



C ·

FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Relatório de Projeto

Gestão de publicações CISUC

Programação Orientada a Objetos

Thomas Fresco | nº 2019219057

Índice

Introdução	1
Principais classes e métodos e seu funcionamento	2
Manual de Utilização	3
Conclusão	4
Referências Bibliográficas	5

Introdução

No âmbito da cadeira de POO (Programação Orientada a Objetos), foi proposto o desenvolvimento de um projeto final, com o intuito de ajudar os alunos a aprofundar e consolidar conhecimentos adquiridos sobre a linguagem de programação “Java”.

Suponha-se que o CISUC (Centro de Informática e Sistemas da Universidade de Coimbra) vai ser alvo de uma avaliação internacional muito importante. O principal objetivo da tarefa proposta é implementar uma aplicação que seja capaz de gerir e listar as publicações do CISUC, de acordo com vários critérios, de modo a dar resposta a todos os pedidos da equipa externa de avaliação.

Principais classes e métodos e seu funcionamento

As principais classes do programa são “Cisuc”, “Grupo”, “Investigador” e “Publicacao”. A sua constituição é a seguinte:

- **Classe “Grupo”:** classe que permite criar objetos “Grupo” que representam os grupos de investigação do Cisuc.
 - **Variáveis:** **nome (String)**, que representa o nome do grupo de investigação; **acronimo (String)**, que representa o acrónimo do grupo de investigação; **responsavel (Investigador)**, que representa o investigador responsável do grupo de investigação; **investigadores (ArrayList<Investigador>)**, que representa a lista de todos os investigadores membros do grupo de investigação.
 - **Principais métodos:** **publicacoesGrupo()**, que devolve um **ArrayList<Publicacao>** com todas as publicações do grupo; **publicacoesGrupo5Anos()**, que devolve um **ArrayList<Publicacao>** com todas as publicações do grupo dos últimos 5 anos; **getEfetivos()**, que devolve um **ArrayList<Efetivo>** com todos os investigadores efetivos do grupo; **getEstudantes()**, que devolve um **ArrayList<Estudante>** com todos os investigadores estudantes do grupo.
- **Classe “Investigador”:** classe abstrata que dá origem às subclasses “Efetivo” e “Estudante”, que representam os investigadores do Cisuc.
 - **Variáveis:** **nome (String)**, que representa o nome do investigador; **email (String)**, que representa o e-mail do investigador; **grupo (Grupo)**, que representa o grupo de investigação a que pertence o investigador; **publicacoes (ArrayList<Publicacao>)**, que representa a lista de todas as publicações das quais o investigador é autor.
 - **Principais métodos:** **tipo()**, que devolve uma **String** com o tipo de investigador (“efetivo” ou “estudante”, por herança e polimorfismo).
- **Classe “Publicacao”:** classe abstrata que dá origem às subclasses “Livro” (que por sua vez dá origem às subclasses “ArtigosConferencia” e “Capitulos”) e “Artigo” (class abstrata que por sua vez dá origem às subclasses “Conferencia” e “Revista”), que representam os tipos de publicação do Cisuc.
 - **Variáveis:** **titulo (String)**, que representa o título da publicação; **palavraChave (String)**, que representa a palavra-chave da publicação; **ano (int)**, que representa o ano de publicação da publicação; **audiencia (int)**, que representa a audiência da publicação; **resumo (String)**, que representa o resumo da publicação; **autores (ArrayList<Investigador>)**, que representa a lista dos autores da publicação.
 - **Principais métodos:** **tipo()**, que devolve uma **String** com o tipo de publicação (“Livro”, “Livro de Artigos de Conferência”, “Livro de Capítulos”, “Artigo de Conferência” ou “Artigo de Revista”, por herança e polimorfismo).
- **Classe “Cisuc”:** class principal, possui o método “main” e faz a gestão de toda a informação recolhida a partir dos ficheiros de entrada. Permite executar várias tarefas, através da interação com o utilizador, sob a forma de um menu.
 - **Variáveis:** **grupos (ArrayList<Grupo>)**, que representa a lista de todos os grupos de investigação do Cisuc.
 - **Principais métodos:** **menu()**, que imprime as instruções da interface de consola, possibilitando a interação com o utilizador; **listaPublicacoes()**, que devolve um **ArrayList<Publicacao>** com todas as publicações do Cisuc; **listaInvestigadores()**, que devolve um **ArrayList<Investigador>** com todos os investigadores do Cisuc; **organizar(ArrayList<Publicacao>)**, que organiza um **ArrayList<Publicacao>** por ano, tipo e fator de impacto e imprime-o; **recolherInfo()**, que recolhe a informação dos ficheiros de entrada; **guardar()**, que corre por último e guarda a informação manipulada num ficheiro de objetos.

