

Fatec Mococa

Linguagem de Programação

Estruturas de Repetição

Aplicações:

Somadores e contadores

Contagens com condição

Busca do maior/menor

Sandra Cristina Costa Prado

1º semestre de 2022

Conceitos Básicos de Programação

- Variáveis
- Comandos de entrada e saída de dados
- Expressões aritméticas e lógicas
- Estruturas condicionais (decisão)
- **Estruturas de repetição**

Laço de repetição

Um laço (ou loop) de repetição é uma estrutura de programação que repete uma sequência de instruções **enquanto uma condição for verdadeira**.

O Java possui três comandos para implementar repetições:

- 1) for
- 2) while
- 3) do while

Algoritmos

Java

Repetição controlada
por contador
(o teste é no início)

```
declare  
i: inteiro  
...  
para i ← 1 até 10 faça  
    escreval i  
fimpara  
...
```

```
for(int i = 1; i<=10; i++){  
    System.out.println(i);  
}
```

i++ é mesmo que
i = i + 1

Repetição controlada
por sentinela com
teste no início

```
declare  
i: inteiro  
...  
i ← 1  
enquanto (i <= 10) faça  
    escreval i  
    i ← i + 1  
fimenquanto  
...
```

```
int i = 1;  
while (i <= 10) {  
    System.out.print(i + " ");  
    i++;  
}
```

Repetição controlada
por sentinela com
teste no final

```
declare  
i: inteiro  
...  
i ← 1  
faça  
    escreval i  
    i ← i + 1  
enquanto (i <= 10)
```

```
i = 1;  
do {  
    System.out.print(i + " ");  
    i++;  
} while (i <= 10);
```

Tipos de laços de repetição

1) Laço controlado por **contador**

O número de vezes que o conjunto de comandos será repetido é conhecido antes de iniciar a repetição.

Também chamado de definido, determinado ou contado.

Como este laço é encerrado?

O laço se encerra quando a variável que controla o laço, chamada de contador do laço, ultrapassar o valor máximo pré-definido.

2) Laço controlado por **sentinela**

O número de vezes que o conjunto de comandos será repetido é arbitrário.

É chamado de repetição indefinida e indeterminada.

Como este laço é encerrado?

O laço se encerra quando a variável que controla o laço recebe o valor sentinela, também chamado de flag. Comumente, é o usuário que digita este valor.

4.8 Formulando algoritmos: repetição controlada por contador

Para ilustrar como os algoritmos são desenvolvidos, modificaremos a classe GradeBook do Capítulo 3 a fim de resolver duas variações de um problema que calcula a média das notas dos alunos. Considere a seguinte declaração do problema:

Uma classe de dez alunos submeteu-se a um questionário. As notas (inteiras no intervalo 0 a 100) para esse questionário estão disponíveis. Determine a média da classe no questionário.

A média de classe é igual à soma das notas divididas pelo número de alunos. O algoritmo para resolver esse problema em um computador deve inserir cada nota, armazenar o total de todas as notas inseridas, realizar o cálculo da média e imprimir o resultado.

O algoritmo em pseudocódigo com repetição controlada por contador

Vamos utilizar o pseudocódigo para listar as ações a executar e especificar a ordem em que elas devem ser executadas. Utilizamos a **repetição controlada por contador** para inserir as notas, uma por vez. Essa técnica utiliza uma variável chamada **contador** (ou a **variável de controle**) para controlar o número de vezes que um conjunto de instruções será executado. A repetição controlada por contador costuma ser chamada de **repetição definida**, porque o número de repetições é conhecido antes de o loop começar a executar. Nesse exemplo, a repetição termina quando o contador excede 10. Esta seção apresenta um algoritmo e uma versão da classe

4.9 Formulando algoritmos: repetição controlada por sentinela

Vamos generalizar o problema de média da classe da Seção 4.8. Considere o seguinte problema:

Desenvolva um programa para tirar a média da classe que processe as notas de acordo com um número arbitrário de alunos toda vez que é executado.

No exemplo anterior de média da classe, a declaração do problema especificou o número de alunos, assim o número de notas (10) era conhecido antecipadamente. Neste exemplo, nenhuma indicação é dada de quantas notas o usuário vai inserir durante a execução do programa. O programa deve processar um número arbitrário de notas. Como podemos determinar quando parar de inserir as notas? Como saber quando calcular e imprimir a média da classe?

Uma maneira de resolver esse problema é utilizar um valor especial chamado **valor de sentinela** (também chamado **valor de sinal**, **valor fictício** ou **valor de flag**) para indicar o ‘final da entrada de dados’. O usuário insere as notas até que todas as notas legítimas tenham sido inseridas. O usuário então digita o valor de sentinela para indicar que nenhuma outra nota será inserida. A repetição controlada por sentinela é frequentemente chamada **repetição indefinida** uma vez que o número de repetições não é conhecido antes de o loop iniciar a execução.

