

Projeto



Programação Orientada a Objetos

Miguel Pedroso 2019218176

Coimbra, dezembro de 2020

Índice

ÍNDICE	2
INTRODUÇÃO	3
DESCRIÇÃO DAS CLASSES / MÉTODOS MAIS RELEVANTES	3-4
MANUAL DE UTILIZADOR	5
DIAGRAMA DE CLASSES	6
CONCLUSÃO.....	7
REFERÊNCIAS	7

Introdução

Neste trabalho, pretende-se desenvolver uma aplicação em Java que permite gerir os grupos de investigação, os investigadores e as publicações do CISUC. Permite também listar todas as publicações de acordo com diversos critérios de pesquisa.

Descrição das classes / métodos mais relevantes

A classe Investigador tem os atributos nome, email, grupo, type e uma ArrayList de publicações. Tem duas subclasses: MembroEfetivo e Estudante. Para além dos atributos da superclasse, a classe Estudante tem os atributos tese, orientador e conclusão, e a classe MembroEfetivo tem os atributos numeroGabinete e tel.

A classe Publicacao tem os atributos título, tipo, ano, dim, fator e chave. Possui a função publicacoes5AnosOrdenar() que lista as publicações de um investigador ou grupo ordenadas por ano, tipo e fator. No caso do grupo, só lista publicações dos últimos 5 anos. Possui 5 subclasses: Livro, LivroConf, Revista, Conf e Capitulo, cada uma com atributos específicos.

A classe Grupo tem os atributos nome, acr, resp, uma ArrayList para as publicações do grupo e outra para os membros. Possui também vários atributos estáticos que servem para contar o número de obras e de investigadores de cada tipo. O método printPubYear() lista na consola todas as publicações nos últimos 5 anos do CISUC. O método printGrupoTipo() lista os membros de um grupo ordenados por tipo. Esta classe possui também métodos que permitem ao utilizador seleccionar um determinado grupo ou investigador.

A leitura e escrita de ficheiros é implementada numa só classe Ficheiros, que é constituída por oito métodos, usados para efetuar todas as operações que envolvam ficheiros:

- A função readFi() lê o ficheiro de texto que lhe é fornecido, linha a linha. Consoante o ficheiro que é (membros, obras ou grupos), esta função invoca uma das funções creator, fornecendo como argumento a linha que acabou de ler. Este processo repete-se para todas as linhas do ficheiro.
- Existem 3 funções destinadas a criar objetos: creatorGrupos(), creatorObras() e creatorInvestigadores(). Cada uma delas recebe como argumento uma linha sob a

forma de string e um int correspondente ao número dessa mesma linha no ficheiro a ser lido. Com a informação da linha, é criado um objeto que pode ser do tipo Grupo, Publicação ou Investigador.

- Para escrever os dados contidos no programa num ficheiro de objetos é usada a função `writeObj()`. Esta função passa todos os objetos do tipo Grupo contidos no programa para um ficheiro “dadosCISUC.obj”. Os objetos deste tipo possuem todos os dados necessários para o perfeito funcionamento do programa, não sendo por isso necessário escrever no ficheiro nenhum objeto do tipo Investigador ou Publicação.
- Para ler os dados contidos no ficheiro de objetos, é usada a função `readObj()`. Esta função lê todos os objetos do tipo Grupo contidos no ficheiro “dadosCISUC.obj”, e adiciona-os à `ArrayList` de grupos. Também invoca os métodos que permitem atualizar os diferentes contadores (Estudantes, Revistas, ...).

Manual de utilizador

1-Certifique-se que os ficheiros de texto com os dados se encontram na mesma diretoria que o ficheiro “Projeto.jar”.

2- Abra o Terminal (Mac) ou Command Prompt (Windows).

3-Escreva “cd” seguido de um espaço e arraste a pasta onde se encontra o “Projeto.jar” para a janela aberta.

