

Programação Orientada a Objetos

Prof. Ricardo Frohlich da Silva

Programação Orientada a Objetos

- Ementa:
 - Orientação a objetos, Sintaxe de linguagens de programação orientadas a objetos, Programação orientada a objetos, Reutilização de classes.

Programação Orientada a Objetos

- Programa:
 - Unidade 1 - Orientação a objetos
 - 1.1 Características de linguagens orientadas a objetos
 - 1.2 Compiladores e máquinas virtuais
 - 1.3 Boas práticas em programação orientada a objetos
 - Unidade 2 - Sintaxe de linguagens de programação orientadas a objetos
 - 2.1 Tipos de dados e operadores
 - 2.2 Comandos de controle de fluxo
 - 2.3 Tratamento de exceções

Programação Orientada a Objetos

- Programa:
 - Unidade 3 - Programação orientada a objetos
 - 3.1 Classes, objetos, atributos e métodos
 - 3.2 Construtores e destrutores
 - 3.3 Encapsulamento do objeto e modificadores de acesso
 - Unidade 4 - Reutilização de classes
 - 4.1 Herança
 - 4.2 Polimorfismo
 - 4.3 Classes abstratas e interfaces

Programação Orientada a Objetos

- O processo de avaliação da disciplina será quantitativo, composto por três notas resultantes dos produtos de aprendizagem e dos exercícios realizados em sala de aula e laboratório.
- As notas 1, 2 e 3 serão obtidas da seguinte forma:
 - $N1 = (\text{Produto de aprendizagem 1} * 0.6) + (\text{Exercícios e teste} * 0.4)$
 - $N2 = (\text{Produto de aprendizagem 2} * 0.6) + (\text{Exercícios, teste e trabalhos} * 0.4)$
 - $N3 = (\text{Produto de aprendizagem 3} * 0.6) + (\text{Exercícios, teste e trabalhos} * 0.4)$

Programação Orientada a Objetos

- A nota final da disciplina será obtida por meio da média aritmética simples entre as notas N1, N2 e N3 conforme cálculo abaixo:
 - $\text{Nota Final} = (N1 + N2 + N3) / 3$
- O critério de arredondamento da nota final levará em conta a participação dos alunos em sala de aula e a entrega dos trabalhos/exercícios.
- Serão considerados aprovados na disciplina os alunos que obtiverem a nota final igual ou superior a sete (6,0) e frequência igual ou superior a setenta e cinco por cento (75%).
- $\text{Aluno Aprovado} = (\text{Nota Final} \geq 6,0) + (\text{Frequência} \geq 75\%).$

Avaliações

- Produto de aprendizagem 1 – 16/04
- Produto de aprendizagem 2 – 28/05
- Produto de aprendizagem 3 – 18/06

Programação Orientada a Objetos - Introdução

- Linguagens?
- Início LM & Assembly
- 195x/6x: FORTRAN, ALGOL 60, COBOL
- 197x: Pascal, Smalltalk, C, BASIC
- 198x: C++, Object Pascal / Objective C
- 1991: VisualBASIC, Oak, Python
- 1995: PHP, Ruby, Java
- 2001: C#

Programação Orientada a Objetos - Introdução

- Linguagens?
- Início LM & Assembly
- 195x/6x: FORTRAN, ALGOL 60, COBOL
- 197x: Pascal, Smalltalk, C, BASIC
- 198x: C++, Object Pascal / Objective C
- 1991: VisualBASIC, Oak, Python
- 1995: PHP, Ruby, Java
- 2001: C#

Paradigmas de programação

- Linguagens Imperativas
 - Estruturadas/Procedurais Ex.: COBOL, FORTRAN, C, Pascal...
 - Orientada a Objetos Ex.: Smalltalk, C++, Python, Java, C#
- Linguagens Declarativas
 - Funcionais Ex.: Erlang, R, XSLT
 - Lógicas Ex.: Prolog, LISP

Onde estamos?

