - ★ Se der problema, acuse o erro
 - ★ Senão, teste o segundo
 - ★ Se der problema, acuse o erro

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um
 - ★ Se der problema, acuse o erro
 - ★ Senão, teste o segundo
 - ★ Se der problema, acuse o erro
 - ★ Senão, continue com o cálculo

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

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*

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

- E em *areaCasa*, como fazemos?

- ▶ Existem 2 parâmetros a serem testados

- ★ Primeiro teste um
- ★ Se der problema, acuse o erro
- ★ Senão, teste o segundo
- ★ Se der problema, acuse o erro
- ★ Senão, continue com o cálculo

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
        sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
            do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                +areaq);  
            System.out.println("A área do quarto é "+areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Condicionais Aninhados

- Vamos agora inserir o mesmo teste em *areaPiscina*
- E em *areaCasa*, como fazemos?
 - ▶ Existem 2 parâmetros a serem testados
 - ★ Primeiro teste um
 - ★ Se der problema, acuse o erro
 - ★ Senão, teste o segundo
 - ★ Se der problema, acuse o erro
 - ★ Senão, continue com o cálculo
 - ▶ Diz-se que os IFs estão aninhados ou encaixados

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

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
        sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
            do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                +areaq);  
            System.out.println("A área do quarto é "+areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos
 - ▶ Inclusive outro condicional

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos
 - ▶ Inclusive outro condicional

```
if (<condicao 1>) <comando 1>;  
else  
    if (<condicao 2>) <comando 2>;  
    else  
        if (<condicao 3>) <comando 3>;  
        else <comando 4>;
```


Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

- ▶ Inclusive outro condicional
- ▶ Note a ausência do `{}` – usamos quando há mais de um comando (um bloco)

```
if (<condicao 1>) <comando 1>;  
else  
    if (<condicao 2>) <comando 2>;  
    else  
        if (<condicao 3>) <comando 3>;  
        else <comando 4>;
```

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

- ▶ Inclusive outro condicional
- ▶ Note a ausência do `{}` – usamos quando há mais de um comando (um bloco)
- ▶ O `if...else` conta como um único comando

```
if (<condicao 1>) <comando 1>;  
else  
    if (<condicao 2>) <comando 2>;  
    else  
        if (<condicao 3>) <comando 3>;  
        else <comando 4>;
```

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

- ▶ Inclusive outro condicional
- ▶ Note a ausência do `{}` – usamos quando há mais de um comando (um bloco)
- ▶ O `if...else` conta como um único comando

```
if (<condicao 1>) <comando 1>;  
else  
    if (<condicao 2>) <comando 2>;  
    else  
        if (<condicao 3>) <comando 3>;  
        else <comando 4>;
```

- Identação é fundamental!

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

- ▶ Inclusive outro condicional
- ▶ Note a ausência do `{}` – usamos quando há mais de um comando (um bloco)
- ▶ O `if...else` conta como um único comando

- Identação é fundamental!

```
if (<condicao 1>) <comando 1>;
else
    if (<condicao 2>) <comando 2>;
    else
        if (<condicao 3>) <comando 3>;
        else <comando 4>;
```

```
if (<condicao 1>)
    if (<condicao 2>)
        if (<condicao 3>)
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

Condicionais Aninhados

- Dentro de um condicional pode ir qualquer tipo de comandos

- ▶ Inclusive outro condicional
- ▶ Note a ausência do `{}` – usamos quando há mais de um comando (um bloco)
- ▶ O `if...else` conta como um único comando

- Identação é fundamental!
 - ▶ Assim podemos ver qual *else* corresponde a qual *if*

