



PROJETO DE PROGRAMAÇÃO ORIENTADA A OBJETOS

DEI 2020-2021

Trabalho desenvolvido por Edgar Filipe Ferreira Duarte
2019216077, edgarduarte@student.dei.uc.pt

Índice

1. Introdução	2
2. Funcionamento do Programa.....	3
3. Manual de utilizador.....	4
4. Diagrama de classes.....	5
5. Conclusão.....	6

1.Introdução

Este projeto foi realizado no âmbito da disciplina de POO (programação orientada a objetos). O projeto tem como objetivo criar uma plataforma do CISUC onde se pode encontrar diversas informações acerca dos seus investigadores e grupos de investigação, incluindo as suas publicações. Elaborou-se um programa que consegue, na consola, executar as seguintes funcionalidades:

- Mostrar ao utilizador informações gerais acerca do CISUC (número de membros, número de membros de cada categoria, número de publicações nos últimos 5 anos e número de publicações por categoria);
- Mostrar ao utilizador as publicações dos últimos 5 anos de um grupo de investigação, organizadas por ano, tipo de publicação e fator de impacto.
- Mostrar ao utilizador os membros de um certo grupo de investigação, agrupados por tipo;
- Mostrar as publicações de um certo investigador, organizadas por ano, tipo de publicação e fator de impacto;
- Mostrar ao utilizador informações acerca de cada grupo de investigação (número de membros, número de membros de cada categoria, número de publicações nos últimos 5 anos e número de publicações agrupados por ano, tipo de publicação e fator de impacto.

Para implementar estas funcionalidades recorreu-se a programação orientada a objetos ao, a partir de 13 classes, criar um sistema de comunicação entre as várias classes de forma simples e eficaz.

2.Funcionamento do programa

Como foi referido na introdução, o programa consiste de 13 classes interligadas que comunicam umas com as outras. As classes com maior importância são:

- *CisucManager* (controlo das operações, é a classe que manda o programa fazer o que o utilizador quer);
- *Researcher* (representa um investigador, tem duas classes filhas (*EffectiveMember* e *Student*));
- *ResearchGroup* (representa um grupo de investigação);
- *Publication* (representa uma publicação e tem várias classes filho);
- *CISUC* (classe onde estão a maior parte dos métodos que envolvem interligação entre publicações, investigadores e grupos de investigação);

As restantes classes são, na sua maioria, classes filho de uma das classes acima referida, sendo a sua função fazer mais específico a classe pai.

O programa começa no *CisucManager* onde, primeiramente, procura um ficheiro de objetos com formato de um objeto *CISUC*. Um objeto *CISUC* é composto por 3 listas que contêm todos os investigadores, todos os grupos de investigação e todas as publicações. Se o ficheiro de objetos existir, é carregado o objeto e prossegue-se para o menu, onde o utilizador poderá fazer as suas escolhas. Se o ficheiro binário não existir, cria-se o objeto *CISUC* a partir de 3 ficheiros de texto (um para os grupos de investigação, um para os investigadores e um para as publicações, carregados nesta ordem). Após criado o objeto, guardamos este objeto num ficheiro de objetos para, numa posterior utilização do programa, saltar-se esta etapa. É de notar que ao longo da criação e leitura dos ficheiros existem várias defesas contra erros nos ficheiros. Quando existe um elemento com informações erradas num ficheiro, regra geral, o elemento é ignorado.

Agora o programa está no método *menu()*. Aqui é onde se faz a interação com o utilizador. Ele pode escolher uma das várias funcionalidades descritas na introdução, podendo sair do programa a qualquer momento. Em qualquer das funcionalidades será preciso comunicação entre investigadores, grupos de investigação e publicações. Ora essa comunicação é realizada, na sua maioria, na classe *CISUC*. Visto que é a classe que contém todos os investigadores, grupos de investigação e publicações, é bastante simples interligar estes três elementos. Agora, a partir dos diversos métodos criados nesta classe, fazem-se as operações necessárias para a execução de cada funcionalidade. Ocasionalmente poderá ser necessário à classe *CISUC* chamar métodos como o *toString()*, *gets* e outros métodos das outras classes.

