

GCC178 - Práticas de Programação Orientada a Objetos

Prática da Linguagem Java

Luiz Henrique de Campos Merschmann
Departamento de Ciência da Computação
Universidade Federal de Lavras

luiz.hcm@ufla.br

Na Aula de Hoje



- A Linguagem Java:
 - Tipos de Dados
 - Declaração de Variáveis
 - Atribuição e Impressão de Variáveis
 - Controle de Fluxo:
 - *if / else, while, for, do / while, switch*
 - Arrays

A Linguagem Java

Tipos de Dados Primitivos

- ▶ *short, int, float, double, long, char, boolean e byte.*

Declaração de Variáveis

tipoDaVariável nomeDaVariável;

Exemplos:

- ▶ **int** tamanho;
- ▶ **double** temperatura;
- ▶ **char** genero;

Declaração, Atribuição e Impressão de Variáveis

```
//Declarando a variável tamanho
```

```
int tamanho;
```

```
//Atribuindo o valor 10 à variável tamanho
```

```
tamanho = 10;
```

```
//Imprimindo o conteúdo da variável tamanho
```

```
System.out.println(tamanho);
```

```
//Concatenando a saída com o operador +
```

```
System.out.println("Tamanho = " + tamanho);
```

Observações sobre Atribuições

- ▶ Incompatibilidade na atribuição.

Exemplo:

```
double x = 5.23
```

```
int d = x;
```

Isso **não compila** e será acusado o seguinte **erro**:

incompatible types: possible lossy conversion from double to int.

- ▶ Situação permitida:

```
int d = 7;
```

```
double x = d;
```

Isso **compila** sem problema, uma vez que um *double* pode armazenar um número com ponto flutuante ou inteiro.

Observações sobre Atribuições

- ▶ O código a seguir compila corretamente?

```
float a = 3.14;
```

Não. Por que?

Resp.: Todos os literais (constantes inseridas explicitamente no código) com ponto decimal são considerados do tipo *double* pelo Java. Como *float* não pode receber um *double* sem perda de informação, esse código não compila corretamente.

- ▶ Como fazer a atribuição acima funcionar?

```
float a = 3.14f;
```

A letra *f* indica que aquele literal deve ser tratado como um *float*.

Observações sobre Atribuições

Casting

- ▶ Em algumas situações precisamos que um valor em ponto flutuante seja **truncado** e armazenado em uma variável inteira. Para não haver erro de compilação, é preciso ordenar que o valor em ponto flutuante seja moldado (casted) como um número inteiro.

Exemplo:

```
double x = 1.23;
```

```
int i = (int) x;
```

Controle de Fluxo

If

```
if (condicaoBooleana) {  
    código;  
}
```

onde *condicaoBooleana* é qualquer expressão que retorne *verdadeiro* ou *falso*.

Exemplo:

```
if (tamanho < 1.20) {  
    System.out.println("Não pode passar");  
}
```


Controle de Fluxo

If \ Else

```
if (condicaoBooleana) {  
    código;  
} else {  
    outro código;  
}
```

onde *condicaoBooleana* é qualquer expressão que retorne *verdadeiro* ou *falso*.

Exemplo:

```
if (tamanho < 1.20) {  
    System.out.println("Não pode passar");  
} else {  
    System.out.println("Pode passar");  
}
```

Controle de Fluxo

If \ Else \ If

```
if (condicaoBooleana1) {  
    código1;  
} else if (condicaoBooleana2) {  
    código2;  
} else {  
    código3;  
}
```

Exemplo:

```
if (nota >= 60) {  
    System.out.println("Aprovado");  
} else if (nota >= 50) {  
    System.out.println("Reprovado – direito a turma Z");  
} else {  
    System.out.println("Reprovado – sem direito a turma Z");  
}
```

Controle de Fluxo

While

```
while (condicaoBooleana) {  
    código;  
}
```

onde *condicaoBooleana* é qualquer expressão que retorne *verdadeiro* ou *falso*.

