



Objetivos:

- Compreender os conceitos de objeto, classe, atributo, método, constructor e encapsulamento
- Exercícios foram retirados/baseados nos exercícios do livro: Curso Prático de Java, Carla Jesus, FCA, 2013

Plano de aula:

Tomar como base as notas das aulas teóricas e os conceitos aprendidos na UC de programação orientada a objetos, para resolver exercícios de revisão que exemplificam o paradigma orientado a objetos

Ferramentas usadas: NetBeans e Java 8

Exercício 1:

1. Crie um projeto denominado “Exercício1”.

a) Defina a classe “Pessoa” apresentada abaixo:

Pessoa
nome: string genero: char idade: int
void falar()

Lembre-se: as propriedades da classe devem ser acedidas apenas através de seus métodos (ou seja, crie os getters e setters para aceder as propriedades de Pessoa e use o modificador “private” para não permitir outra forma de acesso).

b) Escreva na classe anterior os seguintes construtores:

```
/* Construtor default para os objetos da classe
Pessoa*/
public Pessoa() { }
public Pessoa(String n, char g){
    this.nome = n;
    this.genero = g;
}
public Pessoa(String n, char g, int i){
    this(n,g); // usa o construtor anterior
    this.idade = i; }
```



c) Defina a classe “UsaPessoa” como apresentado abaixo:

```
package exercicio1;

public class UsaPessoa {
    // construtor default para objetos da classe
    // UsaPessoa este construtor existe mesmo se não
    // for explicitamente declarado

    public UsaPessoa() {}
    public static void main(String[] args){
        //Declara objeto pessoal
        Pessoa pessoal;
        /*Cria o objeto pessoal que é instância da classe
        Pessoa */

        pessoal = new Pessoa();

        //Define os valores para os atributos de pessoal

        pessoal.setNome("Camila");
        pessoal.setGenero('F');
        pessoal.setIdade(12);

        System.out.println(
            "O objeto pessoal chama-se "+pessoal.getNome()+
            ", é do sexo "+pessoal.getGenero()+
            ", tem "+pessoal.getIdade()+" anos e diz: ");

        pessoal.falar(); }
}
```

d) Acrescente a classe “UsaPessoa” o objeto pessoa2, que deverá ser criado com os valores dos atributos introduzidos pelo utilizador .

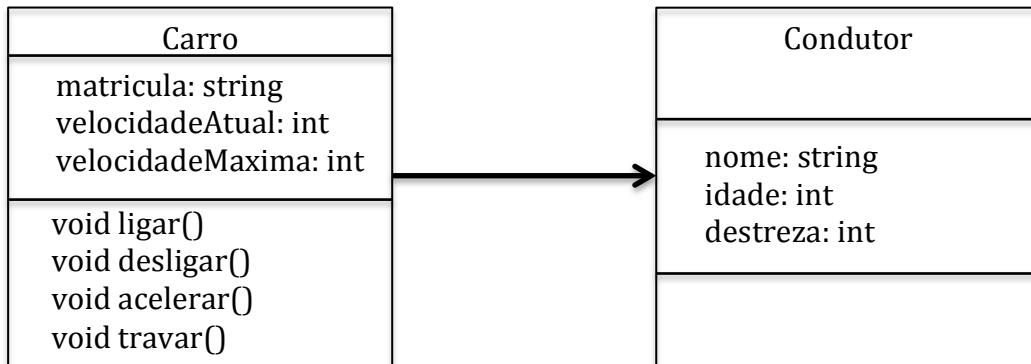
Dicas:

- 1) use *import java.util.Scanner;* para permitir entrada de dados via teclado.
- 2) Crie uma instância da classe Scanner (por exemplo “s”)
- 3) Use *nextLine()* para obter os valores que são texto (por exemplo: *String n = s.nextLine();*)
- 4) Use *nextInt()* para obter valores do tipo inteiro (por exemplo: *int l = s.nextInt();*)
- 5) Use *charAt(0)* para converter uma string em um char.



