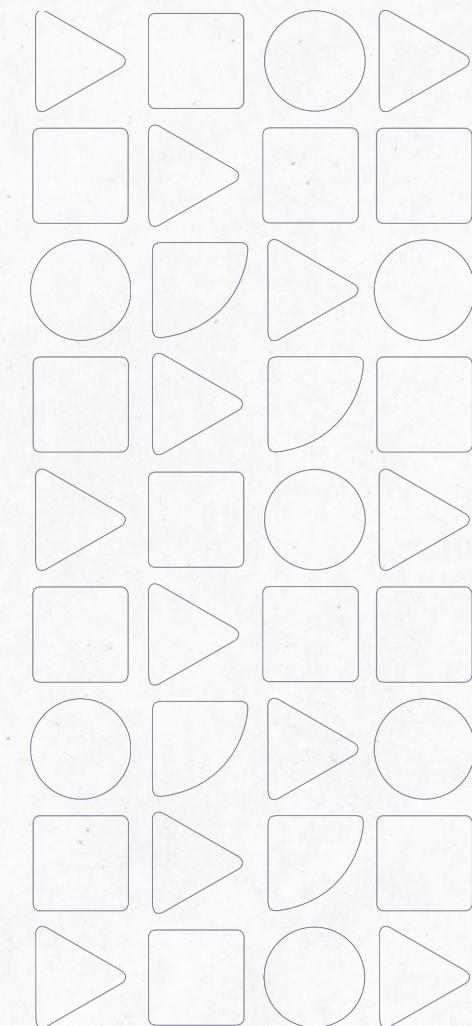


Estruturas de controle: repetição

Disciplina: Desenvolvimento de Aplicações



Conteúdos:

Estruturas de controle: repetição.

Habilidade(s):

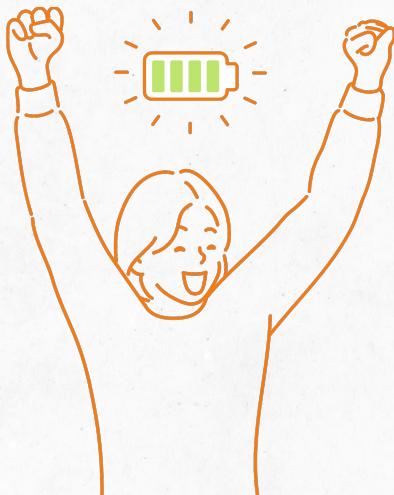
Bloco 1

Abertura

Qual é o seu *mood* de hoje?

Rodando mais que um comando de *do while* infinito.

Tô mais para um *for* que só é executado uma vez.



Estruturas de repetição

As estruturas de repetição **permitem que um conjunto de instruções seja executado repetidamente** enquanto uma condição específica estiver sendo atendida.

São muito úteis quando precisamos realizar tarefas repetitivas sem ter que escrever o mesmo código repetidamente.



Tipos das estruturas de repetição

Laços contados

Sabe-se (ou o usuário sabe) exatamente quantas vezes o comando será executado.

Exemplo: comando *for*.

Laços condicionais

Inicialmente, não se sabe o número de vezes que os comandos serão repetidos. Depende de uma condição de parada.

Exemplo: comandos *while* e *do while*.

Tipos das estruturas de repetição

For

Permite controlar de forma clara e concisa as iterações e o incremento/decremento.

No entanto, pode apresentar erros de inicialização ou incremento inadequados, que podem causar *loops* infinitos ou não executar o código.

While e do while

Tem a flexibilidade para executar o bloco de código até que a condição seja falsa.

Mas é possível cair em *loops* infinitos se a condição nunca se tornar falsa.

Importância e problemas

Vantagens

- Automatização de tarefas repetitivas;
- Redução de código repetido;
- Controle do fluxo de execução.

Desvantagens

- *Loops* infinitos se a condição não for corretamente gerenciada;
- Erros de inicialização ou incremento podem levar a resultados indesejados;
- Pode causar complexidade em código mal estruturado.

Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

15 minutos

Pesquisem sobre uma aplicação do *for* ou *do while*. Entendam como essas estruturas atuam na aplicação que vocês encontraram.

Segundo momento

10 minutos

Comentem com a turma sobre os resultados das suas pesquisas.



Bloco 2

For

A estrutura de repetição *for* é uma maneira eficaz de **executar um bloco de código várias vezes**, com controle preciso sobre o número de iterações.

Útil quando sabemos exatamente quantas vezes queremos repetir um bloco de código.



Comando *for*

```
java  
  
for (inicialização; condição; incremento/decremento) {  
    // Código a ser repetido  
}
```

Por exemplo: int i = 0

O primeiro parâmetro é para
inicialização da variável
conhecida como controladora.



```
for(i = 1; i<=10; i=i+1) {  
    printf("%d", i);  
}
```

*sintaxe da linguagem C

Comando *for*

```
java  
  
for (inicialização; condição; incremento/decremento) {  
    // Código a ser repetido  
}
```

Por exemplo: $i > 5$

O segundo parâmetro é para a condição, que será testada em todo início de ciclo.

```
for(i = 1; i<=10; i=i+1) {  
    printf("%d", i);  
}
```

*sintaxe da linguagem C

Comando *for*

java

```
for (inicialização; condição; incremento/decremento) {  
    // Código a ser repetido  
}
```

Por exemplo: $i = i + 1$

O terceiro parâmetro é para a incremento, que modificará a variável controladora.

```
for(i = 1; i<=10; i=i+1) {  
    printf("%d", i);  
}
```

*sintaxe da linguagem C

Exemplo: contagem crescente

java

```
for (int i = 1; i <= 5; i++) {  
    System.out.println(i);  
}
```

Inicialização: int i = 1

Condição: i <= 5

Incremento: i++

Exemplo: tabuada

java

```
int numero = 5;

for (int i = 1; i <= 10; i++) {
    int resultado = numero * i;
    System.out.println(numero + " x " + i + " = " + resultado);
}
```

Boas práticas e cuidados

- Evite inicializações ou incrementos complexos;
- Garanta que a condição seja eventualmente falsa;
- Mantenha o código dentro do escopo da estrutura *for*.



Vamos praticar?

Pegue uma folha para realizar a atividade

15 minutos

Criem um mapa mental sobre as estruturas de repetição *for* e como utilizá-las.
Lembre de detalhá-lo para lhe auxiliar no momento de criar um código.



Bloco 3

Fofoca do bem

Contem para a turma, em forma de fofoca, um pouco sobre a estrutura de repetição *for*.



Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

30 minutos

Criem um pequeno programa utilizando a estrutura condicional *for*.
Podem utilizar os mapas mentais já construídos para ajudar na atividade.

Segundo momento

10 minutos

Mostrem o código para a turma.



Bloco 4

While e do while

Permitem executar um bloco de código repetidamente **enquanto uma condição for verdadeira**.

Úteis quando o número de iterações não é conhecido antecipadamente ou quando queremos garantir que o bloco seja executado pelo menos uma vez.



Comando *while*

```
java
```

```
while (condição) {  
    // Código a ser repetido  
}
```

Útil quando a quantidade de iterações não é conhecida antecipadamente.



Comando do *while*

```
java  
  
do {  
    // Código a ser repetido  
} while (condição);
```

É semelhante ao *while*, mas garante que o bloco de código seja executado pelo menos uma vez.

Exemplo da estrutura **while**

```
java

int numero = 2;
while (numero <= 10) {
    System.out.println(numero);
    numero += 2;
}
```

- O *loop* continuará enquanto ‘numero <= 10’;
- O número é incrementado em 2 a cada iteração.

Exemplo da estrutura *do while*

```
java
import java.util.Scanner;

Scanner scanner = new Scanner(System.in);
int numero;
do {
    System.out.println("Digite um número (negativo para sair): ");
    numero = scanner.nextInt();
} while (numero >= 0);
scanner.close();
```

- O bloco é executado pelo menos uma vez;
- O *loop* continuará enquanto ‘numero >= 0’.