Exemplo:

```
i = 50;  
while (i < 100) {  
    System.out.println(i);  
    i+=1;  
}
```

Controle de Fluxo

For

```
for (inicialização; condição; incremento) {  
    código; // executa enquanto a condição for verdadeira.  
}
```

Exemplo:

```
for (int i = 50; i < 100; i++) {  
    System.out.println(i);  
}
```

Controle de Fluxo

For - Controlando o *loop*

- ▶ Mesmo tendo a condição booleana em nossos laços (for), em algumas situações podemos desejar parar o *loop* antes da condição booleana se tornar *falsa*.
- ▶ Para isso utilizaremos o comando *break*.

Exemplo:

```
for (int i = x; i < y; i++) {  
    if (i % 19 == 0) {  
        System.out.println("Achei!");  
        break;  
    }  
}
```

O que esse código faz?

Encontra o primeiro número divisível por 19 entre x e y.

Controle de Fluxo

Do \ While

```
do {  
    código; // executa enquanto a condicao for verdadeira;  
} while(condicao);
```

- Observe que agora o teste da condição de parada é feito no fim. Desse modo, esse laço é executado pelo menos uma vez.

Exemplo:

```
char opcao = lerOpcao();  
do {  
    código; // faça algo com base na opção escolhida  
    opcao = lerOpcao();  
} while (opcao != 'S');
```

Controle de Fluxo

Switch

```
switch (expressao) {  
    case a:  
        código do case a;  
    case b:  
        código do case b;  
    case c:  
        código do case c;  
    default:  
        código do default;  
}
```

onde:

- ▶ *expressao* deve ser do tipo *byte*, *short*, *int*, *char* ou *String*.
- ▶ cada **case** deve conter uma expressão integral constante, ou seja, um valor constante do tipo *byte*, *short*, *int*, *char* ou *String* (a expressão também pode ser uma variável constante – *final*).

Controle de Fluxo

Switch

Exemplo:

```
int x;  
:  
switch (x/10) {  
    case 6:  
        código do case 6;  
    case 7:  
        código do case 7;  
    case 8:  
        código do case 8;  
    default:  
        código do default;  
}
```


Arrays

- ▶ Objetos *array* são estruturas de dados consistindo em itens de dados do mesmo tipo.
- ▶ *Arrays* são úteis quando se deseja processar grupos de valores relacionados. Exemplo: conjunto de notas de alunos.

The diagram illustrates an array structure. On the left, the text 'Nome do array (c)' has an arrow pointing to the first column of the table. Below it, the text 'Índice (ou subscripto) do elemento no array c' has an arrow pointing to the second column of the table. The table itself has 12 rows, each representing an element in the array.

c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6453
c[11]	78

Declaração, Criação e Inicialização de *Arrays*

- ▶ Para criar um *array*, especifique o tipo dos elementos do *array* e o número de elementos¹.

Exemplo:

```
int[ ] c; //declara a variável de array
```

```
c = new int[10]; //cria o array
```

OU

```
int[ ] c = new int[10];
```

Criando vários arrays em uma única declaração:

```
String[ ] a = new String[10], b = new String[50];
```

- ▶ Inicialização de um *array*

Exemplo:

```
int[ ] c = {32, 45, 21, 6, 85, 4};
```

¹O tamanho dos *arrays* permanece fixo depois que eles são criados.