```
if (<condicao 1>) <comando 1>;  
else  
    if (<condicao 2>) <comando 2>;  
    else  
        if (<condicao 3>) <comando 3>;  
        else <comando 4>;
```

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

Condicionais Aninhados

- Quando *comando 1* será executado?

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

Condicionais Aninhados

- Quando *comando 1* será executado?
 - ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

Condicionais Aninhados

- Quando *comando 1* será executado?
 - ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

- E quando *comando 2* será executado?

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

- ▶ Quando *condição 1* e *condição 2* forem verdadeiras e *condição 3* for falsa

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

- ▶ Quando *condição 1* e *condição 2* forem verdadeiras e *condição 3* for falsa

- E *comando 4*?

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

- ▶ Quando *condição 1* e *condição 2* forem verdadeiras e *condição 3* for falsa

- E *comando 4*?

- ▶ Quando *condição 1* for verdadeira e *condição 2* falsa

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

- ▶ Quando *condição 1* e *condição 2* forem verdadeiras e *condição 3* for falsa

- E *comando 4*?

- ▶ Quando *condição 1* for verdadeira e *condição 2* falsa

- *Comando 5*? ...

Condicionais Aninhados

- Quando *comando 1* será executado?

- ▶ Quando *condição 1*, *condição 2*, *condição 3* e *condição 4* forem verdadeiras

```
if (<condicao 1>)  
    if (<condicao 2>)  
        if (<condicao 3>)  
            if (<condicao 4>) <comando 1>;  
            else <comando 2>;  
        else <comando 3>;  
    else <comando 4>;  
else <comando 5>;
```

- E quando *comando 2* será executado?

- ▶ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

- E *comando 3*?

- ▶ Quando *condição 1* e *condição 2* forem verdadeiras e *condição 3* for falsa

- E *comando 4*?

- ▶ Quando *condição 1* for verdadeira e *condição 2* falsa

- *Comando 5*? ... quando *condição 1* for falsa

Condicionais Aninhados

- E se removermos essa linha?

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```


Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?

- ★ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?
 - ★ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa
- ▶ O compilador achará que o else é do if mais próximo

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?
 - ★ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa
- ▶ O compilador achará que o else é do if mais próximo
- ▶ Que fazer?

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?
 - ★ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa
- ▶ O compilador achará que o else é do if mais próximo
- ▶ Que fazer?

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

- Mudar a forma de entendimento com {}

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>) {
            if (<condicao 4>) <comando 1>;
        }
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

Condicionais Aninhados

- E se removermos essa linha?

- ▶ Quando *comando 3* será executado?
 - ★ Quando *condição 1*, *condição 2* e *condição 3* forem verdadeiras e *condição 4* for falsa
- ▶ O compilador achará que o else é do if mais próximo
- ▶ Que fazer?

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>
            if (<condicao 4>) <comando 1>;
            else <comando 2>;
            else <comando 3>;
        else <comando 4>;
    else <comando 5>;
```

- Mudar a forma de entendimento com {}

- ▶ Agora sim, o else está alinhado ao if correto

```
if (<condicao 1>
    if (<condicao 2>
        if (<condicao 3>) {
            if (<condicao 4>) <comando 1>;
        }
        else <comando 3>;
    else <comando 4>;
else <comando 5>;
```

- Voltemos ao main

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(20);  
    valorOK = preco >= 0;  
  
    if (valorOK) System.out.println("O valor da  
                                   construção é "+preco);  
    else System.out.println("Valor de área  
                             negativo");  
}
```

- Voltemos ao main
 - ▶ Podemos incrementar um pouco a resposta ao usuário

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(20);  
    valorOK = preco >= 0;  
  
    if (valorOK) System.out.println("O valor da  
                                   construção é "+preco);  
    else System.out.println("Valor de área  
                           negativo");  
}
```


- Voltemos ao main
 - ▶ Podemos incrementar um pouco a resposta ao usuário

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(20);  
    valorOK = preco >= 0;  
  
    if (valorOK) System.out.println("O valor da  
                                   construção é "+preco);  
    else System.out.println("Valor de área  
                           negativo");  
}
```

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(-20);  
    valorOK = preco >= 0;  
  
    System.out.print("O valor da construção ");  
    if (valorOK) System.out.println("é de "  
                                    +preco);  
    else System.out.println("não pode ser  
                           calculado: valor de área negativo");  
}
```

- Voltemos ao main
 - ▶ Podemos incrementar um pouco a resposta ao usuário
- Print faz o mesmo que println, exceto que não dá nova linha