Obviamente, deve-se escolher um valor de sentinela que não possa ser confundido com um valor aceitável de entrada. As notas em um questionário são inteiros não-negativos, portanto, para esse problema, -1 é um valor aceitável de sentinela. Assim, uma execução do programa de média de classe talvez processe um fluxo de entradas como 95, 96, 75, 74, 89 e -1 . O programa então computaria e imprimiria a média de classe para as notas 95, 96, 75, 74 e 89 (-1 é o valor de sentinela, ele não deve entrar no cálculo da média).

Repetição contada

Deitel capítulo 5, página 127, 6ª edição

Princípios básicos da repetição controlada por contador

1. uma **variável de controle** (ou contador de loop)
2. o **valor inicial** da variável de controle
3. o **incremento** (ou **decremento**) pelo qual a variável de controle é modificada a cada passagem pelo loop (também conhecido como **cada iteração do loop**)
4. a **condição de continuação do loop** que determina se o loop deve continuar

Exemplo

Exibir números de 1 a 10 na mesma linha

Comando for

```
package comandosrepeticao;  
public class ComandosRepeticao {  
    public static void main(String[] args) {  
  
        for (int i = 1; i <= 10; i++) {  
  
            System.out.print(i + " ");  
        }  
        System.out.println("");  
    }  
}
```

i++;

é o mesmo

i = i + 1;

Exibir números de
1 a 10 na mesma
linha

A primeira parte atribui um valor inicial à variável **i**, que tem como função **controlar** o número necessário de repetições.

A segunda parte corresponde a uma expressão relacional que, quando assumir valor falso, **determinará o fim da repetição.**

A terceira parte é responsável por **alterar o valor da variável i** (incremento ou decremento) com o objetivo de, em algum momento, fazer com que a condição assumira valor falso.

Comando while

```
package comandosrepeticao;  
public class ComandosRepeticao {  
    public static void main(String[] args) {  
  
        int i = 1;  
  
        while (i <= 10) {  
            System.out.print(i + " ");  
            i++;  
        }  
        System.out.println("");  
    }  
}
```

Exibir números de
1 a 10 na mesma
linha

A primeira parte atribui um valor inicial à variável `i`, que tem como função controlar o número necessário de repetições.

A segunda parte corresponde a uma expressão relacional que, quando assumir valor falso, determinará o fim da repetição.

A terceira parte é responsável por alterar o valor da variável `i` (incremento ou decremento) com o objetivo de, em algum momento, fazer com que a condição assumira valor falso.

Comando do-while

```
package comandosrepeticao;  
public class ComandosRepeticao {  
    public static void main(String[] args) {  
  
        int i = 1;  
  
        do {  
            System.out.print(i + " ");  
            i++;  
        } while (i <= 10);  
        System.out.println("");  
    }  
}
```

Exibir números de
1 a 10 na mesma
linha

A primeira parte atribui um valor inicial à variável `i`, que tem como função controlar o número necessário de repetições.

A segunda parte corresponde a uma expressão relacional que, quando assumir valor falso, determinará o fim da repetição.

A terceira parte é responsável por alterar o valor da variável `i` (incremento ou decremento) com o objetivo de, em algum momento, fazer com que a condição assumira valor falso.

Exibir números de 1 a 10

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i + " ");  
}
```

```
int i = 1;  
while (i <= 10) {  
    System.out.print(i + " ");  
    i++;  
}
```

