

# Aula 13 – Condicionais (parte 3)

Norton T. Roman & Luciano A. Digiampietri

# Switch

- Considere o código que calcula o valor da construção:

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

# Switch

- Considere o código que calcula o valor da construção:

```
double valor(double area) {  
    if (area >= 0)  
        return valorM2*area;  
    return -1;  
}
```
- Suponha agora que haja quatro materiais, com preços diferentes, para a piscina

# Switch

- Considere o código que calcula o valor da construção:

```
double valor(double area) {  
    if (area >= 0)  
        return valorM2*area;  
    return -1;  
}
```
- Suponha agora que haja quatro materiais, com preços diferentes, para a piscina
- Teremos que desmembrar o código, criando uma função apenas para a piscina. Como fazer?

# Switch

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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);
}
```

- **Constantes** também podem ser usadas para definir categorias

# Switch

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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);
}
```

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

# Switch

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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);
}
```

- **Constantes** também podem ser usadas para definir categorias
- Deixam o código bem mais legível
- Haveria como evitar esse **aninhamento**?



# Switch

```
switch (variavel) {  
    case valor1: código1  
        break;  
    case valor2: código2  
        break;  
    ...  
    case valorn: códigon  
        break;  
    default: códigopadrão;  
}
```

Correspondendo a:

```
if (variavel == valor1)  
    código1;  
else  
    if (variavel == valor2)  
        código2;  
    else  
        ...  
    else  
        if (variavel == valorn)  
            códigon;  
        else códigopadrão;
```

# Switch

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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);
}
```

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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**

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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);
}
```

- E o `break`?

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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

- E o `break`?
- Faz com que o switch pare nesse ponto

```
#define ALVENARIA 0
#define VINIL 1
#define FIBRA 2
#define PLASTICO 3
...
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 se tirarmos o break?

```
...
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```



# Switch

- O que acontece se tirarmos o break?

```
$ ./AreaCasa  
-100.00
```

```
...  
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);  
}  
  
...  
int main() {  
    printf("%.2f\n",  
          valorPiscina(100,ALVENARIA));  
    return 0;  
}
```

# Switch

- O que acontece se tirarmos o break?  
\$ ./AreaCasa  
-100.00
- Quando o switch começou, **ALVENARIA** fez valor=1500

```
...
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Como não houve o break, ele continuou, e fez valor=1100, 750, 500 e, finalmente, -1

```
...
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Como não houve o `break`, ele continuou, e fez `valor=1100, 750, 500` e, finalmente, `-1`
- Então pegou esse valor `-1` e multiplicou por `area`, retornando

```
...
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Como não houve o break, ele continuou, e fez valor=1100, 750, 500 e, finalmente, -1
- Então pegou esse valor -1 e multiplicou por area, retornando
- case define o ponto de entrada. Só isso.

```
...
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- É se agora removemos o default?
- Note que também removemos o último break, por ser desnecessário
- Irá compilar?

```
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Em C, irá compilar.

```
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Em C, irá compilar.
  - Porém, talvez a variável valor não receba nenhum valor dentro do switch (e não havia sido inicializada em sua criação).

```
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);
}
...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```



# Switch

- Em C, irá compilar.
  - Porém, talvez a variável valor não receba nenhum valor dentro do switch (e não havia sido inicializada em sua criação).
  - Neste caso, qual valor ela possuirá?

```
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);
}
...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Em C, irá compilar.
  - Porém, talvez a variável valor não receba nenhum valor dentro do switch (e não havia sido inicializada em sua criação).
  - Neste caso, qual valor ela possuirá?
  - O valor que estiver em sua memória!

```
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);
}
...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

- Então inicializemos...

```
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Então inicializemos...
- E agora?

```
$ ./AreaCasa
```

```
0.00
```

```
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);  
}  
...  
int main() {  
    printf("%.2f\n",  
          valorPiscina(100,ALVENARIA));  
    return 0;  
}
```

# Switch

- Então inicializemos...
- E agora?  
\$ ./AreaCasa  
0.00
- Um valor inválido deixou de ser tratado

```
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

- Então inicializemos...
- E agora?  
\$ ./AreaCasa  
0.00
- Um valor inválido deixou de ser tratado
- Neste caso, deveríamos inicializar com **-1**

```
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);
}

...
int main() {
    printf("%.2f\n",
           valorPiscina(100,ALVENARIA));
    return 0;
}
```

# Switch

```
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);
}
```

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

# Switch

```
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);
}
```

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

```
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;
    }
}
```



# Aula 13 – Condicionais (parte 3)

Norton T. Roman & Luciano A. Digiampietri