

1. Pretende-se uma aplicação para gerir os livros de uma biblioteca. Os livros são identificados por um código (um número inteiro que representa a ordem de criação do registo dos livros na biblioteca). O registo de um livro, para além do referido código, tem obrigatoriamente informação sobre o título e os autores.
  - a) Defina a classe `Livro` que representa este conceito de registo de um livro nesta biblioteca. Deve ser possível criar objectos da classe `Livro`, dando informação sobre o título e autores, sendo, neste caso o código gerado automaticamente como um inteiro que representa a ordem de criação do registo). Os membros variáveis devem ser privados, podendo ser acedidos para através de funções `get` e `set`. O membro `código` não deve poder ser modificado a partir do exterior da classe. Nesta classe deve implementar as funções:
    - i) `toString()` que retorna uma *string* com a descrição do objecto;
    - ii) `equals()` que representa o critério de identificação de um livro (dois livros são o mesmo se tiverem o mesmo código).
    - iii) `hashCode()` que retorna o hash code do objecto
  - b) Defina a classe `Biblioteca` que representa um conjunto de livros. Para além dos livros, a biblioteca tem um nome. Ao ser criado um objecto da classe `Biblioteca` deve ser dado o seu nome, ficando, à partida, sem registo de nenhum livro. Esta classe deve ter as funções:
    - i) `acrescentaLivro()` que recebe toda a informação que permite criar o registo de um livro, cria o registo e acrescenta-o à biblioteca;
    - ii) `pesquisaLivro()` que recebe o código de um livro e retorna uma referência para o livro se o encontrar ou `null` se não encontrar;
    - iii) `eliminaLivro()` que recebe o código do livro e o elimina se o encontrar; retorna o valor lógico do sucesso desta operação;
    - iv) `toString()` que retorna uma *string* com a descrição do objecto;
2. Depois de conhecer melhor esta biblioteca, verificou-se que a aplicação desenvolvida no exercício anterior precisa de ser expandida.
  - a) A biblioteca tem livros muito antigos, cuja manipulação só permitida aos funcionários. As pessoas que frequentam a biblioteca têm acesso apenas a reproduções destas obras antigas. Relativamente a estes livros, o registo tem ainda informação acerca de um número de cópias totais feitas. Defina a classe `LivroAntigo` que representa os livros antigos.
  - b) A biblioteca tem ainda livros actuais, cujo registo inclui o código ISBN e o preço, para além da informação considerada na classe `Livro`.

3. A operadora Telefonica oferece serviços de telecomunicações por telemóvel. No decorrer da sua actividade trabalha com conceitos tais como telemóveis, clientes, tarifas, saldos, carregamentos, chamadas, etc. A operadora Telefonica permite aos seus clientes a utilização de diversos tarifários com cartões recarregáveis.

A cada cartão Telefonica corresponde o número de telemóvel, o saldo e o conjunto de chamadas efectuadas. Cada chamada identifica-se pelo número do destinatário, pela duração da chamada (em segundos) e pelo seu custo.

É aos cartões que compete a funcionalidade de fazer carregamentos do saldo e de registar chamadas. Os cartões Telefonica aceitam sempre fazer chamadas se o saldo de que dispõem for positivo, rejeitando, se o saldo for negativo. A Telefonica não interrompe as chamadas, mesmo que o saldo disponível seja inferior ao custo da chamada, ficando, neste caso, com saldo negativo no final da chamada. O cartão deve também calcular o custo da chamada de acordo com o seu tarifário, registar as chamadas realizadas, e manter o saldo actualizado.

O custo da chamada é determinado pelo tarifário em vigor do cartão que efectua a chamada (não é feita distinção entre destinatários pertencentes/não-pertencentes à Telefonica). Os tarifários existentes na Telefonica são os seguintes:

- **PoucoTempo:** As chamadas têm um custo de 0,2 € por minuto.
- **Tagarela:** Este tarifário é pensado para quem gosta muito de conversar. O primeiro minuto é sempre cobrado por inteiro, a 0,5 € e os restantes a 0,02 €.

A Telefónica tem uma colecção de cartões que podem ser do tipo PoucoTempo ou Tagarela.

A interacção com a operadora Telefónica apresenta as seguintes operações:

- Acrescentar cartões lidos de ficheiro de texto
  - Carregamento de um cartão, dado o número e a quantia
  - Registar uma chamada, dado o número origem, destino e a duração em segundos
  - Ver o saldo de um cartão, dado o número
  - Imprimir factura detalhada para ficheiro de texto de um cartão, dado o número
  - Listar cartões por ordem natural
  - Listar cartões por ordem de números
  - Listar cartões por ordem de saldos.
4. Pretende-se um programa em Java para simulação da transmissão e permanência de um vírus numa população humana. O vírus pode ser transmitido pela proximidade de uma pessoa infectada. O espaço onde vive a população em estudo, o seu mundo, é representado por numa grelha bidimensional, em que as coordenadas têm valores inteiros.
- O tamanho do mundo é ilimitado. Cada pessoa ocupa uma posição no mundo, definida pelas suas coordenadas (abscissa, ordenada) que têm valores inteiros. Uma posição pode ser ocupada por mais do que uma pessoa.