- Precisamos desenvolver utilizando um paradigma que é fortemente utilizado no mercado. Qual utilizar?
 - **Programação orientada a objetos!**
- E qual linguagem utilizar?
 - Precisamos de uma linguagem utilizada em grande escala, que possibilite uma alta empregabilidade para vocês, com segurança, multiplataforma.
 - Preciso dizer qual é?

A linguagem de programação



Java

- Uma linguagem similar ao C/C++ cujo foco é economizar tempo do programador e otimizar o processo de desenvolvimento.
- Como?
 - Uma vasta biblioteca pronta
 - Exigência de atendimento aos padrões mais modernos de desenvolvimento de software
 - Compatibilidade **quase** universal: um programa pode ser criado de maneira a ser executado em praticamente qualquer equipamento que seja capaz de processamento

Java

- Como isso se tornou realidade?
 - Até 1994: Foco em Dispositivos Embarcados
 - 1995~: Implementação em navegadores Web
 - Atualmente: Plataforma ampla com múltiplas soluções
 - E como assim, múltiplas soluções?

Java

- A possibilidade de executar em múltiplas plataformas é graças ao Interpretador Java
- O programa Java, depois de compilado, fica em um formato incompreensível para o computador, chamado Byte Code.
- O Interpretador Java é um programa que lê o Byte Code e traduz para o computador, indicando o que ele precisa fazer.
- Basicamente, o interpretador Java faz o trabalho de um intérprete.
- Cada equipamento precisa ter seu próprio interpretador Java.

Java

- Mas o que é este “Interpretador”?
- Nunca ouviram falar?
- Aposto que muitos já, mas não com esta nomenclatura.

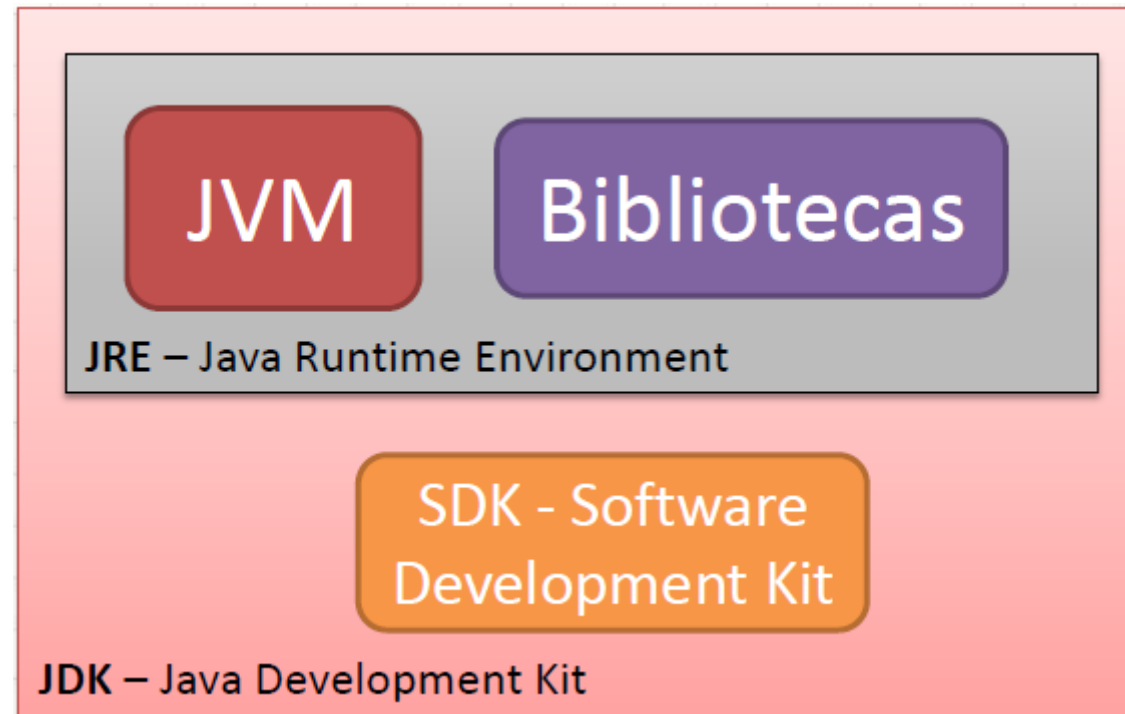
Java

- O nome oficial do Interpretador Java é *Java Virtual Machine*, ou simplesmente JVM.
- Para executar aplicações Java, é necessário instalar a JVM.
- A JVM sozinha, porém, não contém as bibliotecas necessárias para executar tudo que um programa Java precisa.
- Assim, a Oracle distribui um pacote chamado *Java Runtime Environment* (JRE) contendo a JVM e as bibliotecas oficiais do Java.