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(20);  
    valorOK = preco >= 0;  
  
    if (valorOK) System.out.println("O valor da  
                                   construção é "+preco);  
    else System.out.println("Valor de área  
                           negativo");  
}
```

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(-20);  
    valorOK = preco >= 0;  
  
    System.out.print("O valor da construção ");  
    if (valorOK) System.out.println("é de "  
                                   +preco);  
    else System.out.println("não pode ser  
                           calculado: valor de área negativo");  
}
```

- Voltemos ao main
 - ▶ Podemos incrementar um pouco a resposta ao usuário
- Print faz o mesmo que println, exceto que não dá nova linha
 - ▶ Permite, assim, que possamos mudar parte da mensagem, conforme o resultado de algum condicional

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(20);  
    valorOK = preco >= 0;  
  
    if (valorOK) System.out.println("O valor da  
                                   construção é "+preco);  
    else System.out.println("Valor de área  
                           negativo");  
}
```

```
public static void main(String[] args) {  
    double preco; // área da piscina  
    boolean valorOK = false;  
  
    preco = valor(-20);  
    valorOK = preco >= 0;  
  
    System.out.print("O valor da construção ");  
    if (valorOK) System.out.println("é de "  
                                   +preco);  
    else System.out.println("não pode ser  
                           calculado: valor de área negativo");  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?
- O que eles significam?

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?
- O que eles significam?
 - ▶ Que o método acusará erro quando *lateral* < 0 ou *cquarto* < 0

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?
- O que eles significam?
 - ▶ Que o método acusará erro quando *lateral* < 0 ou *cquarto* < 0
- Precisamos de um meio de representar isso

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```


Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?
- O que eles significam?
 - ▶ Que o método acusará erro quando *lateral* < 0 ou *cquarto* < 0
- Precisamos de um meio de representar isso
 - ▶ *or*

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Considere o método para cálculo da área da casa
 - ▶ São necessários mesmo 2 IFs para isso?
- O que eles significam?
 - ▶ Que o método acusará erro quando *lateral* < 0 ou *cquarto* < 0
- Precisamos de um meio de representar isso
 - ▶ *or*
 - ▶ Operador Lógico

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- E como representar um *or* em Java?

Operadores Lógicos

- E como representar um *or* em Java?

▶ ||

```
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) System.out.println("Erro:  
                                                parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Operadores Lógicos

- E como representar um *or* em Java?
 - ▶ ||
- E como o compilador sabe que deve fazer o < antes?

```
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) System.out.println("Erro:  
                                                parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Operadores Lógicos

- E como representar um *or* em Java?
 - ▶ ||
- E como o compilador sabe que deve fazer o < antes?
 - ▶ Precedência: operadores relacionais têm precedência sobre operadores lógicos

```
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) System.out.println("Erro:  
                                                parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Operadores Lógicos

- Haveria um meio alternativo de se dizer que o método acusará erro quando *lateral < 0* ou *cquarto < 0*?

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- Haveria um meio alternativo de se dizer que o método acusará erro quando *lateral < 0* ou *cquarto < 0*?
 - ▶ O método irá executar o código somente se *lateral ≥ 0* e *cquarto ≥ 0*

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```


Operadores Lógicos

- Haveria um meio alternativo de se dizer que o método acusará erro quando *lateral < 0* ou *cquarto < 0*?
 - ▶ O método irá executar o código somente se *lateral ≥ 0* e *cquarto ≥ 0*
 - ▶ *and*

```
static void areaCasa(float lateral, float cquarto) {  
    float areaq; // área do quarto  
    float areas; // área da sala  
    float areat; // área total  
  
