

# Estrutura de Repetição

Informática – IFSULDEMINAS

Primeiro Semestre de 2013

# Roteiro

- 1 Introdução
- 2 Comando for
- 3 Comando while
- 4 Comando do - while
- 5 Loop infinito
- 6 Exercícios Resolvidos

# Problema

Considere o problema a seguir:

## Problema

Um médico deseja fazer uma estatística em relação à obesidade de seus 550 pacientes. Para isso ele precisa saber o peso e a altura de cada paciente. Você é o responsável por resolver este problema.

# Uma solução ...

## Índice de Massa Corporal

```
float peso1,..., peso550, altura1, ..., altura550, imc1,  
    ..., imc550;
```

```
printf("Qual o seu peso e sua altura?\n"),  
scanf("%f%f", &peso1, &altura1);  
imc1 = peso1/(altura1*altura1);
```

```
printf("Qual o seu peso e sua altura?\n"),  
scanf("%f%f", &peso2, &altura2);  
imc2 = peso2/(altura2*altura2);  
...
```

```
printf("Qual o seu peso e sua altura?\n"),  
scanf("%f%f", &peso550, &altura550);  
imc550 = peso550/(altura550*altura550);
```

```
// imprime o imc de cada paciente
```

# Considerações

- Escrevemos vários blocos de código com a mesma função.
- Blocos errados podem ser replicados por todo código.
- O código se torna muito grande.
- Declaramos muitas variáveis.

E se pudéssemos repetir os blocos parecidos usando um *loop*?

# Considerações

- Escrevemos vários blocos de código com a mesma função.
- Blocos errados podem ser replicados por todo código.
- O código se torna muito grande.
- Declaramos muitas variáveis.

E se pudéssemos repetir os blocos parecidos usando um *loop*?

# Considerações

- Escrevemos vários blocos de código com a mesma função.
- Blocos errados podem ser replicados por todo código.
- O código se torna muito grande.
- Declaramos muitas variáveis.

E se pudéssemos repetir os blocos parecidos usando um *loop*?

# Considerações

- Escrevemos vários blocos de código com a mesma função.
- Blocos errados podem ser replicados por todo código.
- O código se torna muito grande.
- Declaramos muitas variáveis.

E se pudéssemos repetir os blocos parecidos usando um *loop*?

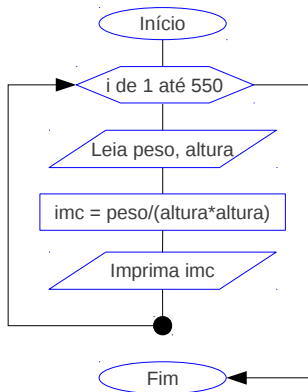


# Considerações

- Escrevemos vários blocos de código com a mesma função.
- Blocos errados podem ser replicados por todo código.
- O código se torna muito grande.
- Declaramos muitas variáveis.

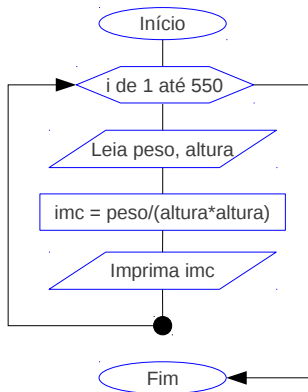
E se pudéssemos repetir os blocos parecidos usando um *loop*?

# Simplificando ...



Conseguimos otimizar trechos de código utilizando laços de repetição

## Simplificando ...



Conseguimos otimizar trechos de código utilizando laços de repetição

# Comando for

```
for  
for (inicialização; condição; incremento) {  
    declaração;  
}
```

- Esta estrutura de repetição, geralmente é utilizada, quando o número de repetições é definido.
- A **inicialização** define o valor inicial de uma variável contadora.
- A repetição continua enquanto a **condição** for verdadeira – é a condição de parada.
- O **incremento** ou **decremento** define como a variável contadora tem seu valor modificado.

# Comando for

```
for  
for (inicialização; condição; incremento) {  
    declaração;  
}
```

- Esta estrutura de repetição, geralmente é utilizada, quando o número de repetições é definido.
- A **inicialização** define o valor inicial de uma variável contadora.
- A repetição continua enquanto a **condição** for verdadeira – é a condição de parada.
- O **incremento** ou **decremento** define como a variável contadora tem seu valor modificado.

# Comando for

```
for  
for (inicialização; condição; incremento) {  
    declaração;  
}
```

- Esta estrutura de repetição, geralmente é utilizada, quando o número de repetições é definido.
- A **inicialização** define o valor inicial de uma variável contadora.
- A repetição continua enquanto a **condição** for verdadeira – é a condição de parada.
- O **incremento** ou **decremento** define como a variável contadora tem seu valor modificado.

# Comando for

```
for  
for (inicialização; condição; incremento) {  
    declaração;  
}
```

- Esta estrutura de repetição, geralmente é utilizada, quando o número de repetições é definido.
- A **inicialização** define o valor inicial de uma variável contadora.
- A repetição continua enquanto a **condição** for verdadeira – é a condição de parada.
- O **incremento** ou **decremento** define como a variável contadora tem seu valor modificado.