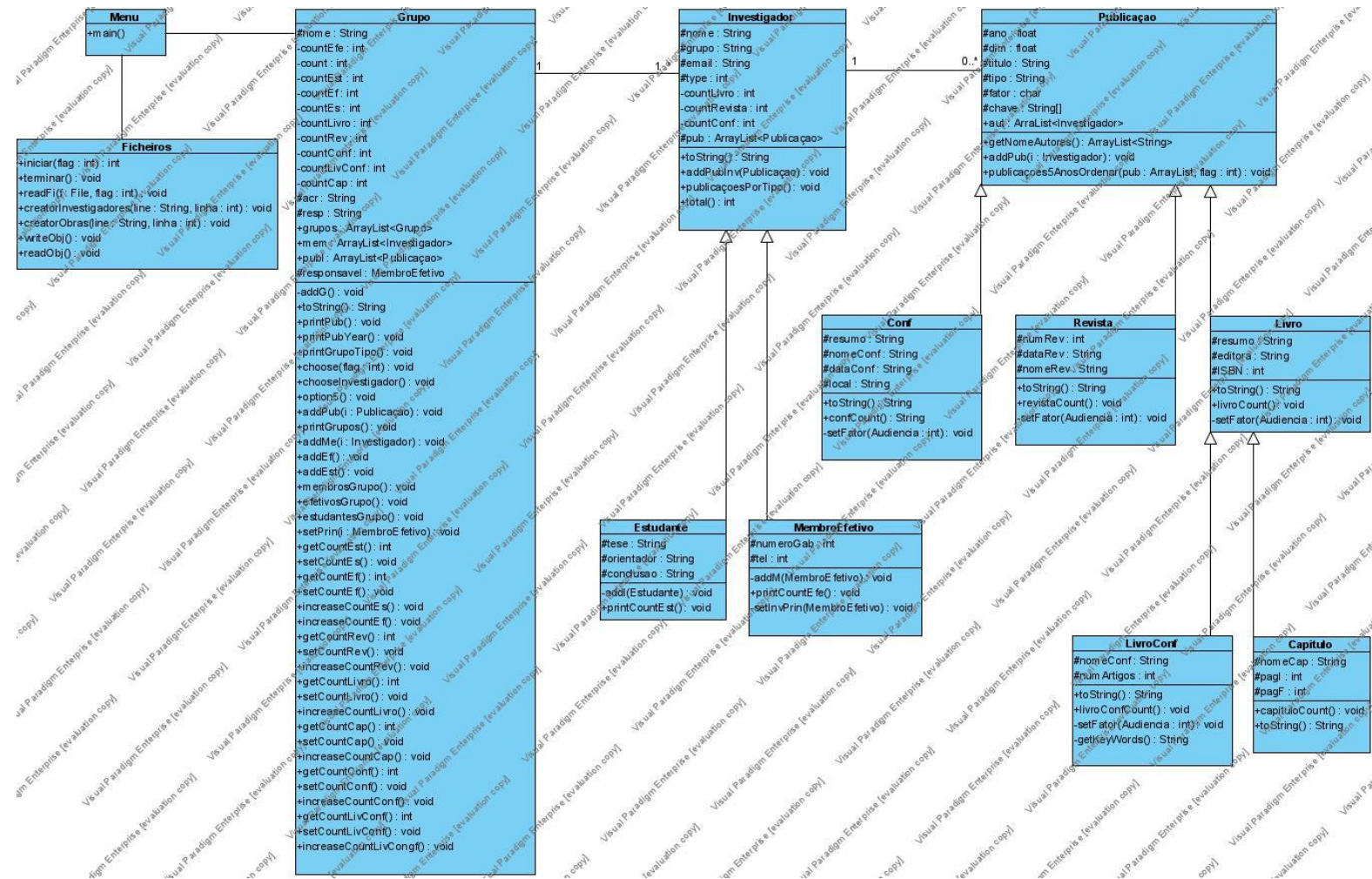
4- Escreva “Java -jar Projeto.jar” e prima Enter para iniciar o programa.

Ao iniciar o programa, **é perguntado ao utilizador se pretende que os dados sejam carregados para o programa a partir dos ficheiros de texto ou do ficheiro de objetos**. Esta escolha é feita para o caso de ter havido alterações nos dados dos ficheiros de texto, permitindo assim que os dados do programa as reflitam. No caso de o ficheiro de objetos não existir, os dados são carregados a partir dos ficheiros de texto, independentemente da escolha do utilizador.

De seguida, o utilizador **deve escolher uma das seis opções que lhe são apresentadas**:

- 1- Apresentar as informações gerais do CISUC
- 2- Apresentar as publicações de um grupo nos últimos 5 anos. Deverá introduzir o nome do grupo pretendido
- 3- Listar todos os membros de um grupo organizados por tipo. Deverá introduzir o nome do grupo pretendido
- 4- Mostrar todas as publicações de um investigador. Deverá introduzir o número correspondente ao investigador pretendido
- 5- Listar todos os grupos e, para cada um deles, o número de membros de cada tipo e as publicações do grupo nos últimos 5 anos
- 6- Sair

Diagrama de classes (UML)



Conclusão

Este projeto foi um bom desafio que me permitiu utilizar todas as competências que fui adquirindo ao longo deste semestre nas aulas de Programação Orientada aos Objetos. Ajudou a preparar-me para os diversos desafios que futuramente terei na minha vida profissional e a entender melhor os benefícios de uma programação orientada a objetos.

Referências

<https://www.visual-paradigm.com/tutorials/uml-class-diagram-in-diff-programming-languages.jsp>