Boas práticas e cuidados

- Garanta que a condição seja eventualmente falsa para evitar *loops* infinitos;
- Tenha cuidado com o escopo das variáveis usadas na condição.



Vamos praticar?

Pegue uma folha para realizar a atividade

15 minutos

Criem um mapa mental sobre as estruturas de repetição *while* e *do while* e como utilizá-las. Lembre de detalhá-lo para lhe auxiliar no momento de criar um código.



Bloco 5

O que vem à sua mente ao ouvir
falar sobre a estrutura *while*?



Vamos praticar?

Dividam-se em grupos para realizar a atividade.

Primeiro momento

30 minutos

Criem um pequeno programa utilizando a estrutura condicional *while/do while*. Podem utilizar os mapas mentais já construídos para ajudar na atividade.

Segundo momento

10 minutos

Mostrem o código para a turma.



Bloco 6

Mas...

o que é são estruturas de repetição?



Comparando

For

Ideal quando se sabe o número de iterações.

VS

While/do while

Repete a depender da condição que o define.



Caso 1

Situação: calcular a média de notas de alunos em um vetor.

Caso 2

Situação: continuar solicitando uma senha até que o usuário forneça uma senha válida.

Caso 3

Situação: exibir os primeiros dez itens de uma lista de produtos.

Discussões e reflexões





Caso 1

For: calcular a média de notas de alunos em um vetor.

Caso 2

While: continuar solicitando uma senha até que o usuário forneça uma senha válida.

Caso 3

For: exibir os primeiros dez itens de uma lista de produtos.



Caso 4

Situação: manter um jogo em execução enquanto a vida do personagem for maior que zero.

Caso 5

Situação: receber mensagens de um servidor enquanto a conexão estiver ativa.

Caso 6

Situação: iterar sobre os elementos de uma lista para encontrar o maior valor.

Discussões e reflexões





Caso 1

While: manter um jogo em execução enquanto a vida do personagem for maior que zero.

Caso 2

While: receber mensagens de um servidor enquanto a conexão estiver ativa.

Caso 3

For: iterar sobre os elementos de uma lista para encontrar o maior valor.

E aí, foi difícil? Compartilha com a turma!

1

Você sentiu alguma dificuldade em entender as estruturas de repetição?

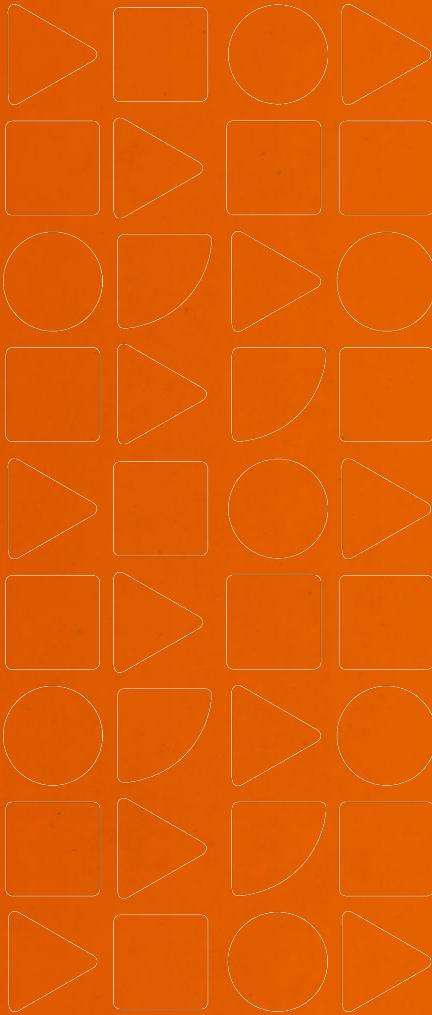
2

Você sentiu alguma dificuldade na hora de programar?

3

Agora, você sabe quando usar *for* e *while*?





Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Desenvolvimento de Aplicações*. 2023.