Manual de Utilização

Quando o programa se inicia, começa por tentar ler informação de um ficheiro de objetos "Cisuc.obj". Caso este ficheiro não exista, o utilizador é informado com a mensagem "Ficheiro 'Cisuc.obj' não disponível.". De seguida, o programa tenta ler informação de 3 ficheiros de texto diferentes, "Grupos.txt", "Investigadores.txt", "Publicações.txt", que contêm, respetivamente, informação sobre os grupos de investigação, investigadores e publicações. Caso algum destes ficheiro não exista, o utilizador é devidamente informado. Na primeira vez que a aplicação é executada, aparecem as seguintes mensagens: "Grupos de investigação adicionados com SUCESSO.", "Investigadores adicionados com SUCESSO." e "Publicações adicionadas com SUCESSO.", resultantes da leitura dos ficheiros de texto. Após a primeira execução, já é apresentada a mensagem "Cisuc atualizado com SUCESSO.", resultante da leitura do ficheiro de objetos.

Uma vez adicionada a informação (ou não), a aplicação mostra, na consola, um menu para que o utilizador possa saber que operações é possível realizar. O menu apresentado tem um aspeto semelhante ao esquema seguinte:

```
*-----C-I-S-U-S-----*
| 1 - Indicadores gerais do CISUC |
| 2 - Publicações de um grupo de investigação |
| 3 - Membros de um grupo de investigação |
| 4 - Publicações de um investigador |
| 5 - Grupos de investigação |
| 0 - Sair |
*-----*
OPÇÃO:
```

Fig.1 - menu da aplicação

A partir deste momento, o menu é mostrado em "loop" e, se a programa tiver recolhido informação com sucesso, o programa executa determinados processos, consoante o que o utilizador inserir no campo "OPÇÃO:":

- O utilizador insere "1": o programa apresenta os indicadores gerais do CISUC (Total de membros, Número de membros de cada categoria, Total de publicações dos últimos 5 anos, Número de publicações de cada tipo);
- O utilizador insere "2": lista as publicações de um grupo de investigação (a escolher pelo utilizador), dos últimos 5 anos, organizadas por ano, por tipo de publicação e por fator de impacto;
- O utilizador insere "3": lista os membros de um grupo de investigação a escolher pelo utilizador), agrupados por categoria;
- O utilizador insere "4": lista as publicações de um investigador (a escolher pelo utilizador), agrupadas por ano, tipo de publicação e fator de impacto (critério de agrupamento a escolher pelo utilizador);
- O utilizador insere "5": listar todos os grupos de investigação, e apresenta informação sobre cada um (Total de membros, Número de membros de cada categoria, Total de publicações dos últimos 5 anos, Número de publicações, dos últimos 5 anos, agrupadas por ano, tipo de publicação e fator de impacto (critério de agrupamento a escolher pelo utilizador))
- O utilizador insere "0": o programa guarda a informação manipulada num ficheiro de objetos ("Cisuc.obj") e encerra (apenas encerra desta forma);
- O utilizador insere qualquer outra opção (inteiro ou String): mostra mensagem de erro e volta a imprimir o menu.

Conclusão

O trabalho realizado levou à concretização, com sucesso, da tarefa proposta, ou seja, à implementação uma aplicação capaz de gerir e listar as publicações do CISUC, de acordo com vários critérios, de modo a dar resposta a todos os pedidos de uma equipa externa de avaliação.

Pode-se concluir que foram atingidos os objetivos do projeto, na medida em que se aprofundaram conhecimentos relacionados com a linguagem de programação estudada na cadeira de Programação Orientada a Objetos. O trabalho realizado permitiu uma consolidação dos conhecimentos utilizados e preparou os executantes, de certo modo, para tarefas futuras no universo “Java”.

Referências Bibliográficas

- Del Nero R. Disponível a 16 de dezembro em: <https://www.infoworld.com/article/3323403/java-challengers-5-sorting-with-comparable-and-comparator-in-java.html>
- Karo I. Disponível a 20 de dezembro em: <https://stackoverflow.com/questions/2942857/how-to-convert-current-date-into-string-in-jav>
- Javapoint. Disponível a 20 de dezembro em: <https://www.javatpoint.com/java-get-current-date>
- Material de apoio da cadeira de POO (Programação Orientada a Objetos), disponibilizado pelos docentes responsáveis, disponível na plataforma “Inforestudante” (inforestudante.uc.pt)