



Parte 2 - 12.5 valores

Considere as seguintes definições de classes de uma aplicação que implementa um sistema de gestão de parques de estacionamento (semelhante ao que temos na universidade com vários parques espalhados pelo campus). A classe **GesParques** possui a **informação dos parques** que gere e das **viaturas registadas**. A classe **GesParques** tem também acesso aos registos de todos os estacionamento feitos nos diversos parques que gere.

O modo de funcionamento é o seguinte: quando uma viatura entra num parque, sendo a entrada validada através de leitura óptica, é criado um registo novo, inicializando a hora de entrada e deixando a hora de saída por preencher. Quando a viatura sai do parque, existe um mecanismo de leitura óptica da matrícula e nessa altura a hora de saída é preenchida e é calculado o valor a pagar. Obviamente só poderão utilizar o parque viaturas registadas na **GesParques**.

Considere os seguintes excertos de código:

```
public class Parque implements Comparable<Parque>, Serializable {
    private String codParque;
    private String designacaoParque;
    private int lotacaoMax;
    private int ocupacao;
    private double precoPorHora;

    public double precoEstacionamento(Registo r) {...} // calcula o valor a pagar por um estacionamento
    ...
}

public class Viatura implements Comparable<Viatura> , Serializable {
    private String matricula;
    private String nomeDono;
    private String tipoViatura; // indica de que tipo é a viatura: carro, mota, etc.
    private Map<String, Registo> meusEstacionamentos;
}

public class Registo implements Comparable<Registo>, Serializable {
    private String numRegisto;
    private Parque parque;
    private Viatura carro;
    private LocalDateTime horaEntrada;
    private LocalDateTime horaSaida; // só deverá ser preenchida quando a viatura sai do parque

    public Registo(Parque parque, Viatura carro) {...} // cria um registo com a hora actual no parque indicado

    public double tempoNoParque() {...} // devolve o número de horas de estacionamento
}
```

Assuma, para as perguntas seguintes, que os métodos usuais (get, set, equals, clone, hashCode, ...) estão disponíveis, a menos que sejam solicitados, e responda às questões:



Questão 6

Efectue a declaração das variáveis de instância de **GesParques**, sabendo que esta deverá ter acesso aos parques, às viaturas e aos registos. **Justifique** a escolha das estruturas de dados que faz. Considere que a estratégia de associação entre a **GesParques** e os parques, viaturas e registos é de composição.

Codifique também o método `public void registaEntrada(String codParque, String matricula) throws ...`, da classe **GesParques**, que efectua a entrada de uma viatura num parque, verificando que ela esteja registada, que exista espaço de estacionamento e que o parque existe (não necessita de fazer o código das classes de excepção).

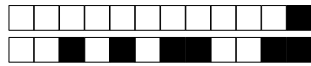
☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 *Reservado aos docentes*



Questão 7

Codifique o método `public Viatura maiorTempoEstacionamento()`, da classe `GesParques`, que determina a `Viatura` que esteve estacionada mais tempo nos diversos parques.

☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 *Reservado aos docentes*



Questão 8

Codifique o método `public void sairDoParque(String matricula) throws ...`, da classe `GesParques`, que efectua a saída de uma viatura do parque em que está estacionado. Note que deverá efectuar todas as mudanças necessárias à actualização do estado do sistema de gestão de parques, bem como lançar as potenciais excepções que possam ocorrer (não precisa de codificar as classes de excepção).

☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 *Reservado aos docentes*



Questão 9

Para a classe `GesParques` codifique o método `public boolean apenasNumParque(String matricula)`, que devolve `true` caso uma viatura só esteja estacionada ao mesmo tempo num único parque. Este método implementa um invariante de estado que verifica que não poderá acontecer que a mesma viatura esteja no momento actual estacionada em mais do que um parque - ou está estacionada num só parque ou então não está estacionada em nenhum.

☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 *Reservado aos docentes*



Questão 10

O preço por hora de cada um dos parques é neste momento fixo para todas as viaturas. Assuma agora que podemos ter também outros parques em que o preço é função do tipo da viatura (existindo assim um preço diferente para as viaturas ligeiras, para as motas, bicicletas, etc.)

Codifique a classe `ParqueComPrecoPorClasse`, declarando as suas variáveis de instância e codificando o método `public double precoEstacionamento(Registo r)`, que calcula o valor a pagar pelo estacionamento associado ao registo passado por parâmetro. O que seria necessário modificar no restante sistema para o compatibilizar com este novo tipo de parques?

☐ 0 ☐ .2 ☐ .4 ☐ .5 ☐ .6 ☐ .8 ☐ 1 *Reservado aos docentes*