

Estruturas de decisão (*if - else*)

- Composto pelas palavras reserv. *if* e *else*
- As instruções seguintes podem estar em um bloco ou em apenas uma linha
- A condição sempre vem dentro de parênteses “()” e seu resultado deve um boolean
- Podem ser aninhadas, misturando blocos de várias linhas ou linhas individuais

Estruturas de decisão (*if - else*)

- Qual o resultado? (x = 35 e y = 10)

```
if (x > 35)
    System.out.println("primeiro");
else{
    if(y < 10)
        System.out.println("segundo");
    if (y == 10)
        System.out.println("terceiro");
    else{
        if(!false) {
            System.out.println("quarto");
        } else{
            System.out.println("quinto");
        }
    }
}
```

Estrutura *switch*

- Composto pelas palavras reservadas: ***switch***, ***case***, ***default*** e ***break***
- Quando a condição é atendida por um ***case***, então pode-se afirmar que todas as instruções serão executadas até que seja encontrado um ***break***
- Caso nenhuma condição seja atendida, é executado o ***default***, se esse existir na estrut.

Estrutura *switch*

- Não é indicado para definir intervalos de valores (nesse caso *if - else* é melhor!)
- Pode ser usado com os tipos primitivos inteiros, char e também tipos enumerados

Estrutura *switch*

- Qual o resultado?

```
char c = '\u0065';  
switch (c) {  
    case 0145:  
        System.out.println("primeiro");  
    case 1:  
        System.out.println("segundo");  
        break;  
    default:  
        System.out.println("terceiro");  
        break;  
}
```

Laço de repetição – *while*

- Executa um bloco de instruções enquanto a condição for ***verdadeira***
- Primeiro testa a condição, depois executa o bloco de instruções
- Um laço infinito será definido se a condição sempre for verdadeira
- Lembrem-se que ***while*** é uma palavra reservada

Laço de repetição – *while*

- Exemplo:

```
int x = 0;
while (x < 10) {
    System.out.println("O valor de x é: " + x);
    x++;
}
```

Laço de repetição – *do while*

- Executa um bloco de instruções, pelo menos uma vez, enquanto a condição for ***verdadeira***
- Primeiro executa as instruções e em seguida testa a condição
- Um laço infinito também será definido se a condição sempre for verdadeira
- A palavra ***do*** também é reservada

Laço de repetição – *do while*

- Exemplo:

```
do {  
    x++;  
    System.out.println("O valor de x é: " + x);  
} while (x < 10);
```

Laço de repetição – *for*

- A partir do Java 5.0, apresenta duas estruturas, sendo que o “segundo” ***for*** é comumente definido como *enhanced for*
- Seu uso é mais indicado quando já se sabe quantas vezes o laço será executado

Laço de repetição – *for*

- A declaração do laço de repetição ***for*** é dividida em 3 partes:
 - **Declaração e inicialização** de variáveis do mesmo tipo (*executado apenas uma vez, antes das instruções*)
 - **Expressão condicional** (teste booleano executado antes de cada iteração)
 - **Expressão de iteração** (executado após cada iteração)

Laço de repetição – *for*

- Exemplo:

```
for (int x = 0; x < 10; x++) {  
    System.out.println("O valor de x é: " + x);  
}
```

Laço de repetição – *for*

- Nenhuma das 3 partes do *for* são obrigatórias!
- Caso as partes **Declaração e inicialização** e **Expressão de iteração** não estejam definidas, o *for* vai se comportar como um *while*
- O escopo das variáveis declaradas no *for* é apenas o seu bloco

Laço de repetição – *enhanced for*

- Simplifica a iteração sobre arrays e coleções
- Exemplo:

```
int[] arrayInt = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
for (int x : arrayInt) {  
    System.out.println("O valor de x é: " + x);  
}
```

Uso de ***break*** e ***continue***

- Ambos são palavras reservadas
- O ***break*** interrompe a iteração por completo
- O ***continue*** pára a iteração atual e vai para a próxima, caso exista
- Podem ser usados sem label (*unlabeled*) ou com label
- Um label é definido por um identificador seguido de dois pontos (Ex: laco_01:)

Uso de *break* e *continue*

- Exemplo de break com label:

```
boolean isTrue = true;
loop_externo:
    for(int i=0; i<5; i++) {
        while (isTrue) {
            System.out.println("Olá");
            break loop_externo;
        }
        System.out.println("Loop externo.");
    }
System.out.println("Saindo do programa.");
```


Uso de *break* e *continue*

- Exemplo de continue com label:

```
loop_externo:
    for (int i=0; i<5; i++) {
        for (int j=0; j<5; j++) {
            System.out.println("Olá.");
            continue loop_externo;
        }
        System.out.println("Loop externo.");
    }
    System.out.println("Saindo do programa.");
```

Maneiras de encerrar um laço de repetição

- **break:** a execução vai imediatamente para a primeira instrução após o loop
- **return:** a execução vai para o método chamador
- **System.exit():** encerra a execução do programa. Desativa a VM.