

FIAP GRADUAÇÃO

Agenda

- Estruturas condicionais
- Estruturas de repetição

Estruturas condicionais

Estruturas condicionais definem que determinada parte do código vai ser executada ou não.

Vamos ver 2 formas de condicionais em Java:

IF/ELSE e Switch Case

Condições?

- Para criarmos estruturas condicionais precisamos de uma condição. No Java isso significa qualquer operação que resulte em um valor do tipo boolean.

Podemos utilizar operadores lógicos, relacionais ou qualquer método que retorne um boolean.

Lógico

Operador	Função
&&	E
	Ou
!	Não

Relacional

Operador	Função
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a

Método

```
if (operacao.equals("+")) {
    System.out.println("SOMA");
}
```

IF

If (se) é a condicional mais básica e está presente em praticamente todas as linguagens de programação.

Sintaxe

```
if( <Condição> ) {  
    <Instruções>  
}
```

Exemplo:

```
int idade = 21;  
if(idade >= 18){  
    System.out.println("Maior de idade");  
}
```

IF

If (se) é a condicional mais básica e está presente em praticamente todas as linguagens de programação.

Sintaxe

```
if( <Condição> ) {  
    <Instruções>  
}
```

Exemplo:

```
int idade = 21;  
if(idade >= 18){  
    System.out.println("Maior de idade");  
}
```

IF / ELSE IF / ELSE

Se tivermos condições mais complexas, podemos utilizar as estruturas else if e else. É importante saber que apenas a primeira condição satisfatória será executada.

Sintaxe

if(<Condição>) { **Exemplo:**

```

    <Instruções>
} else if ( <Condição> ) {
    <Instruções>
} else {
    <Instruções>
}

```

```

int idade = 21;
if(idade < 18){
    System.out.println("Criança ou Adolescente");
} else if (idade < 65) {
    System.out.println("Adulto");
} else {
    System.out.println("Idoso");
}

```

Switch Case

Se tivermos múltiplas condições, podemos utilizar o switch-case

Exemplo:

Sintaxe

```
switch ( <Condição> ) {  
    case <valor> :  
        <Instruções>  
        break;  
    case <outro valor> :  
        <Instruções>  
        break;  
}
```

```
int diaDaSemana = 0;  
switch (diaDaSemana) {  
    case 0:  
        System.out.println("Domingo");  
        break;  
    case 1:  
        System.out.println("Segunda");  
        break;  
    case 2:  
        System.out.println("Terça");  
        break;  
    case 3:  
        System.out.println("Quarta");  
        break;  
    case 4:  
        System.out.println("Quinta");  
        break;  
    case 5:  
        System.out.println("Sexta");  
        break;  
    case 6:  
        System.out.println("Sábado");  
        break;  
    default:  
        System.out.println("Dia desconhecido");  
}
```


Switch Case

Se tivermos múltiplas condições, podemos utilizar o switch-case

Exemplo:

Sintaxe

```
switch ( <Condição> ) {  
    case <valor> :  
        <Instruções>  
        break;  
    case <outro valor> :  
        <Instruções>  
        break;  
}
```

```
int diaDaSemana = 0;  
switch (diaDaSemana) {  
    case 0:  
        System.out.println("Domingo");  
        break;  
    case 1:  
        System.out.println("Segunda");  
        break;  
    case 2:  
        System.out.println("Terça");  
        break;  
    case 3:  
        System.out.println("Quarta");  
        break;  
    case 4:  
        System.out.println("Quinta");  
        break;  
    case 5:  
        System.out.println("Sexta");  
        break;  
    case 6:  
        System.out.println("Sábado");  
        break;  
    default:  
        System.out.println("Dia desconhecido");  
}
```

Exercício

1 - Crie um programa para que o usuário possa informar o nome, altura e o peso de duas pessoas. Imprima o nome do mais pesado e o nome do mais alto.

2 - Crie um programa para cadastrar um eleitor (nome e idade). Em seguida informar se o usuário cadastrado é:

- Não eleitor (menor de 16 anos)
- Eleitor obrigatório (entre 18 e 65 anos)
- Eleitor facultativo (16, 17 ou mais de 65 anos)

Desafio #1: Crie um fluxo para que o eleitor cadastrado possa votar. Esse fluxo deve respeitar as regras de obrigatoriedade. Não eleitores não podem votar, obrigatórios devem votar e para os facultativos deve ser exibida uma opção de voto.

Estruturas de repetição

Quando precisamos que alguma tarefa seja executada múltiplas vezes podemos utilizar as estruturas de repetição.

Vamos ver 3 formas de fazer isso em java:

FOR

WHILE

DO/WHILE

FOR

O “for” tem uma estrutura concisa e pode ser declarado em uma única linha:

Sintaxe

```
for ( <inicializacao> ; <condição>; <incremento>) {  
    <Instruções>  
}
```

Exemplo

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

FOR EACH

Desde o Java 5, podemos declarar o “for” desta forma quando trabalhamos com collections. Este for vai iterar por todos os elementos.

Sintaxe

```
for (    <tipo>    <nome da variavel>;    <Collection>) {  
    <Instruções>  
}
```

Exemplo

```
List<String> listaDePalavras = new ArrayList<>();  
for (String palavra : listaDePalavras) {  
    System.out.println("A palavra é " + palavra);  
}
```

WHILE

Executa determinada tarefa enquanto a condição for verdadeira:

Sintaxe

```
while (                <condição>                ) {  
    <Instruções>  
}
```

Exemplo

```
int numero = 1;  
while (numero < 10) {  
    System.out.println("Numero atual " + numero);  
    numero = numero + numero;  
}
```

DO / WHILE

Funciona de maneira similar ao WHILE, porém executa a instrução pelo menos uma vez:

Sintaxe

```
do  
    {  
        <Instruções>  
    } while ( <condição>);
```

Exemplo

```
int numero = 100;  
do {  
    System.out.println("Numero atual " + numero);  
    numero = numero + numero;  
} while (numero < 10);
```

Exercícios

- 1 - Imprima todos os números de 0 a 100 utilizando for, while e do/while.
- 2 - Crie um programa para imprimir a seguinte saída utilizando 2 for:

```
*  
**  
***  
****  
*****
```

Desafio #1: Imprima primeira letra do seu nome com “*” utilizando for.

Dica: A instrução `System.out.println` pula uma linha após o final da palavra.

Para imprimir na mesma linha utilize `System.out.print`

OBRIGADO

FIAP

Copyright © 2020 | Professor Fabio Tadashi Miyasato
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.