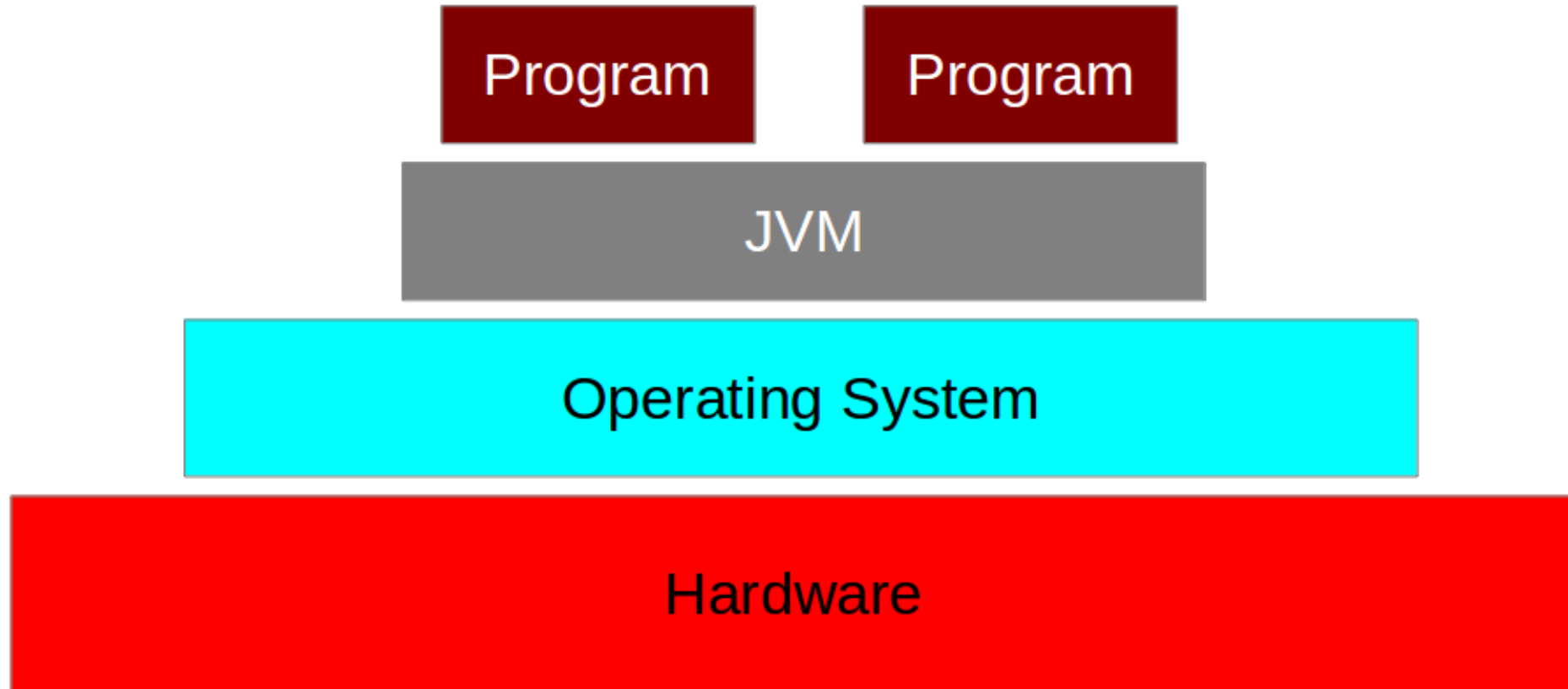
Java

- O JRE inclui apenas os pacotes necessários à execução de programas Java.
- Para poder gerar programas Java, é necessário baixar um pacote mais completo, chamado *Java Development Kit* (JDK).
- O JDK inclui, além dos elementos do JRE, também as ferramentas de desenvolvimento do ambiente Java.

