Fundamentos de programação

Iteração

Uso dos comandos break/continue

Edson Moreno

edson.moreno@pucrs.br

http://www.inf.pucrs.br/~emoreno

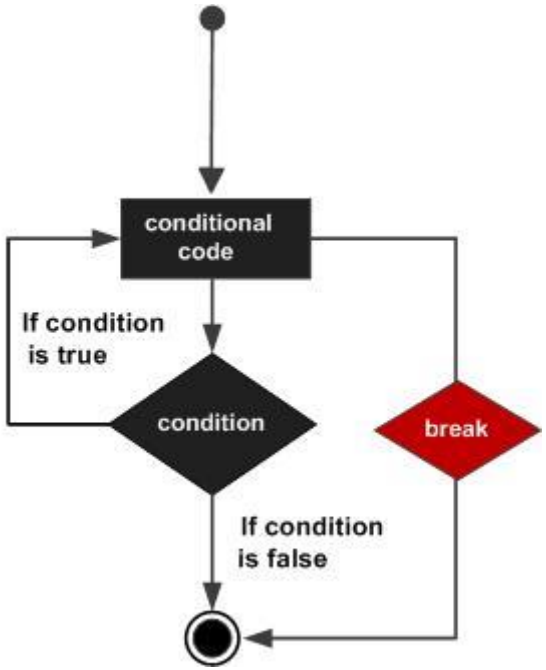
Comandos de controle de fluxo

- Comando de iteração
  - Possuem uma condição lógica para avaliar a repetição

- Blocos de código internos de um comando de iteração
  - Podem incluir comandos puramente sequenciais
  - Comandos de decisão
    - If / switches
  - Comandos de iteração
    - For / while / do while
  - Comandos especiais de controle de fluxo
    - Break / continue

O comando break

- O comando leva ao fim da execução do laço
  - Os comandos logo ao final do laço são executados
  - Equivale a avaliação falsa da condição de repetição



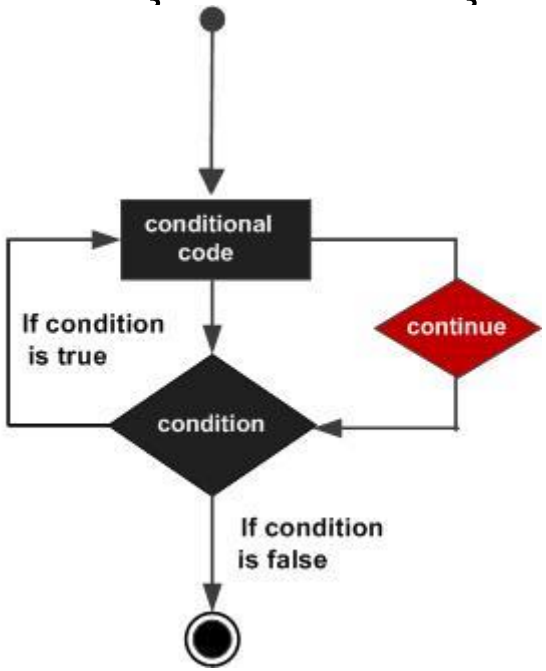
O comando break

- Exemplo de uso

```
public class BreakDemo {
  public static void main(String[] args)
  {
    for (int i = 1; i <= 10; i++)
    {
      if (i == 5)
      {
        break; // terminate loop if i is 5
      }
      System.out.print(i + " ");
    }
    System.out.println("Loop is over.");
  }
}
```

O comando continue

- O comando leva a continuação da iteração
  - Os comandos posteriores ao comando são ignorados
  - O comando de iteração é relançado



O comando continue

- Exemplo de uso

```
public class ContinueDemo {
  public static void main(String[] args)
  {
    for (int i = 1; i <= 10; i++)
    {
      if (i % 2 == 0)
      {
        continue; // skip next statement if i is even
      }
      System.out.println(i + " ");
    }
  }
}
```

Algoritmos comuns com laços

- Somatório
  - Inicialize *total* com 0
  - Pode-se usar o laço com sentinela
  - Acrescentar o valor em total

```
double total = 0;
while (in.hasNextDouble())
{
  double input =
in.nextDouble();
  total = total + input;
}
```

Algoritmos comuns com laços

- Produtório
  - Inicialize *produto* com 1
  - Multiplique o valor por *produto*

```
double produto = 1;
while (in.hasNextDouble())
{
  double input =
in.nextDouble();
  produto = produto * input;
}
```

Algoritmos comuns com laços

- Valor médio
  - Faça o somatório dos valores
  - Inicialize *count* com 0, incrementando-a a cada valor lido
  - Verificar o valor de *count* antes da divisão

```
double total = 0;
int count = 0;
while (in.hasNextDouble()) {
  double input = in.nextDouble();
  total = total + input;
  count++;
}

double average = 0;
if (count > 0)
  average = total / count;
```

