

Aula 11 – Condicionais (parte 2)

Norton T. Roman & Luciano A. Digiampietri

Condicionais Aninhados

- Vamos testar também os parâmetros de `areaPiscina`

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

Condicionais Aninhados

- Vamos testar também os parâmetros de `areaPiscina`

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

- E em `areaCasa`, como fazemos?

```
void areaCasa(float lateral,  
             float cquarto) {  
    float areaq;  
    float areas;  
    float areat;  
    ...  
}
```

- Existem 2 parâmetros a serem testados

Condicionais Aninhados

- Primeiro teste um

```
void areaCasa(float lateral, float quarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (quarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = quarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```

Condicionais Aninhados

- Primeiro teste um
- Se der problema, acuse o erro

```
void areaCasa(float lateral, float quarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (quarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = quarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```

Condicionais Aninhados

- Primeiro teste um
- Se der problema, acuse o erro
- Senão, teste o segundo

```
void areaCasa(float lateral, float quarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (quarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = quarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```

Condicionais Aninhados

- Primeiro teste um
- Se der problema, acuse o erro
- Senão, teste o segundo
- Se der problema, acuse o erro

```
void areaCasa(float lateral, float quarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (quarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = quarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```

Condicionais Aninhados

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

```
void areaCasa(float lateral, float quarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (quarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = quarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```


Condicionais Aninhados

- Diz-se que os IFs estão aninhados ou encaixados

```
void areaCasa(float lateral, float cquarto){
    float areaq;
    float areas;
    float areat;
    if (lateral<0) printf("Erro: lateral
                        da sala < 0\n");
    else {
        if (cquarto<0) printf("Erro: lateral
                            do quarto < 0\n");
        else {
            printf("Programa para cálculo ...\n");
            areas = lateral*lateral;
            printf("A área... é %f\n", areas);
            areaq = cquarto*(lateral/2);
            printf("A área... é %f\n", areaq);
            printf("A área... é %f\n", areaq);
            areat = areas + 2*areaq;
            printf("A área total é %f\n", areat);
        }
    }
}
```

Condicionais Aninhados

- Dentro de um condicional podemos ter qualquer tipo de 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 podemos ter qualquer tipo de comando

```
if (<condicao 1>) <comando 1>;
else
  if (<condicao 2>) <comando 2>;
  else
    if (<condicao 3>) <comando 3>;
    else <comando 4>;
```
- Inclusive outro condicional

Condicionais Aninhados

- Dentro de um condicional podemos ter qualquer tipo de comando

```
if (<condicao 1>) <comando 1>;
else
    if (<condicao 2>) <comando 2>;
    else
        if (<condicao 3>) <comando 3>;
        else <comando 4>;
```
- Inclusive outro condicional
- Note a ausência do `{}` → usamos quando há mais de um comando (um bloco)

Condicionais Aninhados

- Dentro de um condicional podemos ter qualquer tipo de comando

```
if (<condicao 1>) <comando 1>;
else
    if (<condicao 2>) <comando 2>;
    else
        if (<condicao 3>) <comando 3>;
        else <comando 4>;
```
- Inclusive outro condicional
- Note a ausência do `{}` → usamos quando há mais de um comando (um bloco)
- O `if...else` conta como um único comando

Condicionais Aninhados

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

```
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?

```
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>;
```

- Quando

condição 1,

condição 2,

condição 3 e

condição 4

forem

verdadeiras

Condicionais Aninhados

- E quando **comando 2** 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 quando **comando 2** 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 comando 3?

```
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 **comando 3**?

- Quando
condição 1 e
condição 2
forem
verdadeiras e
condição 3 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 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

- E **comando 4**?

- Quando
condição 1 for
verdadeira e
condição 2
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 comando 5?

```
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 **comando 5**?
- Quando condição 1 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 uma linha, quando **comando 3** será executado?

```
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 uma linha, quando **comando 3** será executado?

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

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

Condicionais Aninhados

- 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 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- 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 3>;
        else <comando 4>;
    else <comando 5>;
```

Condicionais Aninhados

- 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 3>;
    else <comando 4>;
else <comando 5>;
```

- Que fazer?
- Mudar a forma de entendimento com {}

Condicionais Aninhados

- 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 3>;
    else <comando 4>;
else <comando 5>;
```

- Que fazer?

- Mudar a forma de entendimento com {}

- Agora sim, o else está alinhado ao if correto

printf

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

```
int main() {
    double preco;
    bool valorOK = false;

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

    printf("O valor da construção ");
    if (valorOK) printf("é %f\n",preco);
    else printf("não foi obtido: área
                negativa\n");

    return 0;
}
```

- Permite, assim, que possamos mudar parte da mensagem, conforme o resultado de algum condicional

```
int main() {
    double preco;
    bool valorOK = false;

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

    printf("O valor da construção ");
    if (valorOK) printf("é %f\n",preco);
    else printf("não foi obtido: área
                negativa\n");

    return 0;
}
```


Aula 11 – Condicionais (parte 2)

Norton T. Roman & Luciano A. Digiampietri