UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
INSTITUTO METRÓPOLE DIGITAL
CURSO DE BTI - ENGENHARIA DE SOFTWARE

RELATÓRIO DO PROJETO 3

STL, bibliotecas e tratamento de exceções

LABORATÓRIO DE LINGUAGENS DE PROGRAMAÇÃO I

Dionísio Dias Aires de Carvalho
Airton Correia de Oliveira Neto

NATAL

JUNHO / 2017

Dionísio Dias Aires de Carvalho
Airton Correia de Oliveira Neto

RELATÓRIO DO PROJETO 3

Relatório apresentado à disciplina de Linguagens de Programação I, correspondente ao laboratório 10 sobre a implementação de STL, bibliotecas, tratamento de exceções em C++ do curso de Bacharel em Tecnologia da Informação (BTI) da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. Dr. Sílvio Sampaio e Prof. Dr. Everton Cavalcante.**

JUNHO / 2017

NATAL

JUNHO / 2017

RESUMO

Projeto de implementação de uma biblioteca sobre uma loja de Conveniência organizados em um namespace próprio. A biblioteca contém classes como Fornecedores, Produtos, Perecíveis, bem como, funções de venda e listagem dos produtos. Para englobar todas estas funcionalidades, foi criado um namespace qlevetudo. A biblioteca também de dispõe de tratamento para exceções. Também foi implementado um “exportarmain.cpp” para realizar a exportação de dados da QleveTudo.

SUMÁRIO

1. INTRODUÇÃO

Uma das principais razões para o uso de bibliotecas é a reutilização de código, permitindo que outros programas utilizem partes de códigos comuns. Para isto, foi apresentado em sala de aula um modelo de criação e implementação destes. Para aplicar isto, o Projeto 3 exigiu que fosse criada uma biblioteca contendo as funções de venda, listagem, criação/remoção/modificação dos produtos de uma loja de conveniência, utilizando os conteúdos que foram vistas ao longo do semestre nas disciplinas IMD0029 - Estruturas de Dados Básicas I e IMD0030 - Linguagem de programação I ou equivalentes.

2. DESENVOLVIMENTO

Para implementar a solução solicitada, foram criados os arquivos “bancodaods.cpp, fornecedor.cpp, funcoes.cpp, npereciveis.cpp, pereciveis.cpp produto.cpp, venda.cpp” contendo os “#includes” para todos os outros arquivos de definição (classes e funções). Cada arquivo também recebeu a definição de namespace “qlevetudo”.

As implementações foram muito semelhantes ao Projeto 2, diferindo no uso do Map, que foi implementado em substiituição a Lista, de forma que o Indice indica o codigo do produto. A outra grande diferença foi que o namespace para geração das bibliotecas.

Foi implementado um programa de exportação para que os dados dos produtos junto com seus exportadores pudessem ser exportados para um arquivo de acordo com 4 argumentos passados através da linha de comando ao executar o programa.

Um pequeno arquivo makefile foi escrito com o propósito de gerar a biblioteca descrita na versão dinâmica (para Windows e Linux). Este script também gera um programa que testa as funcionalidades da biblioteca (também com versões Windows e Linux).

3. DIFICULDADES

Para a implementação das funções genéricas, o uso do “extern 'C'” foi utilizado (conforme exposto em sala) porém, isso ocasionava erro de compilação visto que o 'C' não reconhece templates. Após uma pesquisa nas referências bibliográficas, o comando mudou para “extern 'C++'”.

4. ERROS

Foi implementado um “erros.h” para o tratamento de exceções, que foram solucionadas através de “logic_error” e “invalid_argument” da biblioteca <stdexcept>.

Exceções essas que foram:

Erro na manipulação de arquivo, argumento invalido passado na linha de comando e ausência de nome do arquivo que conterá os dados exportados.

1. CONCLUSÃO

Para testar todas as funcionalidades, criou-se um pequeno programa que testas todas as funcionalidades da biblioteca criada. O programa, utilizando-se das funcionalidades da biblioteca, executa as seguintes funções:

- Cadastrar um fornecedor:
  - Cria um novo dado do tipo Fornecedor
  - Pede ao usuário as informações por linha de comando;
- Remover um fornecedor:
  - Pede que o usuário selecione o fornecedor por CNPJ
  - Busca o Fornecedor e o apaga
- Listar os fornecedores;
- Cadastrar um produto;
  - Cria um dado do tipo Produto
  - Pede o tipo do do produto
  - Pergunta ao usuário as informações do produto
- Remover um produto;
- Alterar um produto;
  - Pede ao usuário que selecione o fornecedor
  - Pede qual produto daquele fornecedor
  - Pede ao usuário se ele quer manter cada informação original ou alterá-la
- Listar todos os produtos por fornecedor;
- Listar todos os produtos de um fornecedor;
- Listar produtos por tipo;
- Listar produtos por codigo;
- Realizar uma venda;
- Controle de estoque;

Portanto, o projeto foi concluído gerando arquivos de bibliotecas dinâmica e binários que utilizam estas bibliotecas. A documentação sobre todo o código foi gerada através do Doxygen e está na pasta “doc”.

REFERÊNCIAS BIBLIOGRÁFICAS

Fórum Stack Overflow. Disponível em:
https://stackoverflow.com/questions/24229814/how-does-extern-c-work/24229844
Acesso em 18 de junho de 2017.

Site C++ Reference. Disponível em:
http://en.cppreference.com/w/cpp/language/class_template
Acesso em 18 de junho de 2017.