3.Manual de utilizador

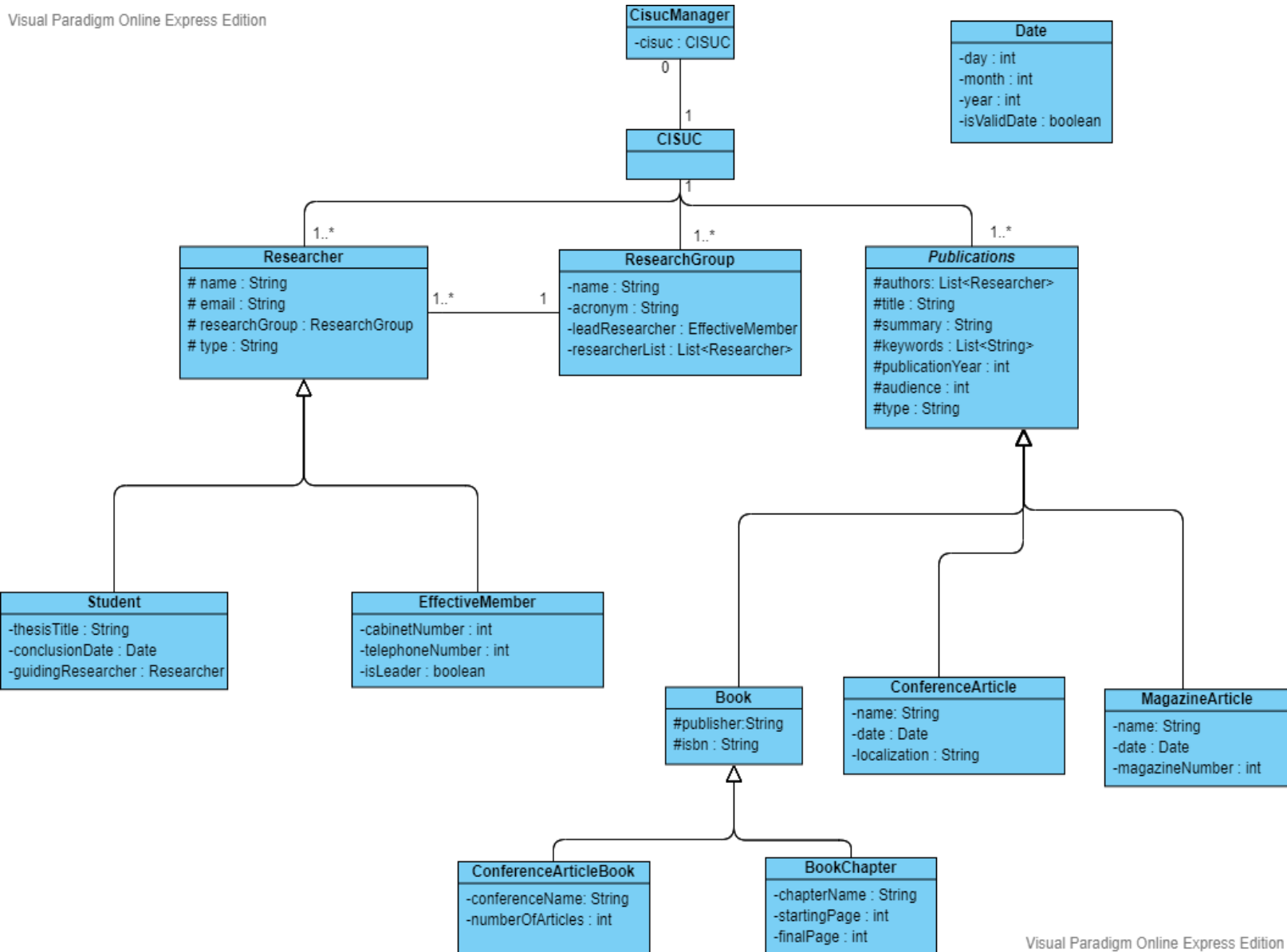
Para executar o programa siga as seguintes etapas:

- Primeiramente o utilizador deve abrir a consola do seu computador, escrevendo na barra de pesquisa do seu computador “*cmd*” e selecione a opção que aparece.
- Seguidamente deverá aceder à pasta onde está o executável na consola. Para este passo, primeiramente copie o *path* (o local onde está guardado o ficheiro) e de seguida escreva na consola escreva “*cd*” e dê *paste* do *path* previamente copiado e depois pressione “*Enter*”.
- Agora basta ativar o executável. Para este passo escreva na consola “*java -jar* “ e à frente escreva o nome do executável. Nota: não se esqueça de colocar “.jar” no fim do nome do executável. Depois de escrito pressione “*Enter*”.
- Parabéns! Atenção que os ficheiros de leitura têm de estar no mesmo *path* que o executável, senão o programa vai dar erro e para com uma mensagem a dizer que o ficheiro não existe. Está agora em condições de utilizar o programa. Escolha uma das opções de 1 a 7 cada uma com uma funcionalidade diferente (7 é para sair do programa). Ocasionalmente poderá ter de escolher um grupo de investigação ou um investigador para as funcionalidades. Nesses casos se não souber nenhum investigador/ grupo de investigação mas mesmo assim quiser ver como o programa corre escreva “*Help*” aquando da pedida dos nomes e terá a lista de opções possíveis.

4.Diagrama de classes

O diagrama de classes simplificado (não contem métodos) para este programa é o seguinte:

Visual Paradigm Online Express Edition



Visual Paradigm Online Express Edition

Devido a problemas de espaço não foi possível mandar o diagrama de classes completo. Esse diagrama está disponível no *zip* do projeto final.

5.Conclusão

Todas as funcionalidades pedidas pelo enunciado são implementadas pelo programa. O programa também implementa várias barreiras de segurança na leitura dos ficheiros. Assim considera-se concluída a criação do programa. Como trabalho futuro sugere-se a implementação de uma nova classe para ler os ficheiros, em vez de se fazer esse passo na classe *CisucManager* e sugere-se uma estruturação diferente dos ficheiros de texto para permitir escrevê-los mais facilmente.