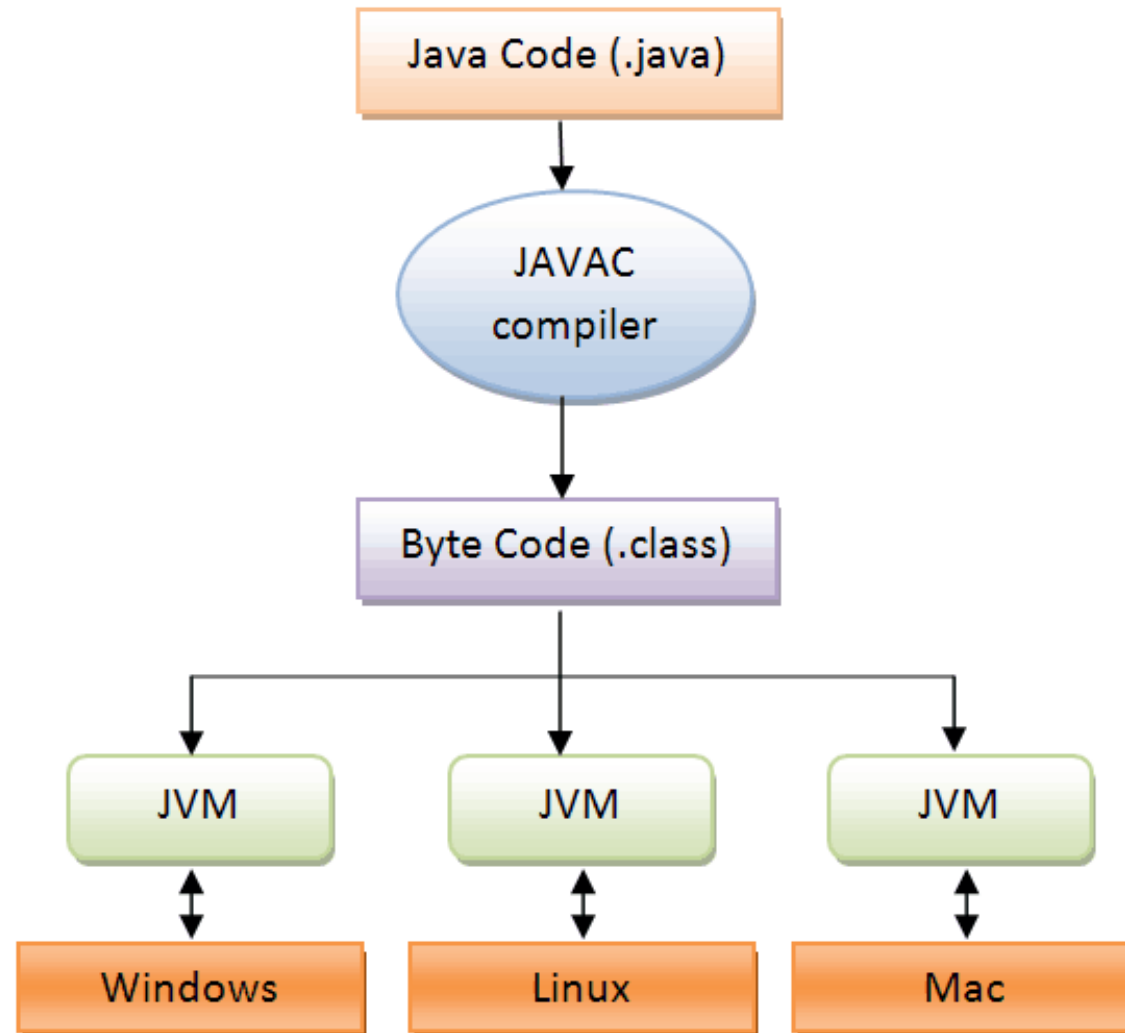
Java



Java



Java



Primeiro programa

- Abra o eclipse e crie um novo projeto Java
- Após, no Package Explorer, clique com o botão direito no nome do projeto e de um New, Class
- Coloque um nome no pacote “Package” e defina o nome da classe (Sugestão: Program). Marque que você deseja que seja criada a main (public static void main (String[] args))
- Clique em Finish