# Comando for

```
for  
for (inicialização; condição; incremento) {  
    declaração;  
}
```

- Esta estrutura de repetição, geralmente é utilizada, quando o número de repetições é definido.
- A **inicialização** define o valor inicial de uma variável contadora.
- A repetição continua enquanto a **condição** for verdadeira – é a condição de parada.
- O **incremento** ou **decremento** define como a variável contadora tem seu valor modificado.



# Usando o comando for

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

for (cont=1; cont<=550; cont++) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
}
```

- A variável contadora, **cont**, começa com 1 e incrementa até 550, quando a repetição termina.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando for

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

for (cont=1; cont<=550; cont++) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
}
```

- A variável contadora, **cont**, começa com 1 e incrementa até 550, quando a repetição termina.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando for

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

for (cont=1; cont<=550; cont++) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
}
```

- A variável contadora, **cont**, começa com 1 e incrementa até 550, quando a repetição termina.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Comando while

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

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- A repetição continua enquanto a **condição** for verdadeira.

# Comando while

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

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- A repetição continua enquanto a **condição** for verdadeira.

# Comando while

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

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- A repetição continua enquanto a **condição** for verdadeira.

# Usando o comando while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

while (cont<=550) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

while (cont<=550) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.



# Usando o comando while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

while (cont<=550) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  550 for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

while (cont<=550) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

while (cont<=550) {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

## Comando do - while

```
for  
do {  
    declaração;  
} while(condição);
```

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- Executa pelo menos uma vez o bloco.
- A repetição continua enquanto a **condição** for verdadeira.

## Comando do - while

```
for  
do {  
    declaração;  
} while(condição);
```

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- Executa pelo menos uma vez o bloco.
- A repetição continua enquanto a **condição** for verdadeira.

## Comando do - while

```
for  
do {  
    declaração;  
} while(condição);
```

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- Executa pelo menos uma vez o bloco.
- A repetição continua enquanto a **condição** for verdadeira.

## Comando do - while

```
for  
do {  
    declaração;  
} while(condição);
```

- Esta estrutura de repetição, geralmente é utilizada, quando não se sabe o número de vezes que se deve repetir.
- Executa pelo menos uma vez o bloco.
- A repetição continua enquanto a **condição** for verdadeira.

# Usando o comando do - while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

do {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}while (cont<=550);
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.



# Usando o comando do - while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

do {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}while (cont<=550);
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando do - while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

do {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}while (cont<=550);
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando do - while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

do {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}while (cont<=550);
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Usando o comando do - while

## Índice de Massa Corporal

```
float cont, peso, altura, imc;

cont = 1;

do {
    printf("Qual o seu peso e sua altura?\n"),
    scanf("%f%f", &peso, &altura);
    imc = peso/(altura*altura);
    printf("Imc: %f", imc);
    cont++;
}while (cont<=550);
```

- A variável contadora, **cont**, foi inicializada com o valor 1.
- A medida em que as repetições são feitas, **cont** é incrementado.
- A repetição continua enquanto **cont**  $\leq$  **550** for verdadeira.
- Em cada execução da repetição o **imc** de um paciente é calculado.

# Loop infinito

- Loop infinito ocorre quando a repetição executa indefinidamente – a condição de parada nunca é alcançada.
- Todas as estruturas de repetição podem entrar em loop:
  - ▶ **for**: basta omitir os parâmetros do for.
  - ▶ **while** e **do - while**: basta fazer a condição sempre ser verdadeira.
- Há casos em que problemas podem ser resolvidos usando loop infinito.

# Loop infinito

- Loop infinito ocorre quando a repetição executa indefinidamente – a condição de parada nunca é alcançada.
- Todas as estruturas de repetição podem entrar em loop:
  - ▶ **for**: basta omitir os parâmetros do for.
  - ▶ **while** e **do - while**: basta fazer a condição sempre ser verdadeira.
- Há casos em que problemas podem ser resolvidos usando loop infinito.

# Loop infinito

- Loop infinito ocorre quando a repetição executa indefinidamente – a condição de parada nunca é alcançada.
- Todas as estruturas de repetição podem entrar em loop:
  - ▶ **for**: basta omitir os parâmetros do for.
  - ▶ **while** e **do - while**: basta fazer a condição sempre ser verdadeira.
- Há casos em que problemas podem ser resolvidos usando loop infinito.

# Loop infinito

- **Exemplo:** escreva um programa que leia números naturais do usuário, até que ele digite -1.

## Números

```
int i, n;  
  
i = 1;  
  
do {  
    printf("Digite um valor: "),  
    scanf("%d", &n);  
  
    if (n == -1)  
        break;  
  
    i++;  
} while (i > 1);
```



## Imprimindo os números inteiros de 80 a 20

```
int i;  
  
for (i = 80; i >= 20; i--) {  
    printf("i ");  
}
```

# Imprimindo os n primeiros números inteiros

```
int i, n;  
  
scanf("%d", &n);  
  
i = 1;  
  
while (i <= n) {  
    printf("i ");  
    i++;  
}
```

## Imprimindo as potências de x de 0 a 9

```
int i, x, pot;

scanf("%d", &x);

i = 0;

pot = 1;

do {
    printf("%d ^ %d = %d\n", x, i, pot),
    pot = pot * x;
    i++;
} while(i <= 9);
```