

Disciplina: Programação Orientada aos Objetos

Nome e número:

- Luís Seco – 2020211927
- António Campos - 2020217530



UNIVERSIDADE D
COIMBRA

Relatório do projeto final

Índice

Introdução.....	3
Estrutura do projeto.....	4
Funcionalidades implementadas.....	6
Conclusão.....	8

Introdução

Neste projeto final foi proposto a criação de uma aplicação de modo a gerir as empresas dos vários ramos da empresa StarThrive. Os ramos da empresa podem ser de restauração bem como de mercearias que por sua vez também possuem diferentes tipos de empresas. Através deste problema podemos aplicar os conceitos aprendidos ao longo do semestre na cadeira de programação orientada aos objetos, sendo eles como os conceitos de herança, que baseia se na aquisição de atributos e métodos de um objeto pai, e de polimorfismo que é um comportamento que pode variar de formas diferentes dependendo da classe em que esteja a ser usada. Além disso também nos é pedido diferentes tipos de funcionalidades para ser implementadas bem como a leitura de um ficheiro de texto com as empresas disponibilizadas de modo a permitir arrancar o programa sendo que depois teremos que passar a informação do ficheiro de texto para um ficheiro de objetos e passar a usá-lo em vez do ficheiro de texto.

Estrutura do projeto

De modo a nos organizar melhor e planejar a estrutura do projeto começámos por construir um diagrama de classes de modo a definir como iríamos construir os diferentes tipos de classes, que métodos e atributos iriam ter de modo a que quando procedêssemos à parte de programação ambos já tivéssemos tudo planeado de modo a que a construção do programa fosse mais fácil e estruturada.

- **Classe Empresa**

Começamos por definir a classe empresa que iria ser a classe pai para todas as outras empresas onde definimos os seus atributos bem como os seus métodos, a receita anual de cada empresa bem como a sua despesa, o lucro anual bem como as empresas que tem mais clientes. Esta classe possui 2 subclasses sendo estas a restauração e a mercearia. É uma classe abstrata uma vez que vai ter métodos que vão ser usados pelas suas subclasses e por isso a definição desses métodos ocorre logo na classe Empresa.

- **Classe Restauração**

Esta classe representa a classe pai para as empresas de café, pastelaria e restaurante, é uma classe abstrata pois não usa qualquer um dos métodos abstratos definidos na classe Empresa.

- **Classe Café**

Descende todos os atributos provenientes das classes Restauração e Empresa, sendo que também possui os seus próprios atributos. Usa os métodos definidos na classe Empresa de modo a obter o seu lucro, a sua despesa, as suas receitas e o seu número de clientes.

- **Classe Pastelaria**

Muito semelhante à classe café, descende os atributos das classes pai e usa os métodos da classe Empresa sendo que também tem atributos próprios.

- **Classe Restaurante**

Esta classe vai ser abstrata uma vez que vai ter duas subclasses que são dois tipos diferentes de restaurantes e por isso não vai usar nenhum método definido na classe empresa. Descende os atributos das suas superclasses e tem atributos próprios.

- **Classe RestauranteLocal e RestauranteFastFood**

Classe descendente da classe restaurante, tem os seus próprios atributos e descende os métodos e atributos das suas classes pai.

- **Classe Mercearia**

Descende da classe empresa e possui os atributos que vão caracterizar as empresas do ramo da mercearia, é uma classe abstrata pois não vai usar nenhuns dos métodos definidos na classe empresa.

- **Classe Frutaria e Mercado**

Descendem da classe Mercearia e possuem cada um os seus próprios atributos que caracterizam este tipo de empresas. Usam os métodos definidos na classe empresa.

- **Classe GereEmpresas**

Nesta classe começa-se por definir um arrayList que irá armazenar todas as empresas, é nesta classe que maior parte das comparações são feitas, tais como a empresa com mais lucro, maior despesa, maior receita, entre outras. Estas comparações são feitas à medida que se percorre o arrayList das empresas e por isso tem de ser definidas nesta classe. Também é aqui que lemos o ficheiro de texto que em cada linha do ficheiro possui uma empresa diferente sendo que no final desse ficheiro ser lido cada empresa é colocada na sua devida classe e de seguida adicionada à arrayList, logo depois é feita a escrita do conteúdo presente no ficheiro de texto para um ficheiro de objetos que a partir desse momento será o ficheiro a ser usado durante o programa para o funcionamento das diversas funcionalidades.

- **Classe Frame**

Classe que permite a interação gráfica com o utilizador de modo a ele poder realizar as diversas funcionalidades do programa.

- **Classe Projeto**

Classe que tem como objetivo o arranque do programa.

Funcionalidades implementadas

Para o funcionamento do programa havia diversas funcionalidades a ser implementadas, começando pelas mais básicas, listar as empresas, editar empresa, apagar empresa e criar empresa.

- **Listar empresas**

Para listar todas as empresas, simplesmente fizemos um print do arraylist que contém toda a informação necessária de cada empresa.

- **Editar empresa**

Para esta funcionalidade a nossa decisão foi caso o utilizador escolhesse a opção de editar, era-lhe pedido que introduzisse as coordenadas gps da empresa de modo a que consigamos identificar a empresa que o utilizador pretende editar, quando isso é feito remove-se a empresa do arraylist e adicionasse uma nova empresa com as informações que o utilizador pretende.

- **Apagar empresa**

Semelhante à funcionalidade anterior, pede-se ao utilizador as coordenadas gps da empresa que ele pretende apagar e depois removemos da arraylist.

- **Criar empresa**

Pergunta-se ao utilizador que tipo de empresa ele pretende criar, consoante a sua resposta é criado um novo objeto da classe correspondente à empresa que o utilizador escolheu e de seguida é pedido ao mesmo que preencha cada parâmetro que vai corresponder aos diferentes tipos de atributos que a classe possui. De seguida adicionamos a empresa à lista e atualizamos o ficheiro de objetos.

Além destas funcionalidades temos a de apresentar a empresa com maior receita anual, maior despesa anual, maior lucro e ainda apresentar as duas empresas de restauração com mais clientes.

- **Maior receita, despesa e lucro anual**

Para estas funcionalidades primeiramente criámos um método abstrato na classe empresa de modo a calcular a receita e despesa anual de cada empresa, depois de obtermos estes valores, fazemos a comparação percorrendo o arraylist e vemos para cada tipo de empresa qual é que teve maior receita, menor despesa e maior lucro.

- **Maior número de clientes nas empresas de restauração**

Criámos um método abstrato para devolver o número de clientes para cada classe, depois criámos um método na classe GereEmpresas que faz a comparação do número de clientes de cada empresa e devolve os dois maiores números de clientes, e fazemos print do nome dessas empresas.

Conclusão

Após a realização deste projeto podemos dizer que consolidamos os principais conceitos que esta cadeira nos propôs, sendo estes os conceitos de herança e polimorfismo bem como a manipulação de objetos. Além disso também consolidamos conceitos como a escrita e leitura de ficheiros sejam eles de texto ou de objetos. Também foi preciso o uso de conceitos já aprendidos em cadeiras passadas como, por exemplo, conceitos básicos de programação.

