

Exercícios práticos – aula 6

Objetos

1. Crie um sistema de gestão de alunos numa sala. Para tal, deverá criar as classes Room e Student. Cada sala tem a sua capacidade, o nome do bloco (ex: A, B, C, D) e o número da sala. Cada aluno terá o número de aluno, o nome e o curso. Deve ser possível realizar as seguintes operações:
 - a. adicionar alunos a uma sala (até ao limite da sua capacidade)
 - b. remover um aluno específico, com base no seu número
 - c. listar todos os alunos que estão na sala

2. Pretende-se desenvolver um programa que simula uma playlist de músicas. A cada música está associado o título e a duração da mesma. A playlist guarda uma lista de músicas e deverá permitir as seguintes operações:
 - a. mostrar a lista de músicas atualmente na playlist
 - b. acrescentar ou retirar músicas à playlist
 - c. calcular a duração total de todas as músicas contidas na playlist

3. Implemente a classe Calculator. O objeto calculadora deve ser capaz de realizar os cálculos matemáticos comuns de uma calculadora normal:
 - a. Soma
 - b. Subtração
 - c. Multiplicação
 - d. Divisão
 - e. Potência
 - f. Resto da divisão
 - g. Fórmula resolvente
 - h. A calculadora possui ainda um histórico das operações realizadas, armazenadas em formato String num vetor. Desenvolva as funções necessárias para:
 - i. guardar o histórico no vetor
 - ii. obter o histórico completo
 - iii. obter as últimas operações realizadas pela calculadora (histórico parcial)

4. Implemente a classe Car. Um carro possui algumas propriedades e características que variam de modelo para modelo, tais como a marca, o modelo, o número de lugares, a matrícula, o mês e ano de registo, o consumo em 100km, etc... Certos atributos devem ser indicados aquando a criação do carro.

Deve criar os testes que achar necessário para testar as funções desenvolvidas.

- a. Desenvolva os getters e os setters dos atributos que lhe pareçam adequados.
 - b. O carro possui um tanque de combustível, com uma capacidade atual e uma capacidade máxima. Implemente a função `encherDeposito()` que simula o abastecimento.
 - c. A função `run()` deve simular o percurso feito pelo automóvel em 1 km. O combustível associado a este percurso deve ser descontado na capacidade atual do tanque.
 - d. Deve ser possível determinar se o carro se encontra em funcionamento. Implemente a função `isLigado()` e outras que considera necessárias.
 - e. O automóvel possui um proprietário, que é representado pela classe `Person` (desenvolvida na aula). Crie as funções e atributos necessários que permitem o registo do proprietário à viatura.
 - f. Implemente a função `toString()` que deve devolver a informação que achar adequada.
5. Implemente a classe `CreditCard` cujas instâncias deverão ter um comportamento semelhante ao do conhecido cartão. Cada instância da classe `CreditCard` deverá possuir, ao ser criada, um titular, um número de 12 dígitos, um mês e ano de validade e um valor máximo de débito autorizado. Deve também guardar o montante despendido até ao momento e um histórico dos movimentos realizados. Por questões de simplicidade, considere que um movimento é uma `String` com o valor e uma descrição (ex: "30EUR - Bilhete de futebol"). A classe deve implementar os seguintes métodos:
- a. `int saldo()` - devolver o saldo do cartão (diferença entre o montante gasto e o limite de endividamento)
 - b. `void pagarCredito(int pag)` - efetuar um pagamento, isto é, abater o valor pag ao montante em dívida
 - c. `void gastar(int quantia, String descr)` - registar um movimento e atualizar o montante gasto
 - d. `String obterTalao()` - devolver a `String` que corresponde ao último movimento realizado
 - e. `String getMovimentos()` - devolver a lista de movimentos efetuados sob a forma de uma `String`