

# LINGUAGEM DE PROGRAMAÇÃO II

---

© 2022, Gizelle Kupac Vianna (DECOMP/UFRRJ)

# ESTRUTURAS DE CONTROLE DE FLUXO

---

## Aula 2

# Estruturas de Seleção

- **IF SIMPLES:**

```
if (<cond>)  
    <comando>;
```

```
if (<cond>) {  
    <comando>;  
    <comando>;  
    <comando>;  
}
```

- Exemplo: Se o grau do aluno for maior ou igual a 7, imprimir “aprovado”

```
if (grau >= 7)  
    printf("aprovado\n");
```

# Estruturas de Seleção

- **IF/ELSE:**

```
if (<cond>)  
    <comando>;  
else  
    <comando>;
```

```
if (<cond>) {  
    <comando>;  
    <comando>;  
}  
else {  
    <comando>;  
    <comando>;  
}
```

# Estruturas de Seleção

- Exemplo: Se o grau do aluno for maior ou igual a 7, imprimir “aprovado”. Caso contrário, imprimir “reprovado”.

```
if (grau >= 7)
    printf("aprovado\n");
else {
    printf("reprovado\n");
    printf("você deve refazer o curso!");
}
```

# Estruturas de Seleção

- **IF/ELSE Aninhados:**

```
if (<cond>)
    <comando>;
else if (<cond>)
    <comando>;
else if (<cond>) {
    <comando>;
    <comando>;
}
else {
    <comando>;
    <comando>;
}
```

# Estruturas de Seleção

- Exemplo:

*Se o grau do aluno for maior ou igual a 9,  
imprimir “A”*

*senão se o grau do aluno for maior ou igual a 8,  
imprimir “B”*

*senão se o grau do aluno for maior ou igual a 7,  
imprimir “C”*

*senão se o grau do aluno for maior ou igual a 6,  
imprimir “D”*

*senão  
imprimir “E”*

# Estruturas de Seleção

```
if (grau >= 9)
    printf("A\n");
else if (grau >= 8)
    printf("B\n");
else if (grau >= 7)
    printf("C\n");
else if (grau >= 6)
    printf("D\n");
else
    printf("E\n");
```



# Estruturas de Seleção

- **SWITCH (Estrutura de múltipla escolha):**

```
switch (grau) {  
    case 10 : printf("A\n");  
        break;  
    case 9  : printf("A\n");  
        break;  
    case 8  : printf("B\n");  
        break;  
    case 7  : printf("C\n");  
        break;  
    case 6  : printf("D\n");  
        break;  
    default: printf("E\n");  
        break;  
}
```

# Estrutura de Repetição com Teste no Início

- **WHILE:**

```
while (<cond>)  
    <comando>;
```

```
while (<cond>) {  
    <comando>;  
    <comando>;  
    <comando>;  
}
```

- Exemplo: Escreva um programa que calcule a média de um teste realizado por uma turma de 10 alunos.

```
#include <stdio.h>

main () {
    int contador, grau, total, media;

    total = 0;
    contador = 1;

    while (contador <= 10) {
        printf ("Entre com o grau: ");
        scanf ("%d", &grau);
        total = total + grau;
        contador = contador + 1;
    }

    media = total / 10;
    printf ("A média da turma é: %d/n", media);

    return 0;
}
```

# Exercício

- Usando, obrigatoriamente, uma estrutura WHILE, escreva um programa para calcular a média de uma turma de qualquer tamanho.

```
#include <stdio.h>

main () {
    float media;
    int cont, grau, total;

    total = 0;
    cont = 0;

    printf( "Entre com o grau, -1 para finalizar:");
    scanf("%d", &grau);

    while (grau != -1) {
        total += grau;
        cont++;
        printf("Entre com o grau, -1 para finalizar:");
        scanf("%d", &grau);
    }

    if (cont != 0) {
        media = (float) total / cont;
        printf("A media da turma é: %.2f", media);
    }
    else
        printf("Nenhum grau foi fornecido\n");

    return 0;
}
```

# Estrutura de Repetição com Contador

- **FOR:**

```
for (<inic>; <parada>; <atualização>)  
    <comando>;
```

```
for (cont = 1; cont <= 10 ; cont++)  
    printf("%d\n", cont);
```

- As estruturas while e for podem ser construídas para executar a mesma tarefa. Veja a correspondência entre while e for:

```
expr1;  
while (expr2) {  
    comando;  
    expr3;  
}
```

```
for (expr1; expr2; expr3)  
    comando;
```

- Onde:

- expr1 = inicialização do contador
- expr2 = teste de fim de execução
- expr3 = incremento do contador

# Estrutura de Repetição com Teste no Final

- **DO / WHILE:**

```
do {  
    <comando>;  
} while (<expressão>)
```

- **Exemplo:**

```
#include <stdio.h>  
  
main () {  
    int cont = 1;  
  
    do {  
        printf("%d  ", cont);  
    } while (++contador <= 10 );  
  
    return 0;  
}
```

# Break e Continue

- O comando **break** é usado para encerrar **completamente** um loop.
- O comando **continue** é usado para encerrar **apenas** a execução corrente do loop.
- Exemplos: Qual é a saída de cada programa?

```
#include <stdio.h>
```

```
main () {  
    int x;  
  
    for ( x = 1; x <= 10; x++) {  
        if (x == 5)  
            break;  
        printf("%d ", x);  
    }  
  
    printf ("\nSaiu do loop em x igual a: %d\n", x);  
    return 0;  
}
```

Resp: 1 2 3 4  
Saiu do loop em x igual a: 5



# Break e Continue

```
#include <stdio.h>

main () {
    int x;

    for ( x = 1; x <= 10; x++) {
        if (x == 5)
            continue;
        printf("%d ", x);

        printf ("\nContinue usado para ignorar o
valor: %d\n", x);
        return 0;
    }
```

Resp: 1 2 3 4 6 7 8 9 10

Continue usado para ignorar o valor: 5