

Programação orientada a objetos

Prof. Ricardo Frohlich da Silva

Laço de repetições

- Os laços de repetição são estruturas de controle que permitem executar um bloco de código várias vezes.
- Eles são muito úteis quando se deseja realizar uma mesma tarefa repetidamente.
- Em Java, existem três tipos de laços de repetição:
 - For;
 - While e;
 - do... while.

Retomando sobre leitura pelo teclado

- Antes de apresentar os tipos de laços, vamos ver como fazer a leitura de dados pelo teclado utilizando a classe Scanner.
- Para isso, precisamos importar a classe `java.util.Scanner` e criar um objeto da classe Scanner.
- Em seguida, podemos utilizar os métodos da classe Scanner para fazer a leitura dos dados.

Retomando sobre leitura pelo teclado

```
1 package intro;
2 import java.util.Scanner;
3
4 public class LeituraTeclado {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.print("Digite um número inteiro: ");
9         int numInt = sc.nextInt();
10
11         System.out.print("Digite um número real: ");
12         double numReal = sc.nextDouble();
13
14         System.out.print("Digite uma palavra: ");
15         String palavra = sc.next();
16
17         System.out.println("O número inteiro digitado foi: " + numInt);
18         System.out.println("O número real digitado foi: " + numReal);
19         System.out.println("A palavra digitada foi: " + palavra);
20
21         sc.close();
22     }
23 }
```

Retomando sobre leitura pelo teclado

- Note que utilizamos os métodos `nextInt()`, `nextDouble()` e `next()` para fazer a leitura de um número inteiro, um número real e uma palavra, respectivamente.
- Ao final, fechamos o objeto `Scanner` utilizando o método `close()`.

For

- O laço For é utilizado quando se sabe exatamente quantas vezes o bloco de código deve ser repetido.
 - A estrutura do laço For é a seguinte:
 - `for (inicialização; condição; incremento) {`
 - `// bloco de código a ser repetido`
 - `}`
 - Inicialização: É a expressão que é executada antes do início do laço. Geralmente, é utilizada para inicializar uma variável contadora.
 - Condição: É a expressão que é avaliada a cada iteração do laço. Se a condição for verdadeira, o bloco de código é executado. Se for falsa, o laço é encerrado.
 - Incremento: É a expressão que é executada após cada iteração do laço. Geralmente, é utilizada para incrementar a variável contadora

For

- Neste exemplo, o bloco de código será executado 5 vezes, com os valores de i variando de 0 a 4.

```
for (int i = 0; i < 5; i++) {  
  
    System.out.println(i);  
  
}
```

While

- O laço While é utilizado quando não se sabe exatamente quantas vezes o bloco de código deve ser repetido, mas se sabe a condição de parada. A estrutura do laço While é a seguinte:
- `while (condição) {`
- `// bloco de código a ser repetido`
- `}`
- Condição: É a expressão que é avaliada a cada iteração do laço. Se a condição for verdadeira, o bloco de código é executado. Se for falsa, o laço é encerrado.

While

- Neste exemplo, o bloco de código será executado enquanto a variável *i* for menor que 5. A cada iteração, o valor de *i* é impresso na tela e a variável é incrementada.

```
4 public class TestaLaco {  
5     public static void main(String[] args) {  
6         int i = 0;  
7  
8         while (i < 5) {  
9             System.out.println(i);  
10            i++;  
11        }  
12    }  
13 }  
14
```

Do... While

- O laço do... while é semelhante ao laço While, porém, a condição é avaliada após a primeira execução do bloco de código.
- Isso significa que o bloco de código será executado pelo menos uma vez.
- A estrutura do laço do... while é a seguinte:
- do {
- // bloco de código a ser repetido
- } while (condição);
- Condição: É a expressão que é avaliada após cada iteração do laço. Se a condição for verdadeira, o bloco de código é executado novamente. Se for falsa, o laço é encerrado.

Do... While

- Neste exemplo imprimirá o valor de *i* de 0 a 4, uma vez que a condição ($i < 5$) será avaliada como verdadeira até que o contador alcance o valor de 5, momento em que a condição se tornará falsa e o laço terminará.

```
4 public class TestaLaco {  
5     public static void main(String[] args) {  
6         int i = 0;  
7  
8         do {  
9             System.out.println(i);  
10            i++;  
11        } while (i < 5);  
12    }  
13 }
```

Exercícios

- 1 - Escreva um programa Java que use um laço imprimir os números de 1 a 10.
- 2 - Escreva um programa Java que use um laço para imprimir os números pares de 2 a 20.
- 3 - Escreva um programa Java que use um laço para imprimir a tabuada do 7.
- 4 - Escreva um programa Java que use um laço para somar os números de 1 a 100.
- 5 - Escreva um programa Java que use um laço para calcular o fatorial de um número inteiro.