    if (lateral<0) System.out.println("Erro: lateral da  
                                     sala < 0");  
    else {  
        if (cquarto<0) System.out.println("Erro: lateral  
                                           do quarto < 0");  
        else {  
            System.out.println("Cálculo da área da  
                               casa");  
            areas = lateral*lateral;  
            System.out.println("A área da sala é "+areas);  
            areaq = cquarto*(lateral/2);  
            System.out.println("A área do banheiro é "  
                               +areaq);  
            System.out.println("A área do quarto é "  
                               +areaq);  
            areat = areas + 2*areaq;  
            System.out.println("A área total é " + areat);  
        }  
    }  
}
```

Operadores Lógicos

- E como representamos um *and* em java?

Operadores Lógicos

- E como representamos um *and* em java?

▶ &&

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código...

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código... O que ele diz agora?

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código... O que ele diz agora?
 - ▶ Se for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código... O que ele diz agora?
 - ▶ Se for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código
- Como faríamos para trocar de posição os códigos dentro do if e do else, usando o mesmo `&&`?

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código... O que ele diz agora?
 - ▶ Se for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código
- Como faríamos para trocar de posição os códigos dentro do if e do else, usando o mesmo `&&`?
 - ▶ Se **não** for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```


Operadores Lógicos

- Considere o código... O que ele diz agora?

- ▶ Se for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

- Como faríamos para trocar de posição os códigos dentro do if e do else, usando o mesmo `&&`?

- ▶ Se **não** for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

★ *not*

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Operadores Lógicos

- Considere o código... O que ele diz agora?

- ▶ Se for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

- Como faríamos para trocar de posição os códigos dentro do if e do else, usando o mesmo `&&`?

- ▶ Se **não** for verdade que $lateral \geq 0$ e $cquarto \geq 0$, execute o código

★ *not*

★ **!**

```
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))  
        System.out.println("Erro: parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1	condição2
V	V

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1	V	condição2
		V
		V

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1	condição2	
	V	F
V	V	

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1	condição2	
	V	F
V	V	F

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

		condição2	
		V	F
condição1	V	V	F
	F		

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

		condição2	
		V	F
condição1	V	V	F
	F	F	

```
if (condição1)
    if (condição2)
        comando1;
```


Tabela Verdade

- **And:** Comando1 será executado?

		condição2	
		V	F
condição1	V	V	F
	F	F	F

```
if (condição1)
    if (condição2)
        comando1;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador $*$ (\times), com $V > 0$ e $F = 0$

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	
V			

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

		condição2	
		V	F
condição1	V	V	F
	F	F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

		condição2	
		V	F
condição1	V	V	F
	F	F	V

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	V

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	V
F			

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	V
F		V	

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```


Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador * (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	V
F		V	F

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

Tabela Verdade

- **And:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	F
F		F	F

```
if (condição1)
    if (condição2)
        comando1;
```

- ▶ Equivale ao operador $*$ (\times), com $V > 0$ e $F = 0$

- **Or:** Comando1 será executado?

condição1		condição2	
		V	F
V		V	V
F		V	F

```
if (condição1) comando1;
else
    if (condição2) comando1;
    else
        comando2;
```

- ▶ Equivale ao operador $+$, com $V > 0$ e $F = 0$

Tabela Verdade

- **Not:** Comando1 será executado?

condição1

V

```
if (!condição1) comando1;
```

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
-----------	---	---

```
if (!condição1) comando1;
```

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	

```
if (!condição1) comando1;
```

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	V

```
if (!condição1) comando1;
```

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	V

```
if (!condição1) comando1;
```

- Operadores aritméticos, relacionais e lógicos podem ser misturados:

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	V

`if (!condição1) comando1;`

- Operadores aritméticos, relacionais e lógicos podem ser misturados:

```
int x = 3;
if (((20-x)>5) && ((4/x) == 1) ||
    ((16-x)>10))
    System.out.println("passou");
```


Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	V

`if (!condição1) comando1;`

- Operadores aritméticos, relacionais e lógicos podem ser misturados:
 - ▶ Como isso será entendido?

