

Programação e Sistemas de Informação – 2º TSI
Ano Letivo 2019/2020
MÓDULO 09 – INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA POR OBJETOS
Exercício 2

Nome: _____ Data: _____

Objetivo

O Objetivo desta ficha é aplicar os conceitos de Objeto, Classe, Atributo, Método, Construtor e Encapsulamento.

Enunciado

1. Crie um projeto **Exercicio2**.
2. Escreva uma classe "**Carro**" como apresentado:

```
public class Carro {
```

```
    //Atributos
```

```
    String matricula;
```

```
    int velocidadeAtual = 0;
```

```
    int velocidadeMaxima = 200;
```

```
    Condutor condutor;
```

```
    /**
```

```
     * A variável velocidadeMaxima poderá ser uma constante contudo,
```

```
     * esta matéria só será abordada mais tarde
```

```
     */
```

```
    //Construtores de classe
```

```
    public Carro () {
```

```
    }
```

```
    //Métodos
```

```
    void ligar () {
```

```
        System.out.println("VRUUUMMMMMMMMMMMMMM");
```

```
    }
```

```
void desligar () {
    System.out.println("MMMMMMMMMMMMMMMMMMMM");
}
```

```
void acelerar () {
    velocidadeAtual += 10 + condutor.destreza * 0.1;
    if (velocidadeAtual > velocidadeMaxima)
        velocidadeAtual = velocidadeMaxima;
}
```

```
void travar (int intensidadeTravagem) {
    if(intensidadeTravagem > velocidadeMaxima)
        intensidadeTravagem = velocidadeMaxima;
    else
        if (intensidadeTravagem <0)
            intensidadeTravagem = 0;
    velocidadeAtual -= intensidadeTravagem;
    if (velocidadeAtual > velocidadeMaxima)
        velocidadeAtual = velocidadeMaxima;
    else if (velocidadeAtual <0)
        velocidadeAtual = 0;
}
}
```

- 3. Na classe “Carro”, acrescente o atributo “condutor” e altere o método acelerar, de forma a que o cálculo da velocidade atual já implementado sejam adicionados 10% do valor da destreza do condutor;**

//Atributos

Condutor condutor;

//Métodos

```
void acelerar () {
    velocidadeAtual += 10 + condutor.destreza * 0.1;
    if (velocidadeAtual > velocidadeMaxima)
        velocidadeAtual = velocidadeMaxima;
}
```

4. Defina a classe “Corrida” dentro do projeto onde estão as classes “Carro” e “Condutor”. Execute a classe e verifique o vencedor:

```
public class Corrida {
    //Construtores
    public Corrida (){}
}
//Métodos
public static void main (String [] args) {
    /**
     * Cria objetos carro1 e carro2 que são instâncias da classe Carro
     */
    Carro carro1 = new Carro();
    Carro carro2 = new Carro();
    /**
     * Define valores para o atributo matrícula dos objetos carro1 e carro2
     */
    carro1.matricula = "11-AA-11";
    carro2.matricula = "22-BB-22";
    /**
     * Cria os objetos condutor1 e condutor2, que são instâncias da classe
    Condutor,
     * e irão conduzir os carros
     */
    Condutor condutor1 = new Condutor ();
    Condutor condutor2 = new Condutor ();
    /**
     * Define os valores para os atributos do objeto condutor1
     */
    condutor1.nome = "António";
    condutor1.idade = 25;
    condutor1.destreza = 75;
    /**
     * Define os valores para os atributos do objeto condutor2
     */
    condutor2.nome = "Manuel";
    condutor2.idade = 27;
    condutor2.destreza = 65;
    //Coloca os condutores nos seus carros
    carro1.condutor = condutor1;
    carro2.condutor = condutor2;
    //Liga os carros
    carro1.ligar ();
    carro2.ligar ();
    //Inicia a corrida
    carro1.acelerar ();
    carro2.acelerar ();
    carro1.acelerar ();
    carro2.travar (10);
    carro1.acelerar ();
}
```

```

        carro1.acelerar ();
        carro1.travar (5);
        carro2.acelerar ();

        System.out.println("Velocidade do carro1: " +
        carro1.velocidadeAtual);
        System.out.println("Velocidade do carro2: " +
        carro2.velocidadeAtual);
        //Desliga os carros
        carro1.desligar ();
        carro2.desligar ();
    }
}