Primeiro programa

- Após, coloque um print. Lembra do “printf” no C?
- Aqui será `System.out.println(“Olá mundo”);`
 - O “In” no final do print significa que após mostrar a mensagem, ele quebrará a linha automaticamente. Caso não queira a quebra de linha, basta usar o `System.out.print()`

Primeiro programa

```
public class Program {  
    ➤    public static void main(String[] args) {  
        System.out.println("Olá mundo!");  
    }  
}
```


Utilizando variáveis

```
public class Program {  
  
    public static void main(String[] args) {  
  
        int idade;  
        idade = 18;  
        System.out.println(idade);  
  
    }  
  
}
```

Utilizando variáveis

```
public class Program {  
  
    public static void main(String[] args) {  
  
        int idade;  
        idade = 18;  
        char c = 'r';  
        double d = 123.4;  
        float f = 3.14f;  
  
        System.out.println(idade);  
        System.out.println(c);  
        System.out.println(d);  
        System.out.println(f);  
  
    }  
  
}
```

Melhorando

```
public class Program {  
  
    public static void main(String[] args) {  
  
        int idade;  
        idade = 18;  
        char c = 'r';  
        double d = 123.4;  
        float f = 3.14f;  
  
        System.out.println("A idade é:" +idade);  
        System.out.println("O valor de c é:" +c);  
        System.out.println("O valor de d é: "+d);  
        System.out.println("O valor de f é:" +f);  
  
    }  
  
}
```

Atalhos

- Cansado de digitar `System.out.println`?

Atalhos

- Cansado de digitar `System.out.println`?
 - `sysout` + `ctrl`+`espaço`

Atalhos

- Outros atalhos:
- Alt+Shift+r
 - Faz com que todas as variáveis que se encontra no código seja modificada. Basta colocar o cursor do mouse na palavra que queira mudar, apertar o atalho e mudar o nome. Simples e rápido;
- Ctrl+d
 - Esse atalho excluí toda a linha onde o cursor do mouse está parado;
- Ctrl+/
 - Com este atalho é possível comentar uma linha, no caso a linha que o cursor está parado. Comentário do tipo “//” e para tirar o comentário basta refazer o mesmo atalho ;

Atalhos

- Outros atalhos:
- Selecionar `Ctrl+ Shift + /`
 - Também gera um comentário do tipo `/* */`, para isso basta selecionar o bloco que se deseja inserir o comentário e logo em fazer os comandos;
- Selecionar `Ctrl+ Shift + \`
 - Agora que vem a grande sacada! Para “descomentar” um bloco como expliquei no tópico 4, basta fazer quase os mesmos passes, a única coisa que muda será o tipo da barra que agora é `“”` e antes era `“/”`;
- `Ctrl+Shift+f`
 - Formata todo o código.

Palavra final

- A palavra final serve para que possamos declarar uma constante, ou seja, o valor atribuído não poderá ser alterado.

```
public class Program {  
  
    public static void main(String[] args) {  
  
        final float PI = 3.14159265f;  
        System.out.println(PI);  
  
    }  
  
}
```


Leitura pelo teclado

- Para fazer a leitura pelo teclado, é necessário utilizar uma biblioteca chamada Scanner

```
package intro;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        String texto;
        Scanner teclado = new Scanner(System.in);
        //lendo um texto
        System.out.println("Digite um texto:");
        texto = teclado.nextLine();

        //apresentando o texto digitado
        System.out.println("O texto digitado foi: "+texto);
    }
}
```

Leitura pelo teclado

- Para fazer a leitura de um inteiro é necessário utilizar outro método (função) da classe Scanner:

```
package intro;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        int n;
        Scanner teclado = new Scanner(System.in);
        //lendo um número
        System.out.println("Digite um número:");
        n = teclado.nextInt();

        //apresentando o texto digitado
        System.out.println("O número digitado foi: "+n);
    }
}
```

Leitura pelo teclado

- E o que fazer para ler um float?

