

P.PORTO	ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO E GESTÃO	Tipo de Prova Exame Época de Recurso	Ano lectivo 2020/2021	Data 12-7-2021
		CURSO Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
		Unidade Curricular Paradigmas de Programação	Duração 2:00 horas	
Observações O exame é sem consulta				

Grupo I (8 valores)

Nota: Respostas erradas não descontam.

- a) A sobrecarga (overloading) de métodos construtores é: *(2 valores)*
- Um processo de substituição de comportamento ao nível do construtor numa subclasse.
 - Uma forma de inicializar objetos com valores por defeito.
 - É uma forma de criar diferentes construtores com nomes diferentes.
 - Nenhuma das anteriores.
- b) Assinale a opções corretas: *(2 valores)*
- Apenas métodos abstratos são permitidos nas interfaces enquanto nas classes abstratas são permitidos métodos concretos e abstratos.
 - As interfaces permitem herança múltipla enquanto as classes abstratas não.
 - Uma classe abstrata não pode implementar interfaces.
 - Uma interface pode estender de uma classe abstrata.
- c) Considere o seguinte excerto de código: *(2 valores)*

```
public class Car {
    public void setDriver(Driver driver) {
        driver = null;
    }

    public static void main(String[] args) {
        Car car = new Car();
        Driver driver = new Driver();
        car.setDriver(driver);
        System.out.println(driver.toString());
    }
}
```

Assinale a opção correta:

- A execução do código resulta na exceção: `NullPointerException`.
- O output gerado na consola é: `null`.
- O output gerado na consola é a *string* correspondente ao retorno do método `toString()`.
- Nenhuma das anteriores

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame Época de Recurso	Ano lectivo 2020/2021	Data 12-7-2021
	CURSO Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Paradigmas de Programação	Duração 2:00 horas	

d) Um método declarado com o modificador `static`: (2 valores)

- i) Pertence exclusivamente aos objetos instanciados.
- ii) Pode modificar o valor de uma variável de instância.
- iii) Não pode ser sobreposto (*override*).
- iv) Nenhuma das anteriores.

Grupo II (6 valores)

Considere o seguinte excerto de código:

```

public class Carro {

    private final String marca;
    private final String modelo;
    private final String matricula;

    public Carro(String marca, String modelo, String matricula) {
        this.marca = marca;
        this.modelo = modelo;
        this.matricula = matricula;
    }

    public String getMarca() { ...3 lines }

    public String getModelo() { ...3 lines }

    public String getMatricula() { ...3 lines }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        }
        if (obj == null) {
            return false;
        }
        if (getClass() != obj.getClass()) {
            return false;
        }
        final Carro other = (Carro) obj;
        return this.matricula.equals(other.getMatricula());
    }

}

public class Taxi extends Carro {
    private final String licenca;

    public Taxi(String licenca, String marca,
                String modelo, String matricula) {
        super(marca, modelo, matricula);
        this.licenca = licenca;
    }
}

public static void main(String[] args) {
    Carro carro1 = new Carro("Toyota", "Yaris", "00-HH-VV");
    Taxi carro2 = new Taxi("00000", "Toyota", "Yaris", "00-HH-VV");
    System.out.println(carro1.equals(carro2));
}

```

- a) Qual é o output apresentado na consola? Justifique explicando o código apresentado (2 valores)
- b) Defina e implemente na classe Carro, uma interface GPS. A interface GPS estipula uma funcionalidade que permite retornar uma string com as coordenadas atuais do veículo. Implemente a interface considerando que existe uma classe Geocoder que possui um método de classe com o nome:

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO E GESTÃO</small>	Tipo de Prova Exame Época de Recurso	Ano lectivo 2020/2021	Data 12-7-2021
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Paradigmas de Programação	Duração 2:00 horas	

`getCurrentLocation()`, que retorna uma string contendo a latitude e longitude da posição atual. (4 valores)

Grupo III (6 valores)

Considere o seguinte domínio:

Pretende-se que desenvolva uma API para suportar o funcionamento de uma empresa de gestão das viagens dos seus clientes nos veículos dos condutores associados.

Considere que:

- Cada cliente é caracterizado pelo seu nome, NIF e numero de telefone;
- Para cada veículo, é mantido um identificador, o nome do seu condutor e a sua capacidade em termos de passageiros. Existem veículos de dois tipos:
 - Bicicleta Tandem - caracterizada por uma cor.
 - Carro – para o qual se regista a cilindrada e a marca.

A Viagem é o conceito central no modelo de negócio. Para cada viagem, a empresa pretende registar o cliente, o veículo em que esta decorre, o nº de kms a percorrer e o seu estado (se está terminada ou ainda a decorrer).

Os serviços são centrados em dois tipos de viagem, com características distintas, e com formas diferentes de cálculo do seu preço:

- Viagem Low Cost – Este tipo de viagem pode ser feita tanto em Carro como em Bicicleta Tandem. Se feita de carro, o seu custo (em euros) é obtido de acordo com a seguinte fórmula: $preco = kms * 2 + cilindrada / 1000 * kms$. Se feita de Bicicleta Tandem, o seu custo é de 3 € por km.
- Viagem VIP – Este tipo de viagem apenas pode ser feito de Carro. O cliente pode ainda pedir um determinado serviço (por exemplo, garrafas de champanhe a consumir), que deve ser registado. O preço de uma viagem deste tipo é de 2.5€ por km mais o custo do serviço.

- Apresente as interfaces (pode optar pela utilização de um diagrama de classes) que considera necessárias para representar o problema apresentado (3 valores)
- Apresente uma classe `GestaoViagens` (não tem que implementar as interfaces anteriormente apresentadas) que realize a gestão de uma coleção de viagens e que possua os seguintes métodos:
 - Calcular o preço total a pagar numa determinada viagem. O método deverá retornar o preço a pagar, ou caso a viagem não exista, deverá lançar uma *checked exception*. (1,5 valores)
 - Disponibilizar um método que o número de viagens VIP realizadas até ao momento. (1,5 valores)