```
int x = 3;
if (((20-x)>5) && ((4/x) == 1) ||
    ((16-x)>10))
    System.out.println("passou");
```

Tabela Verdade

- **Not:** Comando1 será executado?

condição1	V	F
	F	V

if (!condição1) comando1;

- Operadores aritméticos, relacionais e lógicos podem ser misturados:

```
int x = 3;
if (((20-x)>5) && ((4/x) == 1) ||
    ((16-x)>10))
    System.out.println("passou");
```

- ▶ Como isso será entendido?

- ▶ Precedência:

<i>maior</i>	– (unário), !
	*, /, %
	+, –
↓	==, !=, >, <, >=, <=
	&&
<i>menor</i>	=

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	– (unário), ! *, /, % +, – ==, !=, >, <, >=, <=
	↓	&&
<i>menor</i>	<hr/>	=

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

- $x = 3$

<i>maior</i>	<hr/>	– (unário), ! *, /, % +, – ==, !=, >, <, >=, <= && =
<i>menor</i>	<hr/>	

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

• $x = 3$

▶ $V \ \&\& \ V \ || \ V \rightarrow$

<i>maior</i>	<hr/>	$-$ (unário), $!$
		$*, /, \%$
		$+, -$
\downarrow		$==, !=, >, <, >=, <=$
		$\&\&$
		$ $
<i>menor</i>	<hr/>	$=$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	\neg (unário), !
		*, /, %
		+, -
	↓	==, !=, >, <, >=, <=
		&&
<i>menor</i>	<hr/>	=

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow \text{Imprime}$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow \text{Imprime}$
- $x = 5$

<i>maior</i>	<hr/>	$-$ (unário), $!$
		$*, /, \%$
		$+, -$
\downarrow		$==, !=, >, <, >=, <=$
		$\&\&$
		$ $
<i>menor</i>	<hr/>	$=$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	\neg (unário), !
		*, /, %
		+, -
	↓	==, !=, >, <, >=, <=
		&&
<i>menor</i>	<hr/>	=

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow$ Imprime
- $x = 5$
 - ▶ $V \ \&\& \ F \ || \ V \rightarrow$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	\neg (unário), !
		*, /, %
		+, -
	↓	==, !=, >, <, >=, <=
		&&
<i>menor</i>		=
	<hr/>	

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow$ Imprime
- $x = 5$
 - ▶ $V \ \&\& \ F \ || \ V \rightarrow$ Imprime

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	– (unário), ! *, /, % +, – ==, !=, >, <, >=, <= && =
<i>menor</i>	<hr/>	

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow$ Imprime
- $x = 5$
 - ▶ $V \ \&\& \ F \ || \ V \rightarrow$ Imprime
- $x = 6$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	– (unário), ! *, /, % +, – ==, !=, >, <, >=, <= && =
<i>menor</i>	<hr/>	

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow$ Imprime
- $x = 5$
 - ▶ $V \ \&\& \ F \ || \ V \rightarrow$ Imprime
- $x = 6$
 - ▶ $V \ \&\& \ F \ || \ F \rightarrow$

Operadores Lógicos

```
int x = ?;  
if (((20-x)>5) && ((4/x) == 1) ||  
    ((16-x)>10))  
    System.out.println("passou");
```

<i>maior</i>	<hr/>	\neg (unário), !
		*, /, %
		+, -
	↓	==, !=, >, <, >=, <=
		&&
<i>menor</i>	<hr/>	=

- $x = 3$
 - ▶ $V \ \&\& \ V \ || \ V \rightarrow$ Imprime
- $x = 5$
 - ▶ $V \ \&\& \ F \ || \ V \rightarrow$ Imprime
- $x = 6$
 - ▶ $V \ \&\& \ F \ || \ F \rightarrow$ Não imprime

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

▶ `20 - x > 5 && 4 / x == 1 || 16 - x > 10`

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

- ▶ `20 - x > 5 && 4 / x == 1 || 16 - x > 10`
- ▶ `((20 - x) > 5) && ((4 / x) == 1) ||
((16 - x) > 10)`

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

- ▶ `20 - x > 5 && 4 / x == 1 || 16 - x > 10`
 - ▶ `((20 - x) > 5) && ((4 / x) == 1) ||
((16 - x) > 10)`

- Ou então, removendo os espaços...

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

- ▶ `20 - x > 5 && 4 / x == 1 || 16 - x > 10`
- ▶ `((20 - x) > 5) && ((4 / x) == 1) ||
((16 - x) > 10)`

- Ou então, removendo os espaços...

- ▶ `20-x>5&&4/x==1||16-x>10`

Atenção!

Use e abuse de parênteses

- O que é mais fácil de entender?

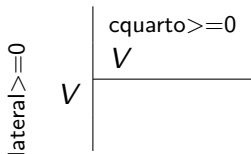
- ▶ `20 - x > 5 && 4 / x == 1 || 16 - x > 10`
 - ▶ `((20 - x) > 5) && ((4 / x) == 1) || ((16 - x) > 10)`

- Ou então, removendo os espaços...

- ▶ `20-x>5&&4/x==1||16-x>10`
 - ▶ `((20-x)>5)&&((4/x)==1)||((16-x)>10)`

Tabela Verdade Revista

- **And:** O cálculo da área será feito?