```
int i = 1;  
do {  
    System.out.print(i + " ");  
    i++;  
} while (i <= 10);
```

A primeira parte atribui um valor inicial à variável **i**, que tem como função controlar o número necessário de repetições.

A segunda parte corresponde a uma expressão relacional que, quando assumir valor falso, determinará o fim da repetição.

A terceira parte é responsável por alterar o valor da variável **i** (incremento ou decremento) com o objetivo de, em algum momento, fazer com que a condição assuma valor falso.

Laço de repetição

Pode-se usar qualquer um dos 3 comandos de repetição para construir qualquer um dos 2 tipos de repetição, mas

- for é o mais conveniente para repetições contadas;
- while é conveniente quando o conjunto de comandos deve ser executado só depois do teste da condição. Pode ocorrer do laço não ser executado nem 1 vez;
- do while é conveniente quando o conjunto de comandos precisa ser executado antes do teste da condição. O laço é executado pelo menos 1 vez.

Exemplo

1) Faça um programa que receba 5 números inteiros e calcule e mostre:

- a) a soma dos números digitados
- b) a média dos números digitados
- c) o maior e o menor número digitado
- d) a qtde de números pares digitados
- e) a qtde de números ímpares digitados

Exemplo

1) Faça um programa que receba 5 números inteiros e calcule e mostre:

a) a soma dos números digitados **soma = soma + valor**

b) a média dos números digitados **média = soma dos valores / quantidade de valores**

c) o maior e o menor número digitado

d) a qtde de números pares digitados **if(num % 2 == 0){
contPar = contPar + 1;
}**

e) a qtde de números ímpares digitados

Exemplo

1) Faça um programa que receba 5 números inteiros e calcule e mostre:

a) a soma dos números digitados **soma = soma + valor**

b) a média dos números digitados **média = soma dos valores / quantidade de valores**

c) o maior e o menor número digitado

d) a qtde de números pares digitados **if(num % 2 == 0){**
contPar = contPar + 1;
}

e) a qtde de números ímpares digitados **if(num % 2 == 1){**
contImpar = contImpar + 1;
}

ou **conlpar = 5 - contPar**

Exemplo

1) Faça um programa que receba 5 números inteiros e calcule e mostre:

a) a soma dos números digitados

b) a média dos números digitados

c) o maior e o menor número digitado

d) a qtde de números pares digitados

e) a qtde de números ímpares digitados

Exemplo

Busca do maior

7 3 1 2 8 9 5

Busca do menor

7 3 1 2 8 9 5

Exemplo

Busca do maior

7 3 1 2 8 9 5

Busca do menor

7 3 1 2 8 9 5

Exemplo

Busca do maior

7 3 1 2 8 9 5

7

Busca do menor

7 3 1 2 8 9 5

3

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5

Exemplo

Busca do maior

7 3 1 2 8 9 5

7 1 2 8 9 5

Busca do menor

7 3 1 2 8 9 5

3 1 2 8 9 5

Exemplo

Busca do maior

7 3 1 2 8 9 5

7 1 2 8 9 5

7

Busca do menor

7 3 1 2 8 9 5

3 1 2 8 9 5

1

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
7	2	8	9	5		

Busca do menor

7	3	1	2	8	9	5
3	1	2	8	9	5	
1	2	8	9	5		

Exemplo

Busca do maior

7 3 1 2 8 9 5

7 1 2 8 9 5

7 2 8 9 5

Busca do menor

7 3 1 2 8 9 5

3 1 2 8 9 5

1 2 8 9 5

Exemplo

Busca do maior

7 3 1 2 8 9 5

7 1 2 8 9 5

7 2 8 9 5

7

Busca do menor

7 3 1 2 8 9 5

3 1 2 8 9 5

1 2 8 9 5

1

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
7	2	8	9	5		
7	8	9	5			

Busca do menor

7	3	1	2	8	9	5
3	1	2	8	9	5	
1	2	8	9	5		
1	8	9	5			

Exemplo

Busca do maior

7 3 1 2 8 9 5
7 1 2 8 9 5
7 2 8 9 5
7 8 9 5
8

Busca do menor

7 3 1 2 8 9 5
3 1 2 8 9 5
1 2 8 9 5
1 8 9 5
1

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	
			8	9	5	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5
				1	9	5

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	
			8	9	5	
			8	9	5	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5
				1	9	5
				1	9	5

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	
			8	9	5	
			8	9	5	

9

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5
				1	9	5
					1	5

1

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	
			8	9	5	
				9	5	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5
				1	9	5
					1	5

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
7	2	8	9	5		
7	8	9	5			
8	9	5				
9	5					

Busca do menor

7	3	1	2	8	9	5
3	1	2	8	9	5	
1	2	8	9	5		
1	8	9	5			
1	9	5				
1	5					

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
7	2	8	9	5		
7	8	9	5			
8	9	5				
9		5				
9						

Busca do menor

7	3	1	2	8	9	5
3	1	2	8	9	5	
1	2	8	9	5		
1	8	9	5			
1	9	5				
1		5				
1						

Exemplo

Busca do maior

7	3	1	2	8	9	5
7	1	2	8	9	5	
	7	2	8	9	5	
		7	8	9	5	
			8	9	5	
				9	5	
					9	

Busca do menor

7	3	1	2	8	9	5
	3	1	2	8	9	5
		1	2	8	9	5
			1	8	9	5
				1	9	5
					1	5
						1

Netbeans

Exemplo laço controlado por sentinela

Ascêncio e Campos página 142

16. Faça um programa que receba várias idades, calcule e mostre a média das idades digitadas. Finalize digitando idade igual a zero.

Exercícios

1) A prefeitura de uma cidade fez uma pesquisa entre 5 habitantes, coletando dados sobre o salário e o número de filhos. A prefeitura deseja saber:

- a) média do salário da população;
- b) média do número de filhos;
- c) maior salário;
- d) percentual de pessoas com salário até R\$1000,00.

2) Faça um programa que receba a idade, a altura e o peso de 5 pessoas, calcule e mostre:

- a) a quantidade de pessoas com idade superior a 50 anos;
- b) a média das alturas das pessoas com idade entre 10 e 20 anos;
- c) a porcentagem de pessoas com peso inferior a 40 kg entre todas as pessoas analisadas.