


INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DO RIO GRANDE DO NORTE
CAMPUS JOÃO CÂMARA

COMANDOS DE REPETIÇÃO

Introdução

2

```
public class Papagaio {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        String frase = sc.nextLine();  
  
        System.out.println(frase);  
        System.out.println(frase);  
        System.out.println("CURUPACO");  
  
    }  
  
}
```



E se eu
quisesse repetir
a frase 1000x
??

Antes de começar!

3

- **Contadores:** utilizados para realizar o controle dos laços de repetição. Normalmente são incrementados ou decrementados.
 - $\text{cont} = \text{cont} + 1$
- **Acumuladores:** utilizados para realizar somatórios e acumular o valor total. Normalmente recebe o próprio valor incrementado de outros valores.
 - $\text{soma} = \text{soma} + \text{cont}$

Comandos de repetição

4

- São utilizados quando é necessário repetir várias vezes um determinado trecho de código.
- A quantidade de repetições pode ser fixa ou depender de uma condição (**expressão lógica**).
- Os comandos de repetição também são conhecidos como laços de repetição.
- Podem ser de três tipos:
 - Repetição com testes no início;
 - Repetição com testes no fim;
 - Repetição com variável de controle.

Repetição com teste no início

5

- Esta estrutura repete uma sequência de comandos enquanto uma determinada condição (**expressão lógica**) for **VERDADEIRA**.
- A expressão lógica fica localizada no início da estrutura do comando, ou seja, se o valor da expressão for falsa, não executa o bloco **NENHUMA** vez.

```
while (<expressão lógica>) {  
    <sequencia de comandos>  
}
```

Repetição com teste no início

6

- Voltando ao algoritmo do papagaio...
- Como faríamos para escrever a frase 1000x ??

Repetição com teste no início

7

```
public class Papagaio {  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        String frase = sc.nextLine();  
        int cont = 1;  
  
        while (cont <= 1000) {  
  
            System.out.println(frase);  
            System.out.println(frase);  
  
            cont = cont + 1;  
  
        }  
        System.out.println("CURUPACO");  
    }  
}
```

Repetição com teste no fim

8

```
do {  
    <sequencia de comandos>  
} while <expressão lógica>;
```

- **do**: indica o início do laço de repetição.
- **while <expressão lógica>**: indica o fim da sequência de comandos a serem repetidos.
- Como a condição é apenas no fim da estrutura, a sequência de comandos é executada pelo menos uma vez.

Repetição com teste no fim

9

```
do {  
    <sequencia de comandos>  
while <expressão lógica>;
```

- A cada momento que a execução do programa chega no comando **while** <expressão lógica>, a expressão lógica é avaliada.
 - Se ela for **VERDADEIRA**, a execução retorna para o início do bloco;
 - Caso contrário, a execução continua com o primeiro comando após o bloco.

Repetição com teste no fim

10

- Crie um algoritmo que escreva os números de 1 a 5.

```
public static void main(String[] args) {  
  
    int cont = 1;  
    do{  
        System.out.println(cont);  
        cont = cont +1;  
    }while (cont <= 5);  
  
    System.out.println("FIM!");  
  
}
```

Comparação

11

- **while;**

- Teste no Início;

- A sequência de comandos é repetida zero ou mais vezes;

- **do... while;**

- Teste no Fim;

- A sequência de comandos é repetida 1 ou mais vezes;

Repetição com variável de controle

12

- O número de repetições é conhecido.
- Uma variável de controle é utilizada para realizar as repetições.
- Fixamos os valores iniciais, finais e o passo (incremento) dessa variável.
- Estrutura “for”

```
for (<variável>; <limite>; <incremento>){  
    <sequencia de comandos>  
}
```

Repetição com variável de controle

13

<i><variável></i> <i>Ex: int cont = 0;</i>	É um contador (do tipo inteiro) que controla o número de repetições do laço. E também determina o valor inicial.
<i><limite></i> <i>Ex: cont < 100;</i>	Determina o valor máximo da variável.
<i>passo <incremento></i> <i>Ex: cont++;</i>	É opcional. Determina o incremento que será acrescentado à variável de controle. Quando omitida, o valor é 1.

Repetição com variável de controle

14

- Quando o programa alcança este ponto, é acrescentado a variável o valor *<incremento>* e comparado a *<limite>*. Se a variável for menor ou igual, a sequência de comandos será executada mais uma vez. Caso contrário, prossegue-se para o primeiro comando após o **fechamento do bloco for (}**).

Repetição com variável de controle

15

- Crie um algoritmo para escrever os número de 1 a 5.

```
public static void main(String[] args) {  
    for (int cont = 1; cont <= 5; cont++) {  
        System.out.println(cont);  
    }  
}
```

Exercício

16

- ❑ Crie um algoritmo que leia um número e escreva a sequência de zero até o número.
- ❑ Crie um algoritmo que leia um número e escreva a sequência decrescente até o zero.
- ❑ Crie um algoritmo que recebe 2 números e multiplica o num1 pelo num2 através de somas repetidas. (ex: 2 e 3 = $2 + 2 + 2$).