

Programação Orientada a Objetos

Trabalho Prático 2:

Nomes:

Alessandro Azevedo Duarte

Caio Conti Guidote Ribeiro

Matrículas:

2017072642

2017072677

Introdução

O trabalho prático foi criar um sistema de gerenciamento de uma biblioteca.

Como compilar e executar:

O programa foi feito em Windows 10 na IDE Code::Blocks.

Implementação do Código:

Header files:

Foram utilizadas as bibliotecas `iostream`, `ctime`, `vector`, `string`. Foram criados Headers para as classes, Empréstimo, Biblioteca, Interface, Exceções, Livro, Periódico e um `Us_pub` que contém as classes Usuário, Date e Publicação

Classes:

Como funciona:

O programa começa carregando uma interface que se inicia com várias opções sendo que elas são selecionadas através de letras maiúsculas.

Para adicionar um novo usuário é necessário preencher os dados solicitados e esses dados são armazenados em um vetor.

O mesmo acontece ao adicionar livros ou publicações e um novo empréstimo.

Um novo empréstimo sempre se inicia sem itens, que são adicionados através de outra função, utilizando de seus códigos que são únicos e adicionados pelo usuário.

Por meio desses códigos o usuário consegue também, excluir os itens, empréstimos e também usuários, além de, devolver itens em separado de um empréstimo ou todos os itens de uma só vez.

De acordo com o título do livro ou autor da publicação é possível fazer uma busca por esse item .

Por fim temos funções para listar todos os usuários, publicações e os empréstimos e seus itens.

A última opção do menu sai do programa e temos uma parte para salvar arquivos e lê-los, através dos vetores de usuários, publicações e empréstimos, não se encontra no menu, mas que também são funcionais.

Usuário:

A classe usuário contém os atributos e funções pedidas, além disso contém getters para o nome e cpf e uma função newPenalização que faz contas da penalização e atualiza a data de penalização.

Publicação:

Além dos atributos e funções pedidas contém getters retornando os atributos e também uma função virtual para fazer o ligamento com a classe Livro.

Data:

Como solicitado pela descrição do trabalho foi criado um tipo Data com um construtor padrão, um construtor que seta a data do sistema e um construtor com dados inseridos pelo usuário.

Tendo os atributos de dia, mês e ano como privados, criamos getters para ter acesso a esses dados.

Com a utilização do tipo principalmente na parte dos empréstimos e penalizações foi necessário fazer sobrecarga dos operadores = , <<, +, - e <. Para que pudéssemos comparar datas, imprimir-las, somá-la com inteiros e subtraí-las.

Livro:

É uma classe derivada de função, usando uma função virtual, com as funções e atributos solicitados. Colocamos também uma sobrecarga do == e foi necessário um get para retornar a quantidade de exemplares e para os autores.

Periódico:

Classe derivada de publicação, com o construtor e a impressão.

Empréstimo:

Contém os atributos e funções desejadas e também uma função que procura se existe um livro no vetor de livros retornando a posição que se encontra, também com funções gets retornando o número do empréstimo e uma que retorna o vetor de itens. Os argumentos que são passados do tipo livro são como ponteiro, para poder haver o dynamic cast corretamente.

Biblioteca:

A classe biblioteca é a principal conexão entre a Interface do programa e as outras classes e funcionalidades, utilizando os dados inseridos pelo usuário nas funções necessárias para fazer o gerenciamento dos usuários, publicações e empréstimos.

Ela contém 3 vetores principais, um de usuários, um vetor de ponteiros para as publicações, para que possa ser feito o dynamic_cast e um vetor de empréstimos, através da biblioteca pode-se preencher esses vetores, fazer buscas, devolver itens, deletar itens, além de salvar os dados desse vetor em um arquivo e conseguir ler esses dados posteriormente.

Interface:

A Interface além das funções pedidas cria uma biblioteca B que é a que será manipulada no programa. O void menu que apenas imprime as opções que podem ser utilizadas na interface. A função de casos faz um switch case que o usuário entra com a letra maiúscula e aperta enter para selecionar a opção desejada. A função Menu_main roda os casos até o usuário escolher sair do programa.

Conclusão

Um dos principais problemas encontrados foi em relação ao uso do dynamic cast relacionado ao vetor de publicações que continha livros e periódicos, dessa forma, foi utilizado um vetor de ponteiros para tais itens.

Em relação a possíveis melhoras, podem acontecer erros em situações não previstas.

O trabalho se mostrou muito interessante, pois foi abordada a matéria toda estudada, além de que o paradigma de orientação a objetos é o mais utilizado no mundo, e de grande ajuda para programas complexos e desenvolvimento de softwares em conjunto.