Algoritmos comuns com laços

- Contagem de ocorrências
  - Inicialize *count* com 0
  - Use um laço *for*
  - Incremente o contador a cada ocorrência

```
int upperCaseLetters = 0;
for (int i = 0; i < str.length(); i++)
{
  char ch = str.charAt(i);
  if (Character.isUpperCase(ch))
{
    upperCaseLetters++;
  }
}
```

Algoritmos comuns com laços

- Encontrar a primeira ocorrência
  - Inicialize uma variável sentinela/booleana com *false*
  - Inicialize o contador de posição com 0 (primeiro caracter do *string)*
  - Use uma condição composta no laço

```
boolean found = false;
char ch;
int position = 0;
while (!found && position < str.length()) {
  ch = str.charAt(position);
  if (Character.isLowerCase(ch)) {
    found = true;
  }
  else { position++; }
}
```

Algoritmos comuns com laços

- Pesquisar até que uma ocorrência seja encontrada
  - Inicialize uma variável sentinela/booleana com *false*
  - Teste a variável sentinela no laço de while
    - Leia a entrada, e compare com o limite
      - Se a entrada está dentro do limite então altere a sentinela para *true*
      - Senão imprime a mensagem

```
boolean valid = false;
double input;
while (!valid) {
  System.out.print("Please enter a positive value < 100:
");
  input = in.nextDouble();
  if (0 < input && input < 100) { valid = true; }
  else { System.out.println("Invalid input."); }
}
```

Algoritmos comuns com laços

- Mínimo e máximo
  - Leia o primeiro valor: este é o maior (ou menor) valor de partida
  - Permanecer no laço enquanto tiver um valor
    - Leia o próximo valor
    - Comparar com o último valor armazenado
    - Se maior/menor, atualiza-se a variável

```
double largest = in.nextDouble();
while (in.hasNextDouble()) {
  double input = in.nextDouble();
  if (input > largest) {
    largest = input;
  }
}
```

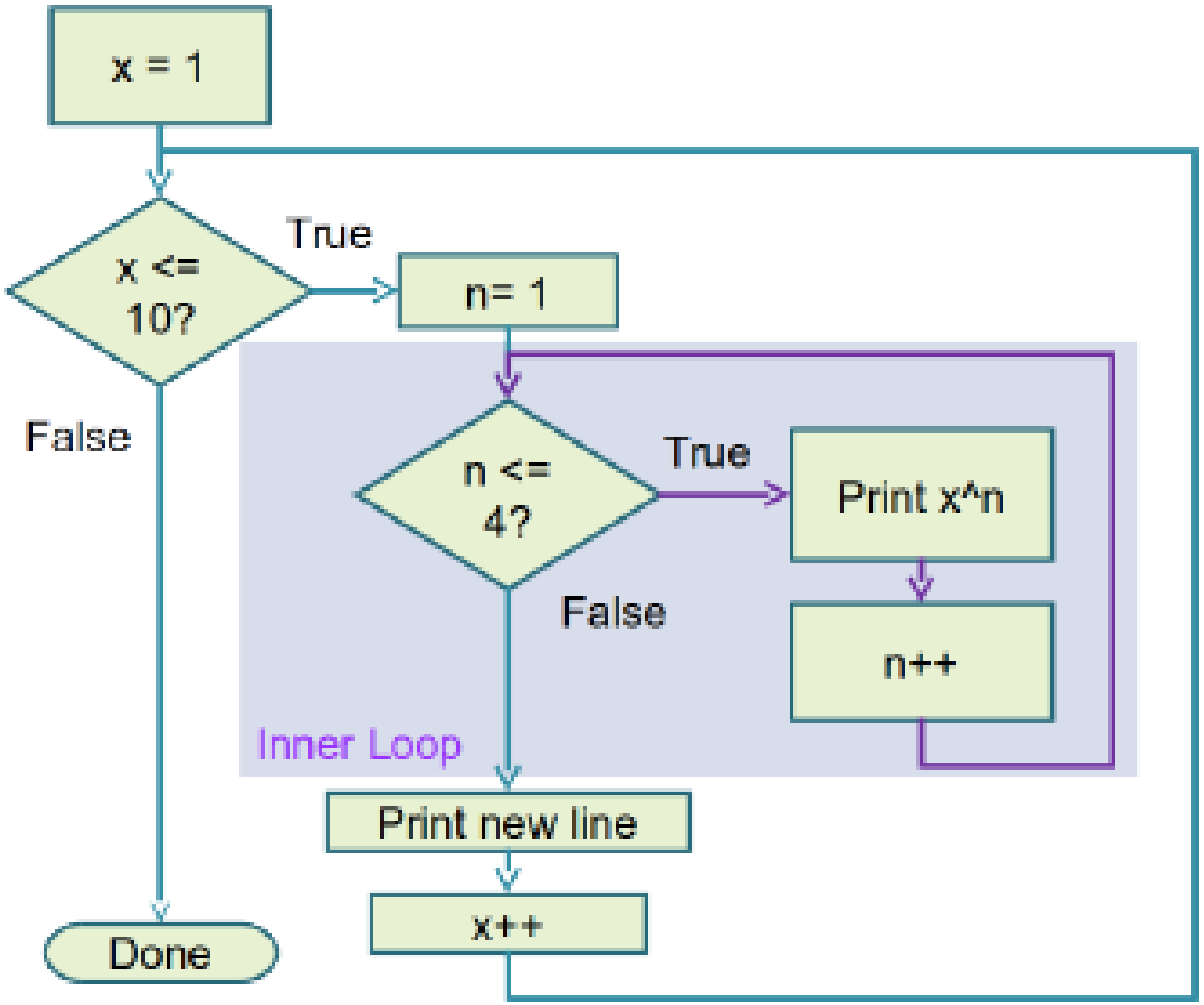
```
double smallest = in.nextDouble();
while (in.hasNextDouble()) {
  double input = in.nextDouble();
  if (input < smallest) {
    smallest = input;
  }
}
```