Leitura pelo teclado

- E o que fazer para ler um float?

```
package intro;
import java.util.Scanner;

public class Program {

    public static void main(String[] args) {

        float f;
        Scanner teclado = new Scanner(System.in);
        //lendo um número
        System.out.println("Digite um número:");
        f = teclado.nextFloat();

        //apresentando o texto digitado
        System.out.println("O número digitado foi: "+f);
    }
}
```

Leitura pelo teclado

```
public static void main(String[] args) {  
  
    Scanner teclado = new Scanner(System.in);  
    char c;  
    System.out.print("Digite um caractere: ");  
    c = teclado.next().charAt(0);  
  
    System.out.println("Caractere digitado: "+c);  
  
}
```

Condicionais

```
public class Program {  
    public static void main(String[] args) {  
        Scanner teclado = new Scanner(System.in);  
        int idade;  
  
        System.out.println("Digite a idade: ");  
        idade = teclado.nextInt();  
  
        if(idade >= 18)  
            System.out.println("Maior de idade!");  
        else  
            System.out.println("Menor de idade!");  
    }  
}
```

Switch Case

```
public class Program {  
    public static void main(String[] args) {  
        Scanner entrada = new Scanner(System.in);  
        System.out.println("Escolha um número entre 1 a 5");  
        int numero = entrada.nextInt();  
        switch (numero) {  
            case 1:  
                System.out.println("O número escolhido foi: 1.");  
                break;  
            case 2:  
                System.out.println("O número escolhido foi: 2.");  
                break;  
            case 3:  
                System.out.println("O número escolhido foi: 3.");  
                break;  
            case 4:  
                System.out.println("O número escolhido foi: 4.");  
                break;  
            case 5:  
                System.out.println("O número escolhido foi: 5.");  
                break;  
            default:  
                System.out.println("O número escolhido é inválido! Digite um número entre 1 a 10.");  
        }  
    }  
}
```

Atividades:

- 1 - Faça um programa para leitura de dois números e após faça as quatro operações matemáticas (Soma, Subtração, Multiplicação e divisão)
- 2 - Um motorista de taxi deseja calcular o rendimento de seu carro na praça. Sabendo-se que o preço do combustível é de R\$4,90, escreva um algoritmo para ler a marcação do odômetro (marcador de quilometragem) no início do dia, a marcação no final do dia, o número de litros de combustível gasto e o valor total (R\$) recebido dos passageiros. Calcule e escreva a média do consumo em Km/l e o lucro líquido do dia.
- Fórmulas: Total quilometragem = marcação odômetro final do dia - marcação odômetro início do dia
- Média do consumo = Total quilometragem/ quantidade de combustível gasto
- Lucro do dia = Valor total recebido - (quantidade de combustível gasto * 4,90)

Atividades:

- 3 - Escrever um algoritmo para ler quatro valores float, calcular a sua média, e escrever na tela os que são superiores à média.
- 4 - Escreva um algoritmo que leia um ponto (x,y) pelo teclado e informe em qual quadrante o ponto se encontra no plano cartesiano ou se o ponto está sobre um dos eixos.
- 5 - Escreva um algoritmo que leia 3 valores pelo teclado e então informe qual o maior deles.

Atividades:

- 6 - Ajuste o exercício 5 de maneira que mostre os valores informados em ordem crescente.
- 7 - Escreva um algoritmo que leia dois valores pelo teclado e informe se os mesmos são múltiplos um do outro ou não.
- 8 – Escreva um algoritmo para ler a idade de uma pessoa e mostrar qual sua situação de eleitor, conforme as seguintes condições:
 - Se a idade for menor que 16 anos, informar que a pessoa não está apta a votar.
 - Se a idade for maior ou igual a 16 anos e menor que 18 anos ou superior a 70 anos, informar que o voto é facultativo.
 - Caso contrário, informar que o voto é obrigatório.
- 9 - Altere o exercício 1 e faça um menu utilizando switch case