



FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA

# Programação Orientada aos Objetos

2022/2023

Relatório do projeto

StarThrive – Gestor de Empresas

Trabalho realizado por:

Pedro Ramalho nº 2019248594

André Pinto nº 2021213497

## Índice

.....	1
Programação Orientada aos Objetos.....	1
2022/2023.....	1
Relatório do projeto.....	1
StarThrive – Gestor de Empresas.....	1
Trabalho realizado por:.....	1
Introdução.....	2
Descrição de Classes e Métodos mais importantes.....	3
Manual do utilizador.....	4
Conclusão.....	4
Referências.....	5

## Introdução

Este projeto foi realizado no âmbito da unidade curricular de Programação Orientada aos Objetos.

Este consistiu em criar um programa para permitir aos responsáveis da StarThrive gerir as suas diferentes empresas do ramo da restauração e mercearias. A linguagem de programação utilizada foi “Java”. O projeto foi realizado seguindo os princípios de programação orientada aos objetos, nomeadamente, os conceitos de herança e polimorfismo.

A interação com o utilizador é realizada através da “JFrames”, uma interface gráfica (GUI) e as alterações efetuadas pelo mesmo são guardadas através de ficheiros e ArrayList de objetos.

# Descrição de Classes e Métodos mais importantes

As classes estão organizadas conforme é exemplificado no diagrama UML que será entregue juntamente com este relatório. Não são colocados os métodos *“getters”* e *“setters”* de modo a diminuir a extensão do mesmo.

Entre os diversos métodos destacam -se os seguintes:

## *“Backend”:*

- `despesaAnual()`, `receitaAnual()`, `lucro()`, `retLucro()`, `alteraIndiceReceita()`, `devolveIndiceReceita()`, e similares para a despesa e lucro. Estes métodos são utilizados para cumprir com as exigências feitas no ponto 2 do enunciado.
- `capacidadeDeClientesDia()`, `alteraIndiceCapacidade1()`, `alteraIndiceCapacidade2()`, `devolveIndiceCapacidade1()` e `devolveIndiceCapacidade2()`. São os responsáveis pela implementação do ponto 3.
- `gereLeituraFicheiros()`, `leFicheiroObj()`, `leFicheirotxt()` e `escreveFicheiroObj()`. Carregam e alteram a `ArrayList` de empresas de modo a permitir a interação do utilizador com os dados.
- `toString()`. São essenciais para apresentar a informação adequada a cada tipo de empresa.

## *“Frontend (JFrames):”*

- `input()` invoca o menu, responsável por apresentar ao utilizador todas as opções propostas pelo enunciado.
- `caixatexto()` é essencial para criar as janelas que permitem a interação entre o utilizador e os métodos que manipulam o `ArrayList`.
- `jframeCondicoes()` e `jframeMaiorCapacidade()` são métodos que imprimem numa janela os métodos responsáveis pelos pontos 2 e 3 acima referidos.
- `listaEmpresas()` lista todas as empresas presentes na `ArrayList` de empresas apresentando-as numa tabela com a informação requerida no ponto 1.

# Manual do utilizador

1. O utilizador deverá começar por correr o programa na Class Main, no método Main(). Será aberta a janela “Bem-vindo à StarThrive” que contém o menu.
2. Para selecionar qualquer uma das opções disponíveis deverá clicar na opção desejada e de seguida carregar no botão selecionar. Cada opção fará o seguinte:
3. “Criar Empresa”-> deverá escolher qual o tipo de empresa que quer criar e de seguida preencher os campos solicitados.
4. “Listar Empresas”-> será apresentada uma tabela com a informação de todas as empresas a cargo da StarThrive.
5. “Editar Empresa”-> deverá escolher qual a empresa que deseja editar e de seguida preencher os campos pedidos.
6. “Melhores Empresas”-> apresenta para todos os tipos de empresas que a Starthrive gere: as empresas com maior receita anual, menor despesa anual e maior lucro anual.
7. “Empresas Restauração com maior capacidade de clientes por dia”-> apresenta a informação das duas empresas do tipo restauração que têm maior capacidade de clientes por dia.
8. “Apagar empresas”-> apaga todas as empresas a cargo da StarThrive.
9. Se quiser guardar as alterações que efetuou deverá clicar no botão guardar.
10. Se quiser sair sem guardar as alterações que efetuou deverá carregar no botão “x” no canto superior direito e em seguida carregar em “yes” para terminar sem gravar as alterações.

## Conclusão

A realização deste projeto permitiu-nos consolidar os conhecimentos adquiridos ao longo do semestre, também avaliados nos exercícios de avaliação anteriores, tais como o polimorfismo e o conceito de herança. Além disso levou-nos a desenvolver os conceitos adquiridos mais recentemente, nomeadamente a elaboração de diagramas UML, leitura e escrita de ficheiros txt e de objetos e o desenvolvimento de GUI.

O projeto foi submetido com todas as classes .java e o ficheiro de dados para teste, com este relatório, com o diagrama UML em pdf e com uma pasta zipada que contém todos os ficheiros JavaDoc.

# Referências

- “Slides” das aulas fornecidos no “uc student”.
- Exemplos de código mostrados pela professora nas aulas.
- <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
- <https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>
- <https://docs.oracle.com/javase/7/docs/api/java/nio/file/Files.html>