Exercício 2:

1. Crie um projeto denominado “Exercicio2”.
2. Dentro do projeto anterior, defina as seguintes classes:



velocidadeAtual recebe como valor inicial 0 (zero) e pode ser alterado durante a execução do programa

velocidadeMaxima recebe o valor 200 e não se altera durante a execução do programa

```
void ligar() {
    System.out.println("VRUUUUUmmmmmmmmmm"); }
void desligar() {
    System.out.println("MMMmmmmmm"); }
void acelerar() {
    velocidadeAtual += 10;
    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; }
```

Métodos:

Lembrete: as propriedades da classe devem ser acedidas apenas através de seus métodos (ou seja, crie os getters e setters para aceder as propriedades de Carro e Condutor e use o modificador “private” para não permitir outra forma de acesso). O Código acima deve ser alterado para attender a esses critérios.



3. Crie o atributo “conductor” na classe “Carro”, o qual é um objeto da classe “conductor” e altere o método acelerar, de forma a que o cálculo da velocidade atual já implementado sejam adicionados 10% do valor da destreza do conductor.
4. Defina a classe “Corrida” dentro do projeto onde estão as classes “Carro” e “Conductor”. Execute a classe e verifique o vencedor.

```
public class Corrida {

    //construtores
    public Corrida(){ }

    //Métodos
    public static void main(String[] args) {

        /*Cria os objetos carro1 e carro2 que são
        instancias da classe Carro*/
        Carro carro1 = new Carro();
        Carro carro2 = new Carro();
        carro1.setMatricula("11-AA-11");
        carro2.setMatricula("22-BB-22");

        /* Cria os objetos condutor1 e condutor2 que são
        instancias da classe Conductor*/
        Conductor condutor1 = new Conductor();
        Conductor condutor2 = new Conductor();
        condutor1.setNome("Antonio");
        condutor1.setIdade(25);
        condutor1.setDestreza(75);
        condutor2.setNome("Manuel");
        condutor2.setIdade(27);
        condutor2.setDestreza(65);

        /*Coloca os condutores nos seus carros*/
        carro1.setConductor(condutor1);
        carro2.setConductor(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.getVelocidadeAtual());
System.out.println("Velocidade do carro2: " +
carro2.getVelocidadeAtual());

//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 que o “carro2” passe a ser o vencedor.
6. Altere a classe “Carro”, de forma a que um carro não possa ser acelerado ou travado sem antes ter sido ligado.
7. Na classe “Corrida”, coloque em comentário as linhas de código correspondents à invocação do método ligar. Execute a classe e analise os resultados. Retire o comentário que efetuou anteriormente.
8. Acrescente à classe “Carro” um método que permita saber se um carro está ligado ou não.
9. Modifique a classe “Corrida”, testando se os carros estão ligados antes de começarem a correr e desligados depois de terminarem.



Exercício 3 (exercício 4 do livro):

1. Crie um projeto denominado “Exercicio3”.
2. Crie a classe “Calculadora” com as características seguintes:
 - a) Um constructor sem qualquer parâmetro;
 - b) Quatro métodos que retornem o resultado das operações de soma, subtração, multiplicação e divisão de dois valores recebidos como parâmetro.
3. Crie a classe “FazCalculos1” com as seguintes características:
 - a) Cria uma instância da classe “Calculadora”;
 - b) Soma, subtrai, multiplica e divide os valores 2.7 e 5.2, passados como parâmetro à instância da classe “Calculadora”. Mostra no ecrã o resultado de cada operação.
- 4) Crie a classe “FazCalculos2” com as seguintes características:
 - a) Cria uma instância da classe “Calculadora”;
 - b) Soma, subtrai, multiplica e divide os valores introduzidos pelo utilizador, passados como parâmetro à instância da classe “Calculadora”. Mostra no ecrã o resultado de cada operação.