U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Algoritmos e Estruturas de Dados

Gestão de uma Biblioteca

Época Especial

Luís Miguel Guimas Marques

201104354

25 de maio de 2020

Descrição do Problema

No âmbito da unidade curricular Algoritmos e Estruturas de Dados, foi proposto o desenvolvimento de um programa para gestão de uma biblioteca.

O principal foco do programa desenvolvido é a gestão de empréstimos efetuados pela biblioteca. Para suportar esta funcionalidade também faz parte a gestão de livros, funcionários e leitores.

Cada empréstimo tem de ter um leitor e um funcionário associado. O leitor não pode ter mais do que três empréstimos em simultâneo.

Por cada dia que passa, para além do máximo estipulado pela biblioteca, o leitor incumpre numa multa de 0.25Eur por dia.

Dos funcionários pertencentes à biblioteca fazem parte supervisores e um administrador. Cada supervisor tem à sua responsabilidade um ou mais funcionários. Um supervisor não pode ser responsável por outro supervisor. O número de funcionários à responsabilidade de cada supervisor tem de ser equilibrado.

Os livros não emprestados devem fazer parte de uma árvore binária. A ordenação é efetuada por defeito pelo seu id e também deve ser possível a ordenação por ano, título e autores.

Os leitores inativos devem fazer parte de uma tabela de dispersão. Um leitor é considerado inativo quando não há qualquer registo de um empréstimo, ou o seu último empréstimo seja superior a 12 meses.

Para melhorar a gestão de empréstimos também é implementada uma fila de espera para os pedidos feitos pelos leitores. Estes pedidos devem fazer parte de uma fila de prioridade que deve respeitar os seguintes critérios, data do pedido e o tipo de leitor.

A prioridade a respeitar deve ser a data seguido de um leitor que padece de alguma deficiência seguido pelos leitores com idade igual ou inferior a 12 anos e por fim os restantes.

Descrição da Solução

A solução implementada foi criar uma classe biblioteca em que armazena a informação relativa à mesma, que é o caso dos livros em que os livros não emprestados são guardados numa árvore de pesquisa binária, permitindo assim uma pesquisa mais eficaz e por diversas ordenações.

Um leitor que queira reservar um livro e caso esse livro não tenha exemplares disponíveis é-lhe proposto fazer um pedido. Esse pedido é guardado numa lista de prioridade ordenada por data seguido do tipo de leitor, sendo o mais prioritário os leitores que tenham alguma deficiência seguido por crianças com 12 anos ou menos e por último os restantes. Assim a pessoa que seja a primeira da fila é notificada assim que um exemplar esteja disponível.

Os leitores inativos aqueles que não tenham um empréstimo em mais de 12 meses são guardados numa tabela de dispersão.

Para gerir por completo a biblioteca e os seus funcionários foi adicionado um administrador que faz a gestão de todos os funcionários, sendo que ele não é encarregue de fazer empréstimos ou pedidos. Essas tarefas estão destinadas para os funcionários e supervisores.

Quando a aplicação é aberta é pedido um login, caso o utilizador não faça login apenas lhe é possível ver os livros existentes na biblioteca.

Estrutura de ficheiros

Para dar apoio ao programa implementado existem vários ficheiros .txt que guardam a informação de várias classes importantes para o funcionamento do mesmo.

Assim que o programa começa, a informação contida nos ficheiros .txt que se encontram no diretório files é lida e preenchida todos os containers.

Durante a execução do programa, o utilizador pode fazer alterações a várias estruturas dependendo da sua posição na biblioteca. Um utilizador que faça login no sistema pode ocupar três posições, sendo elas de Administrador, Supervisor e Funcionário. Apenas estes três cargos permitem fazer operações de CRUD.

Quando o utilizador termina de usar o programa é questionado se pretende guardar a informação que possa ter sido alterada durante a sua utilização.

Existe um ficheiro que guarda as credenciais de administrador e outros que guardam vários supervisores, funcionários, leitores, livros, empréstimos e pedidos.

A classe Library tem um método que faz o carregamento de todos os ficheiros e no fim do programa, se o utilizador assim o pretender, a classe Library chama o método saveFiles e procede à escrita de toda a informação para os ficheiros.

Lista de ficheiros:

- admin.txt: id ; username ; password
- books.txt: id ; titulo ; ISBN ; autores ; nº de páginas ; nº de exemplares ; ano
- borrows.txt: id ; id do livro ; id do leitor ; id do funcionário ; data do empréstimo
- employees.txt: id ; nome ; password
- readers.txt: id ; nome ; nº de telemóvel ; email ; localidade ; tipo de utilizador ; lista de empréstimos constituída pelos os ids ; data do ultimo empréstimo
- requests.txt: id ; id do livro ; id do leitor ; id do funcionário ; data
- supervisors.txt: id ; nome ; password ; lista de funcionários constituída pelos os seus ids

Lista de Funcionalidades

App

Classe responsável pelas interações que o utilizador tem com o programa. É onde são mostrados os menus e as várias opções. Assim que o utilizador faz login a classe guarda a informação do utilizador de maneira a que todas as operações feitas ficam registadas com o seu id.

Library

Classe onde fica guardada toda a informação. É aqui que grande parte da lógica do programa se encontra. É responsável elas as operações de CRUD.

É constituída por várias estruturas de dados como vetores onde são guardados os funcionários, livros e leitores, uma BST que organiza os livros não emprestados e uma *hash table* para os leitores inativos.