Exercícios

- 6- Escreva um programa Java que use um laço para ler uma sequência de números inteiros do usuário até que o número 0 seja lido.
- 7 - Escreva um programa Java que use um laço para ler a nota de vários alunos até que seja lida a nota -1.
- 8 - Escreva um programa Java que use um laço para ler vários nomes de cidades do usuário até que o nome "São Paulo" seja lido.
- 9 - Escreva um programa Java que use um laço para imprimir os números primos de 1 a 50.
- 10 - Escreva um programa Java que use um para calcular a média de vários números reais lidos do usuário até que seja lido o número 0.

Vetores e Matrizes

- Vetores e matrizes são estruturas de dados em Java que permitem armazenar e manipular um conjunto de valores de um mesmo tipo.

Vetores e Matrizes

- Um vetor é uma coleção unidimensional de elementos do mesmo tipo.
- Cada elemento é acessado através de um índice que começa em 0.
- A sintaxe para declarar um vetor em Java é a seguinte:
 - `tipo[] nomeDoVetor = new tipo[tamanho];`

Vetores

- Exemplo de uso de vetor em Java:

- ```
4 public class TestaVetorMatriz {
5 public static void main(String[] args) {
6 int[] numeros = new int[5];
7 numeros[0] = 10;
8 numeros[1] = 20;
9 numeros[2] = 30;
10 numeros[3] = 40;
11 numeros[4] = 50;
12 for (int i = 0; i < numeros.length; i++) {
13 System.out.println("O elemento de índice " + i + " é " + numeros[i]);
14 }
15 }
16 }
```



# Matrizes

---

- Já uma matriz é uma coleção bidimensional de elementos do mesmo tipo.
- Cada elemento é acessado através de dois índices que começam em 0.
- A sintaxe para declarar uma matriz em Java é a seguinte:
  - `int[][] matriz = new int[3][3];`

# Matrizes

---

- Exemplo:

```
4 public class TestaVetorMatriz {
5 public static void main(String[] args) {
6 int[][] matriz = new int[3][3];
7
8 matriz[0][0] = 1;
9 matriz[0][1] = 2;
10 matriz[0][2] = 3;
11 matriz[1][0] = 4;
12 matriz[1][1] = 5;
13 matriz[1][2] = 6;
14 matriz[2][0] = 7;
15 matriz[2][1] = 8;
16 matriz[2][2] = 9;
17
18 for (int i = 0; i < matriz.length; i++) {
19 for (int j = 0; j < matriz[i].length; j++) {
20 System.out.print(matriz[i][j] + " ");
21 }
22 System.out.println();
23 }
24 }
25 }
```

# Exercícios

---

- 1 - Escreva um programa Java que use um vetor para ler 5 nomes de pessoas e depois os imprima na tela.
- 2 - Escreva um programa Java que use uma matriz para ler os valores de uma tabela de multiplicação (tabuada) de 10x10 e depois a imprima na tela.
- 3 - Escreva um programa Java que use um vetor para ler as notas de 5 alunos e depois calcule e imprima a média das notas.
- 4 - Escreva um programa Java que use uma matriz para ler as notas de 3 alunos em 4 disciplinas e depois calcule e imprima a média de cada disciplina.
- 5 - Escreva um programa Java que use um vetor para ler 10 números inteiros lidos do usuário e depois imprima o maior e o menor valor.

# Exercícios

---

- 6 - Escreva um programa Java que use um vetor para ler os nomes de 5 cidades e outro vetor para armazenar as suas respectivas populações e depois imprima na tela o nome da cidade mais populosa.
- 7 - Escreva um programa Java que use uma matriz para armazenar as notas de 3 alunos em 4 disciplinas e depois imprima na tela a nota mais alta e a nota mais baixa em cada disciplina.
- 8 - Escreva um programa Java que use um vetor para armazenar 10 números inteiros lidos do usuário e depois imprima na tela quantos números são pares e quantos são ímpares.