Exemplo que Utiliza *Array*

```
public class ManipularArray{
    public static void main(String[] args){
        int[] v = {23,34,45,56};

        //Título das colunas
        System.out.printf("%s%8s%n", "Índice", "Valor");

        //Imprimindo o valor de cada elemento do array
        for(int cont = 0; cont < v.length; cont++){
            System.out.printf("%5d%8d%n", cont, v[cont]);
        }
    }
}
```

Saída:

Índice	Valor
0	23
1	34
2	45
3	56

Exercícios



Exercício 1

Devido ao déficit previdenciário brasileiro, o governo propôs novas regras de aposentadoria. Para os trabalhadores que já estão na ativa, existe uma regra de transição que prevê um pedágio (tempo extra) de 30% sobre o tempo que faltaria para atingir 30 anos de trabalho, se for mulher, ou 35 anos de trabalho, se for homem. Por exemplo, uma mulher de 50 anos de idade que já trabalhou 25 anos deverá pagar um pedágio de 30% sobre os 5 anos restantes (que faltariam para completar 30 anos de trabalho), ou seja, terá que trabalhar por mais 6,5 anos. Com isso, ela irá se aposentar aos 56,5 anos, quando alcançará 31,5 anos trabalhados. No entanto, essa regra de transição só é válida para pessoas que já estão trabalhando por mais de 15 anos (15 não incluso), no caso de mulheres, ou mais de 20 anos (20 não incluso), no caso de homens. Para todos os demais casos, a aposentadoria se dará por idade mínima, ou seja, 62 anos para mulheres e 65 anos para homens. Faça um programa que leia o gênero, a idade e a quantidade de anos já trabalhados por uma pessoa e calcule a idade com a qual a mesma irá se aposentar.

Exercício 1 (continuação)

Entradas:

- ▶ O gênero da pessoa, que será o valor inteiro 1 se for homem ou 0 se for mulher.
- ▶ Idade da pessoa (valor inteiro).
- ▶ Quantidade de tempo (anos) já trabalhados pela pessoa (número real).

Saídas:

- ▶ Idade que a pessoa terá a se aposentar (número real).

Exemplo de Entrada:

0
50
25

Exemplo de Saída:

56.5

Exemplo de Entrada:

1
35
10

Exemplo de Saída:

65

Exercício 2

Você quer fazer um programa para determinar quanto de dinheiro terá após vários meses de investimento. No início de cada mês, você deposita uma quantidade. Ao fim de cada mês, é adicionado $x\%$ ao valor total possuído até aquele momento.

Entradas:

- ▶ Valor real x entre 0 e 100%, representando o retorno mensal do investimento
- ▶ Inteiro N representando o número de meses
- ▶ N valores reais correspondendo aos depósitos em cada mês

Saídas:

- ▶ Valor total no final do último mês

Exercício 2 (continuação)

Exemplo de Entrada:

0.1

5

100

200

300

400

500

Exemplo de Saída:

1503.5035021007

Exercício 3

Dois pilotos resolveram se desafiar para saber quem faz o melhor tempo no Autódromo Internacional de Lavras. Sabe-se que a cada km a partir do km 1 é registrado o tempo do piloto. Faça um programa que receba o nome do piloto e os tempos registrados dos dois pilotos e imprima em uma linha quem estava na liderança a cada km e em outra linha quem fez o melhor tempo em cada trecho. O tempo de cada trecho é dado pelo tempo registrado no km atual menos o tempo registrado no km anterior, considere que no km 0 o tempo é 0.

Todas as comparações podem ser feitas supondo que não haverá empate.

Exercício 3 (continuação)

Entradas:

- ▶ Quantidade de km em que houve medição.
- ▶ Nome do primeiro piloto.
- ▶ Tempos do primeiro piloto em cada km (números reais).
- ▶ Nome do segundo piloto.
- ▶ Tempos do segundo piloto em cada km (números reais).

Saídas:

- ▶ Nome do piloto na liderança a cada km.
- ▶ Nome do piloto com o melhor tempo no último km.

Exemplo de Entrada:

5

Rubinho

25.1 52.8 75.2 98.4 117.7 (um número em cada linha)

Zeca

24.9 53 75.3 97.6 118.1 (um número em cada linha)

Exemplo de Saída:

Zeca Rubinho Rubinho Zeca Rubinho

Zeca Rubinho Zeca Zeca Rubinho

Perguntas?

