

Aula 15 – Laços (parte 2)

Norton T. Roman & Luciano A. Digiampietri

Do ... While

- Existe ainda um outro tipo de laço: do...while

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- Existe ainda um outro tipo de laço: do...while

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);  
  
int main() {  
    double area = 50;  
    int tipo = ALVENARIA;  
    printf("Área\tValor\n");  
    do {  
        printf("%6.1f\t%9.2f\n", area,  
            valorPiscina(area,tipo));  
        area = area+50;  
    } while (area <= 200);  
    return 0;  
}
```

Do ... While

- O while faz o teste antes de rodar o laço pela primeira vez
- O do...while faz o teste depois de rodar o laço pela primeira vez
 - Rodando novamente apenas se o teste for positivo

```
while (condição) {  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- E quando usar um ou o outro?

```
while (condição) {  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- E quando usar um ou o outro?
- Depende de quando o teste deve ser feito:
 - Se antes ou depois do corpo do laço rodar uma vez

```
while (condição) {  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- E qual dentre o *while* e *do...while* é melhor?

Do ... While

- E qual dentre o *while* e *do...while* é melhor?
- São totalmente equivalentes

Do ... While

- E qual dentre o *while* e *do...while* é melhor?

- São totalmente equivalentes

- Todo *while* pode ser escrito como *do... while*

```
while (condição) {  
    comando;  
}
```

```
if (condição) {  
    do {  
        comando;  
    } while (condição);  
}
```

Do ... While

- E qual dentre o *while* e *do...while* é melhor?

```
do {  
    comando;  
} while (condição);
```

- E vice versa:

```
condição = true;  
while (condição) {  
    comando;  
    recalcula_condição;  
}
```

Do ... While

- O que ditará qual deles será usado é a conveniência para o programador

```
while (condição) {  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- O que ditará qual deles será usado é a conveniência para o programador
- Será escolhido, naturalmente, aquele que exigir a escrita de menos código

```
while (condição) {  
    comando1;  
    comando2;  
    ...  
    comandon;  
}
```

```
do {  
    comando1;  
    comando2;  
    ...  
    comandon;  
} while (condição);
```

Do ... While

- Em nosso código, esses trechos são equivalentes:

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    while (area <= 200) {
        printf("%6.1f\t%9.2f\n", area,
            valorPiscina(area,tipo));

        area = area+50;
    }
    return 0;
}
```

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    if (area <= 200) {
        do {
            printf("%6.1f\t%9.2f\n",
                area,
                valorPiscina(area,tipo));
            area = area + 50;
        } while (area<=200);
    }
    return 0;
}
```

Do ... While

- Nesse caso, contudo, o **if** sempre será verdadeiro

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    while (area <= 200) {
        printf("%6.1f\t%9.2f\n", area,
            valorPiscina(area,tipo));

        area = area+50;
    }
    return 0;
}
```

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    if (area <= 200) {
        do {
            printf("%6.1f\t%9.2f\n",
                area,
                valorPiscina(area,tipo));
            area = area + 50;
        } while (area<=200);
    }
    return 0;
}
```

Do ... While

- Então podemos removê-lo

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    while (area <= 200) {
        printf("%6.1f\t%9.2f\n", area,
            valorPiscina(area,tipo));

        area = area+50;
    }
    return 0;
}
```

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Área\tValor\n");
    do {
        printf("%6.1f\t%9.2f\n",
            area,
            valorPiscina(area,tipo));
        area = area + 50;
    } while (area<=200);
    return 0;
}
```

Do ... While

Da mesma forma, esses trechos são equivalentes:

```
int main() {
    double area = 100;
    int tipo = ALVENARIA;

    printf("Material\tValor\n");
    while (tipo <= PLASTICO) {
        printf("%8i\t%9.2f\n", tipo,
            valorPiscina(area,tipo));
        tipo = tipo+1;
    }
    return 0;
}
```

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Material\tValor\n");
    if (tipo <= PLASTICO) {
        do {
            printf("%8i\t%9.2f\n",tipo,
                valorPiscina(area,tipo));
            tipo = tipo + 1;
        } while (tipo <= PLASTICO);
    }
    return 0;
}
```


Do ... While

Também aqui o if era redundante

```
int main() {
    double area = 100;
    int tipo = ALVENARIA;

    printf("Material\tValor\n");
    while (tipo <= PLASTICO) {
        printf("%8i\t%9.2f\n", tipo,
            valorPiscina(area,tipo));
        tipo = tipo+1;
    }
    return 0;
}
```

```
int main() {
    double area = 50;
    int tipo = ALVENARIA;

    printf("Material\tValor\n");
    do {
        printf("%8i\t%9.2f\n", tipo,
            valorPiscina(area,tipo));
        tipo = tipo + 1;
    } while (tipo <= PLASTICO);
    return 0;
}
```

Do ... While

- E, finalmente, assim como no while...

```
int main() {
    double area = 50;
    int tipo;

    printf("Área\tTipo\tValor\n");
    while (area <= 200) {
        tipo = ALVENARIA;
        while (tipo <= PLASTICO) {
            printf("%6.1f\t%4i\t%9.2f\n",
                area, tipo,
                valorPiscina(area,tipo));
            tipo = tipo+1;
        }
        area = area+50;
    }
    return 0;
}
```

Do ... While

- Também o do...while pode estar aninhado

```
int main() {
    double area = 50;
    int tipo;

    printf("Área\tTipo\tValor\n");
    do {
        tipo = ALVENARIA;
        do {
            printf("%6.1f\t%4i\t%9.2f\n",
                area, tipo,
                valorPiscina(area,tipo));
            tipo = tipo+1;
        } while (tipo <= PLASTICO);
        area = area+50;
    } while (area <= 200);
    return 0;
}
```

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Primeiro testa a condição

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Primeiro testa a condição
- Se ela for verdadeira, executa o laço (corpo do `while`)

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Primeiro testa a condição
- Se ela for verdadeira, executa o laço (corpo do while)
- Primeiro executa o laço (corpo do do...while)

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Primeiro testa a condição
- Se ela for verdadeira, executa o laço (corpo do while)
- Primeiro executa o laço (corpo do do...while)
- Ao final do corpo, testa então a condição

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Ao final do corpo, volta ao início, testando novamente a condição

Em Suma...

```
while (condição) {  
    comando;  
}
```

```
do {  
    comando;  
} while (condição);
```

- Ao final do corpo, volta ao início, testando novamente a condição
- Se ela for falsa, passa à próxima instrução após o `while`

Em Suma...

```
while (condição) {  
    comando;  
}
```

- Ao final do corpo, volta ao início, testando novamente a condição
- Se ela for falsa, passa à próxima instrução após o `while`

```
do {  
    comando;  
} while (condição);
```

- Se a condição for verdadeira, executa o laço novamente

Em Suma...

```
while (condição) {  
    comando;  
}
```

- Ao final do corpo, volta ao início, testando novamente a condição
- Se ela for falsa, passa à próxima instrução após o `while`

```
do {  
    comando;  
} while (condição);
```

- Se a condição for verdadeira, executa o laço novamente
- Se ela for falsa, passa à próxima instrução após o `do...while`

Aula 15 – Laços (parte 2)

Norton T. Roman & Luciano A. Digiampietri