```

5. Modifique o resultado da corrida, através do atributo “destreza” e dos métodos acelerar e travar, de forma a que o “carro2” passe a ser o vencedor.
6. Adicione à classe “Corrida” a linha de código que se segue e reflita sobre o resultado obtido: Carro2.velocidadeAtual =300;
7. Acrescente o modificador private aos atributos da classe “Carro” e construa um getter e um setter para os atributos “matricula” e “condutor” e um getter para o atributo “velocidadeAtual”.
8. Acrescente o modificador private aos atributos da classe “Condutor” e construa um getter e um setter para cada um dos seus atributos.
9. Modifique as classes “Carro” e “Corrida”, tendo em conta que o acesso aos atributos das classes “Carro” e “Condutor” passou a ser privado.
10. Altere a classe “Carro”, de forma a que um carro não possa ser acelerado ou travado sem antes ter sido ligado.
11. Na classe “Corrida”, coloque em comentário as linhas de código correspondentes à invocação do método ligar. Execute a classe e analise os resultados. Retire o comentário que efetuou anteriormente.
12. Acrescente à classe “Carro” um método que permita saber se um carro está ligado ou não.
13. Modifique a classe “Corrida”, testando se os carros estão ligados antes de começarem a correr e desligados depois de terminarem.

No Final deverá ter:

Classe Carro:

```
public class Carro {
```

```
    //Atributos
```

```
    private String matricula;
```

```
    private int velocidadeAtual = 0;
```

```
    private int velocidadeMaxima = 200;
```

```
    private Condutor condutor;
```

```
    private boolean ligado = false;
```

```
    //Getters e Setters
```

```
    public String getMatricula () {
```

```
        return matricula;
```

```
    }
```

```
    public void setMatricula (String matricula) {
```

```
        this.matricula = matricula;
```

```
    }
```

```
    public Condutor getCondutor () {
```

```
        return condutor;
```

```
    }
```

```
    public void setCondutor (Condutor condutor) {
```

```
        this.condutor = condutor;
```

```
    }
```

```
    public int getVelocidadeAtual () {
```

```
        return velocidadeAtual;
```

```
    }
```

```
    /*
```

```
    * Não é necessário implementar o método setVelocidadeAtual, uma vez que o
```

```
    * valor do atributo velocidadeAtual só é suposto ser alterado através do uso
```

```
    * dos métodos acelerar e travar.
```

```
    */
```

```
    /**
```

Prof Matilde Vieira



UNião Europeia
Fundo Social Europeu

* A variável velocidadeMaxima poderá ser uma constante contudo, esta matéria só será abordada mais