Laços aninhados

- Como você imprimiria uma tabela com linhas e colunas?
  - Imprima a primeira linha com o cabeçalho
    - Use um laço
  - Imprima o corpo da tabel
    - Quantas linhas?
    - Quantas colunas?
  - Faça um laço para as linh
    - Faça um laço para as colun

x^1	x^2	x^3	x^4
1	1	1	1
2	4	8	16
3	9	27	81
...	...	...	...
10	100	1000	10000

Laços aninhados



Laços aninhados

```
public class PowerTable {
  public static void main(String[] args)
{
    final int NMAX = 4;
    final double XMAX = 10;

    // Print table header
    for (int n = 1; n <= NMAX; n++) {
      System.out.printf("%10d", n);
    }
    System.out.println();

    for (int n = 1; n <= NMAX; n++) {
      System.out.printf("%10s", "x ");
    }
    System.out.println();
    // Print table body
    for (double x = 1; x <= XMAX; x++)
    // Print table row
      for (int n = 1; n <= NMAX; n++) {
        System.out.printf("%10.0f",
                                Math.pow(x
      }
      System.out.println();
    }
  }
}
```

Laços aninhados

Laço	Saída	Explicação

`for (i = 1; i <= 3 ; i++) {`
`  for (j = 1; j <= 4; j++) {`
`  System.out.print("*");`
`  }`
`  System.out.println();`
`}`
	****
	****
	****
		Imprime 3 linhas de 4 asteriscos cada.

`for (i = 1; i <= 4 ; i++) {`
`  for (j = 1; j <= 3; j++) {`
`  System.out.print("*");`
`  }`
`  System.out.println();`
`}`
	***
	***
	***
	***
		Imprime 4 linhas de 3 asteriscos cada.

`for (i = 1; i <= 4 ; i++) {`
`  for (j = 1; j <= i; j++) {`
`  System.out.print("*");`
`  }`
`  System.out.println();`
`}`
	*
	**
	***
	****
		Imprime 4 linhas com tamanhos 1, 2, 3 e 4.

Laços aninhados

	Saída	Explicação

`for (i = 1; i <= 3 ; i++) {`
`  for (j = 1; j <= 5; j++) {`
`  if (j % 2 == 0) {`
`  System.out.print("*");`
`  }`
`  else {`
`  System.out.print("-");`
`  }`
`  }`
`  System.out.println();`
`}`
	-*-*-
	-*-*-
	-*-*-
		Imprime asteriscos nas colunas pares e traços nas colunas ímpares.

`for (i = 1; i <= 3 ; i++) {`
`  for (j = 1; j <= 5; j++) {`
`  if (i % 2 == j % 2) {`
`  System.out.print("*");`
`  }`
`  else {`
`  System.out.print(" ");`
`  }`
`  }`
`  System.out.println();`
`}`
	* * *
	 * *
	* * *
		Imprime o padrão de um tabuleiro de damas.

Exercício

- Faça laços aninhados para imprimir os seguintes padrões

a)

```
*****
.****
..***
...**
....*
```

b)

```
*....
.*...
..*..
...*.
....*
```

c)

```
....*
...*.
..*..
.*...
*....
```

d)

```
*****
****.
***..
**...
*....
```

e)

```
....*
...**
..***
.****
*****
```