O mundo é uma entidade que tem como função principal agregar as pessoas, e proporcionar os mecanismos necessários à sua interacção no contexto da simulação.

Os parâmetros que descrevem o mundo são o tempo que dura a imunidade duma pessoa à doença e o tempo de duração da infecção (depois deste tempo, a pessoa morre ou fica imune).

A simulação desenrola-se através de sucessão de instantes. Em cada instante, todos os elementos que estão colocados no mundo executam uma acção específica do tipo a que pertencem.

A cada novo instante de simulação o mundo manda agir todos os seus elementos e remove os elementos que morreram.

As pessoas podem ser saudáveis, estar infectadas ou estar imunes ao vírus. Podem morrer de doença.

Uma pessoa adquire a infecção se estiver na mesma posição de outra pessoa infectada.

As pessoas deslocam-se para posições adjacentes. Uma posição adjacente a  $(x, y)$  é uma das seguintes posições:  $(x-1, y)$ ,  $(x+1, y)$ ,  $(x, y-1)$ ,  $(x, y+1)$ .

A cada novo instante de simulação, a pessoa desloca-se, pode ser infectada e reage à infecção. A sua reacção à infecção depende do facto de ser jovem ou idoso.

Um jovem tem uma característica de resistência à infecção determinada por uma probabilidade. Se for infectado, terminado o tempo de duração da infecção, tem essa probabilidade de ficar imune. Se não ficar imune, morre. O tempo que fica imune é um parâmetro do mundo. Durante o período de imunidade nada o pode infectar. Depois desse tempo, fica em estado saudável, podendo ser novamente infectado.

Se um idoso for infectado, morre depois do tempo de duração da infecção.

Defina as classes que representam o mundo, os diversos tipos de pessoas e as operações envolvidas na actividade da simulação (passagem de cada instante).

5. Uma agência de publicidade faz a divulgação de um conjunto de restaurantes, dando informação acerca de diversos tipos de menus (quais as refeições e aonde são servidas). Informa, por exemplo, a um cliente de comida vegetariana, quais os restaurantes que oferecem este tipo de refeições e quais as refeições vegetarianas que constam do menu de cada um destes restaurantes.

Pretende-se uma aplicação que permita aos restaurantes registar na agência de publicidade menus de tipo vegetariano, dieta, italiano, *fast-food* e geral, para que constem da publicidade da referida agência.

Esta agência de publicidade tem um serviço de “Publicidade Gastronómica” representado pela classe `PublicidadeGastronomica`. Esta classe tem diversas colecções, uma para cada tipo de menu de que faz publicidade (colecção de restaurantes que oferecem menus do tipo vegetariano, do tipo de dieta, etc.). Nesta classe podem-se registar restaurantes com menus de tipo vegetariano, dieta, italiano, *fast-food* e geral. Para isso, basta invocar funções da classe `PublicidadeGastronomica` como `addVegetariano()`, `addDieta()`, `addItaliano()`, `addFastFood()` e `addGeral()`, respectivamente.

A função `main()` da classe que inicia a execução da aplicação cria a classe `PublicidadeGastronomica` e os restaurantes que utilizam os seus serviços publicitários.

```
public static void main(String[] args) {
    PublicidadeGastronomica publicidade = new PublicidadeGastronomica();
    new GaleriaDeSabores(publicidade);
    new BurgerKing(publicidade);
    new Italia(publicidade);
    String [] opcoes={"Vegetariano","Dieta","Italiano","Fast food","Geral", "Sair"};
    int opcao = 0;
    //...
    while (opcao != 6) {
        // ler opção
        switch (opcao){
            case 1: System.out.println(publicidade.divulgaVegetarianos());
                break;
            // etc...
        }
    }
```

As classes que representam os restaurantes requerem os serviços publicitários da classe `PublicidadeGastronomica` relativamente aos tipos de menus que oferecem.

O restaurante **Galeria de Sabores** pretende publicitar o seguinte:

Como menu vegetariano:

- Empadão de tofu
- Requeijão com molho de ervas e mel

Como menu italiano:

- Tagliateli a la carbonara
- Risoto de legumes
- Gelado de café

Como menu de dieta:

- Salada mediterrânica
- Salada niçoise
- Maçã assada

Como menu geral:

- Bifes de peru com molho de cenoura
- Bacalhau com natas
- Salmão grelhado
- Arroz doce

O restaurante **Burger king** pretende publicitar o seguinte:

Como menu *fast food*:

- FishBurger
- ChickenBurger
- Tarte de maçã com gelado

Como menu vegetariano:

- BeanBurger
- TofuBurger
- Sorvete de maçã

O restaurante **Itália** pretende publicitar o seguinte:

Como menu italiano:

Lasanha de dourada com Mozarella

Lasanha do mar

Tiramisu

Como menu geral:

Lulas recheadas com presunto

Leite creme