* tarde

*/

//Construtores de classe

```
public Carro () {  
}
```

//Métodos

```
void ligar ()  
{  
    ligado = true;  
    System.out.println("VRUUUMMMMMMMMMMMMM");  
}
```

```
void desligar () {  
    ligado = false;  
    System.out.println("MMMMMMMMMMMMMMMMMMMM");  
}
```

```
void acelerar () {  
    if(!ligado)  
        System.out.println("O carro não está ligado por isso não pode ser acelerado!");  
    else {  
        velocidadeAtual += 10 + condutor.getDestreza () * 0.1;  
        if (velocidadeAtual > velocidadeMaxima)  
            velocidadeAtual = velocidadeMaxima;  
    }  
}
```

```
void travar (int intensidadeTravagem) {  
    if (!ligado)  
        System.out.println("O carro não está ligado por isso não pode ser travado!");  
    else {  
        if(intensidadeTravagem > velocidadeMaxima)  
            intensidadeTravagem = velocidadeMaxima;  
        else  
            if (intensidadeTravagem <0)
```

```

        intensidadeTravagem = 0;
    }
    /**
     * 10.Em vez de:
     * if(intensidadeTravagem > velocidadeMaxima)
     *     intensidadeTravagem = velocidadeMaxima;
     * else
     *     deverá ter:
     *     if(intensidadeTravagem > 200)
     *         intensidadeTravagem = 200;
     *     else
     */
    velocidadeAtual -= intensidadeTravagem;
    if (velocidadeAtual > velocidadeMaxima)
        velocidadeAtual = velocidadeMaxima;
    else if (velocidadeAtual < 0)
        velocidadeAtual = 0;
}
//12.Método que permite saber se o carro está ligado
public boolean isLigado () {
    return ligado;
}
}

```

Classe Condutor:

```

public class Condutor
{
    //Atributos
    private String nome;
    private int idade;
    private int destreza;

    //Getters e Setters
    public String getNome () {
        return nome;
    }

    public void setNome(String nome) {

```

Prof Matilde Vieira

```

        this.nome = nome;
    }

    public int getIdade () {
        return idade;}

    public void setIdade (int idade) {
        this.idade = idade;
    }

    public int getDestreza () {
        return destreza;
    }

    public void setDestreza (int destreza) {
        this.destreza = destreza;
    }

    //Construtores
    public Condutor () {
    }
}

```

Classe Corrida:

```

public class Corrida {
    //Construtores
    public Corrida (){}
}

//Métodos
public static void main (String [] args) {
    /**
     * Cria objetos carro1 e carro2 que são instâncias da classe Carro
     */
    Carro carro1 = new Carro();
    Carro carro2 = new Carro ();
    /**
     * Define valores para o atributo matrícula dos objetos carro1 e carro2
     */
}

```



```

carro1.setMatricula("11-AA-11");
carro2.setMatricula("22-BB-22");
/**
 * Cria os objetos condutor1 e condutor2, que são instâncias da classe Condutor,
 * e irão conduzir os carros
 */
Condutor condutor1 = new Condutor ();
Condutor condutor2 = new Condutor ();
/**
 * Define os valores para os atributos do objeto condutor1
 */
condutor1.setNome("António");
condutor1.setIdade(25);
condutor1.setDestreza(75);
/**
 * Define os valores para os atributos do objeto condutor2
 */
condutor2.setNome("Manuel");
condutor2.setIdade(27);
//5. Passar a destreza do condutor 2 para 90 ao invés de 65.
condutor2.setDestreza(90);

//Coloca os condutores nos seus carros
carro1.setCondutor (condutor1);
carro2.setCondutor (condutor2);

//Liga os carros
carro1.ligar ();
carro2.ligar ();

//Testa se os carros estão ligados ou desligados
System.out.println("O carro1 está ligado? " + carro1.isLigado());
System.out.println("O carro2 está ligado? " + carro2.isLigado());

//Inicia a corrida
carro1.acelerar ();
carro2.acelerar ();
carro1.acelerar ();

```

```

//5. Passar a travagem de 10 para 2
carro2.travar (2);
carro1.acelerar ();
carro1.acelerar ();
carro1.travar (5);
carro2.acelerar ();

/**
 * 6. É possível fazer carro2.velocidadeAtual = 300; contudo viola duas regras:
 * a velocidade máxima foi de 200 e é suposto que a velocidadeAtual seja apenas
 * modificada pelos Métodos acelerar e travar.
 */

System.out.println("Velocidade do carro1: " + carro1.getVelocidadeAtual());
System.out.println("Velocidade do carro2: " + carro2.getVelocidadeAtual());

//Desliga os carros
carro1.desligar ();
carro2.desligar ();
}
}

```