```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **And:** O cálculo da área será feito?

lateral >= 0		cquarto >= 0
	V	V
	V	V

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **And:** O cálculo da área será feito?

		cquarto >= 0	
lateral >= 0	V	V	F
		V	

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **And:** O cálculo da área será feito?

		cquarto >= 0	
lateral >= 0	V	V	F
		V	F

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral * lateral;  
        System.out.println("A área da sala é " + areas);  
        areaq = cquarto * (lateral / 2);  
        System.out.println("A área do banheiro é " + areaq);  
        System.out.println("A área do quarto é " + areaq);  
        areat = areas + 2 * areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **And:** O cálculo da área será feito?

		cquarto >= 0	
		V	F
lateral >= 0	V	V	F
	F	V	F

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **And:** O cálculo da área será feito?

		cquarto >= 0	
		V	F
lateral >= 0	V	V	F
	F	F	

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```


Tabela Verdade Revista

- **And:** O cálculo da área será feito?

		cquarto >= 0	
		V	F
lateral >= 0	V	V	F
	F	F	F

```
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) {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
    else  
        System.out.println("Erro: parâmetro < 0");  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

lateral<0	V	cquarto<0
		V

```
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) System.out.println("Erro:  
                                                    parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

lateral<0	V	cquarto<0
		V
		V

```
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) System.out.println("Erro:  
                                                    parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

		cquarto<0	
lateral<0	V	V	F
		V	

```
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) System.out.println("Erro:  
                                                    parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

		cquarto<0	
lateral<0	V	V	F
		V	V

```
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) System.out.println("Erro:  
                                                    parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

		cquarto<0	
		V	F
lateral<0	V	V	V
	F	V	V

```
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) System.out.println("Erro:  
                                                    parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

		cquarto<0	
		V	F
lateral<0	V	V	V
	F	V	

```
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) System.out.println("Erro:  
                                                parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

Tabela Verdade Revista

- **Or:** A mensagem de erro será mostrada?

		cquarto<0	
		V	F
lateral<0	V	V	V
	F	V	F

```
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) System.out.println("Erro:  
                                                parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areaq = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```


- **Not:** A mensagem de erro será mostrada?

lateral >= 0 && cquarto >= 0

V

```
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))  
        System.out.println("Erro: parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

- **Not:** A mensagem de erro será mostrada?

$lateral \geq 0 \ \&\& \ cquarto \geq 0$

V	F
---	---

```
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))  
        System.out.println("Erro: parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

- **Not:** A mensagem de erro será mostrada?

lateral >= 0 && cquarto >= 0

V		F
F		

```
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))  
        System.out.println("Erro: parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```

- **Not:** A mensagem de erro será mostrada?

$lateral \geq 0 \ \&\& \ cquarto \geq 0$	V	F
	F	V

```
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))  
        System.out.println("Erro: parâmetro < 0");  
    else {  
        System.out.println("Cálculo da área da casa");  
        areas = lateral*lateral;  
        System.out.println("A área da sala é "+areas);  
        areaq = cquarto*(lateral/2);  
        System.out.println("A área do banheiro é "+areaq);  
        System.out.println("A área do quarto é "+areaq);  
        areat = areas + 2*areaq;  
        System.out.println("A área total é " + areat);  
    }  
}
```