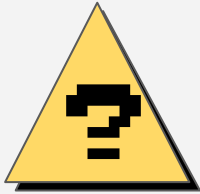




COMPETEC



COMPETEC



Aula anterior

Don't show this message again

Ok



J main.java ×

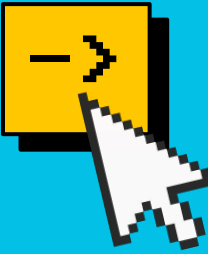
J main.java

1. Fazer um programa para ler um número inteiro e dizer se este número é par ou ímpar.



Aula n°02

Loops



Assuntos da aula

- >While
- >Do while
- >For

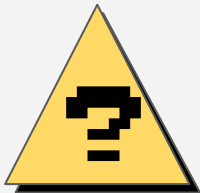
Ok

Cancel

Options<<

Loops são estruturas de repetição, que repetem uma parte do seu código enquanto uma condição estabelecida estiver sendo cumprida.

COMPETEC



while

Don't show this message again

Ok



```
while (condição) { //enquanto a condição for verdade
    executa;       //ele executa o que está dentro
}
```


while



```
int x = 0;
while (x < 10) {
    System.out.print(x + " ");
    x++;
}
```

```
//imprime: 0 1 2 3 4 5 6 7 8 9
```

Tome sempre cuidado para garantir que o seu loop se encerrará. A variável presente na condição deve sempre ser atualizada dentro do loop.

```
int x = 0;
```

```
while (x >= 0) {
```

```
    System.out.print(x + " ");
```

```
    x++;
```

```
}
```

```
//imprime: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 ...
```

J main.java X

J main.java

1. Crie um programa que calcule a soma de 1 até 100.
2. Escreva um programa que repita a leitura de uma senha até que ela seja válida. Para cada leitura de senha incorreta informada, imprima a mensagem "Senha Invalida". Quando a senha for informada corretamente deve ser impressa a mensagem "Acesso Permitido" e o programa encerrado.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

O *do while* possui um funcionamento muito semelhante ao *while*, porém ele garante que o código dentro do loop execute ao menos uma vez, independente da condição.

```
do {  
    executa;  
} while (condição);
```

do while



```
int x = 0;

do{

    System.out.print(x + " ");

    x++;

}while (x < 10);

//imprime: 0 1 2 3 4 5 6 7 8 9
```

```
int x = 10;

do{

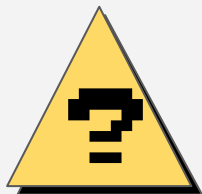
    System.out.print(x + " ");

    x++;

}while (x < 10);

//imprime: 10
```

COMPETEC



for

Don't show this message again

Ok



```
for (inicialização; condição; atualização){  
    executa;  
}
```

```
for (int i = 0; i < 10; i++){  
    System.out.print(i + " ");  
}
```

```
//imprime: 0 1 2 3 4 5 6 7 8 9
```

Exercício:

Conserte o código abaixo utilizando o *loop for*:

```
for (int i = 10; i <=0; i = i/2){  
    System.out.print(i + " ");  
}
```

//objetivo: imprimir "10 8 6 4 2"

Possibilidade

de

Resposta:

```
for (int i = 10; i >0; i = i-2){  
    System.out.print(i + " ");  
}
```

```
//imprime "10 8 6 4 2"
```

Assim como vimos no switch case, o *break* pode ser usado para sair de um loop.

```
for (int i = 0; i < 10; i++){  
    if (i == 5) break;  
    System.out.print(i + " ");  
}  
  
//imprime: 0 1 2 3 4
```

Já o *continue* irá ignorar o resto da iteração vigente e pular para a próxima.

```
for (int i = 0; i < 10; i++){  
    if (i == 5) continue;  
    System.out.print(i + " ");  
}  
  
//imprime: 0 1 2 3 4 6 7 8 9
```

J main.java X

J main.java

1. Crie um programa que solicite ao usuário um número inteiro e imprima a tabuada desse número de 1 a 10.
2. Leia um número N e imprima todos os números ímpares até chegar nele.

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

COMPETEC



Obrigado pela
atenção! :)

Don't show this message again

Ok