Employee

Classe que tem contém a informação de todos os funcionários e tem como classes derivadas Supervisor e Admin.

Reader

Classe que tem como membros dados a informação de cada leitor.

Book

Classe que tem com membros dados a informação de cada livro e uma *priority queue* com os pedidos feitos pelos leitores.

Borrow

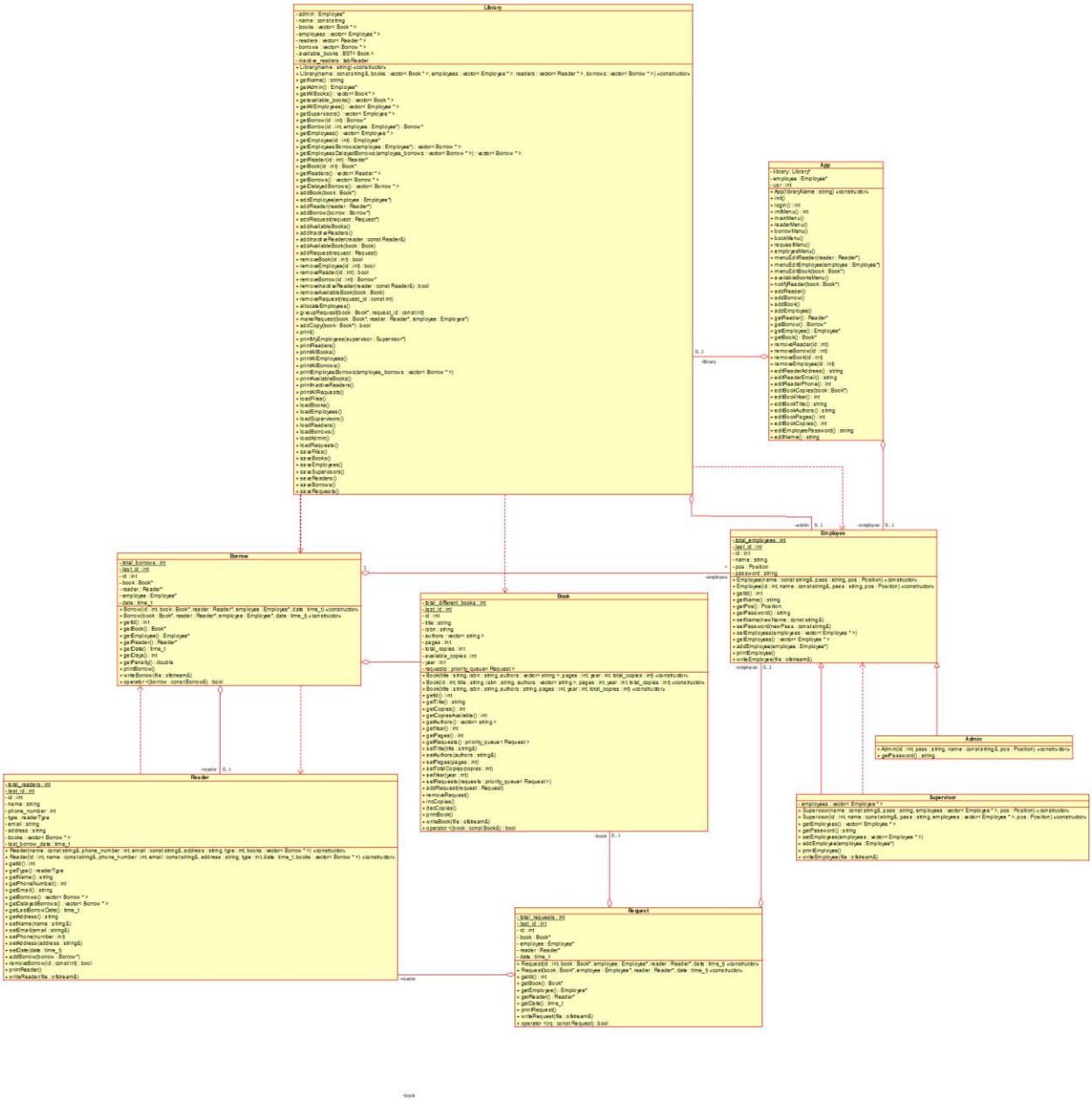
Classe que tem como membros dados a informação do empréstimo.

Request

Classe que tem como membros dados a informação do pedido.

UML

Nota: Imagem mais detalhada incluída à parte.



Destaque de funcionalidade

Quando um empréstimo é feito, e caso não haja exemplares disponíveis, o leitor é questionado se pretende fazer um pedido para esse livro.

Quando um exemplar é adicionado, e caso exista algum pedido, o leitor no topo da lista é notificado se pretende fazer o empréstimo. Caso ele responda “não” o pedido é removido e caso a fila ainda tenha pedidos, o próximo leitor é notificado.

Principais dificuldades

O principal obstáculo ocorreu ao implementar certas estruturas de dados como a BST. Até ao momento usava sempre *pointers* para representar os membros dados de cada classe. A BST ao não usar *pointers* gerava conflitos com os *includes* que usava em cada classe. Após uma revisão mais pormenorizada esses conflitos foram colmatados.

Como correr o programa

Durante o desenvolvimento foi usado o compilador MinGW.

Para compilar foi usado o comando “g++ -Wall *.cpp -o library.exe” dentro da pasta do *src*.

O programa faz uso dos ficheiros .txt que se encontram na pasta *files*. Não é recomendado mudar o nome desses mesmos ficheiros.