МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ

ВИКОНАВ

студент академічної групи КБ-23

Ліннікова Софія

ПЕРЕВІРИЛА

викладач кафедри кібербезпеки

та програмного забезпечення

Ганна ДРЄЄВА

Кропивницький – 2024

**ТЕМА: РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ**

**МЕТА:** Набути ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**ЗАВДАННЯ:**

1. У складі команди ІТ-проєкту розробити програмні модулі оброблення динамічної структури даних.

2. Реалізувати програмний засіб на основі розроблених командою ІТ-проєкту модулів.

**Завдання на розроблення ПЗ:**

1. Створити е-довідник кодів товарів згідно з Українською класифікацією товарів зовнішньоекономічної діяльності (номер розділу, назва розділу, група/код/ товару, назва товару).
2. За вибором користувача застосунок забезпечує:

* пошук запису в е-довіднику за введеним кодом товару. зберігання е-довідника у заданий текстовий файл;
* додавання нового запису в е-довідник;
* вилучення заданого запису з е-довідника;
* завершення роботи програми з автоматичним записом даних у файл.

**Лістинг структури:**

#ifndef STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

#include <iostream>

using namespace std;

struct Edirectory{

int directoryPart;

string directoryPartName;

int groupArticle;

int numberArticle;

string articleName;

};

#endif // STRUCT\_TYPE\_PROJECT\_2\_H\_INCLUDED

**Лістинг модуля prj\_3\_Zaharchenko.h:**

#include "ZakharchenkoModules.h"

void printEntryByProductNumber(string filename, int productNumber) {

ifstream infile(filename);

if (!infile.is\_open()) {

cout << "Error opening file: " << filename << endl;

return;

}

string line;

bool found = false;

while (getline(infile, line)) {

if (line.find("Номер продукту: " + to\_string(productNumber)) != string::npos) {

cout << line << endl;

found = true;

break;

}

}

infile.close();

if (!found) {

cout << "Запису " << productNumber << " не знайдено." << endl;

}

}

void copyDirectoryToFile(string sourceFilename, string targetFilename) {

ifstream sourceFile(sourceFilename);

if (!sourceFile.is\_open()) {

cout << "Помилка відкриття файлу: " << sourceFilename << endl;

return;

}

ofstream targetFile(targetFilename);

if (!targetFile.is\_open()) {

cout << "Помилка відкриття файлу: " << targetFilename << endl;

sourceFile.close();

return;

}

string line;

while (getline(sourceFile, line)) {

targetFile << line << endl;

}

sourceFile.close();

targetFile.close();

}

**Лістинг модуля prj\_3\_Linnikova.h:**

#include "LinnikovaModules.h"

void writeToFile(string filename, int directoryPart, string directoryPartName, int groupArticle, int numberArticle, string articleName){

ofstream outfile(filename, ios::app);

if (!outfile) {

cout << "Помилка запису: " << filename << endl;

return;

}

outfile << "Розділ: "<< directoryPart << ","

<< directoryPartName << ","

"Группа "<< groupArticle << ","

"Номер продукту: "<< numberArticle << ","

"Назва продукту: " << articleName << endl;

}

void deleteEntryByProductNumber(string filename, int productNumber) {

ifstream infile(filename);

if (!infile.is\_open()) {

cout << "Помилка відкриття: " << filename << endl;

return;

}

vector<string> lines;

string line;

while (getline(infile, line)) {

if (line.find("Номер продукту: " + to\_string(productNumber)) == string::npos) {

lines.push\_back(line);

}

}

infile.close();

ofstream outfile(filename);

if (!outfile.is\_open()) {

cout << "Error opening file for writing: " << filename << endl;

return;

}

for (const auto& l : lines) {

outfile << l << endl;

}

}

**Лістинг модуля prj\_3\_Gonchar.h:**

#include <iostream>

#include "ModulesGonchar.h"

#include "LinnikovaModules.h"

#include "ZakharchenkoModules.h"

#include "struct\_type\_project\_2.h"

using namespace std;

int main()

{

string filename = "edirectory.txt";

createFileIfNotExists(filename);

string data = startup(filename);

cout << data << endl;

int action;

while (true){

cout<<"1 - пошук запису за введеним кодом товару\n2 - зберігання довідника у заданий текстовий файл\n3 - додавання нового запису в довідник\n4 - вилучення заданого запису\n5 - завершення роботи"<< endl;

cin >> action;

switch (action){

case 1: {

int productNumber;

cout << "Введіть номер продукта для пошуку: ";

cin >> productNumber;

printEntryByProductNumber(filename, productNumber);

break;

}

case 2: {

string targetFilename;

cout << "Введіть назву файлу для зберігання довідника: ";

cin >> targetFilename;

createFileIfNotExists(targetFilename);

copyDirectoryToFile(filename, targetFilename);

cout << "Довідник збережено в файл " << targetFilename << endl;

break;

}

case 3: {

Edirectory entry;

cout << "Введіть номер розділу: ";

cin >> entry.directoryPart;

cout << "Введіть назву розділу: ";

cin >> entry.directoryPartName;

cout << "Введіть номер группи: ";

cin >> entry.groupArticle;

cout << "Введіть номер товару: ";

cin >> entry.numberArticle;

cout << "Введіть назву товару: ";

cin >> entry.articleName;

writeToFile(filename, entry.directoryPart, entry.directoryPartName, entry.groupArticle, entry.numberArticle, entry.articleName);

break;

}

case 4:{

int productNumber;

cout << "Введіть номер продукта для видалення: ";

cin >> productNumber;

deleteEntryByProductNumber(filename, productNumber);

break;

}

case 5:{

return 0;

}

}

}

return 0;

}

**Лістинг програмного засобу:**

#include <iostream>

#include "ModulesGonchar.h"

#include "LinnikovaModules.h"

#include "ZakharchenkoModules.h"

#include "struct\_type\_project\_2.h"

using namespace std;

int main()

{

string filename = "edirectory.txt";

createFileIfNotExists(filename);

string data = startup(filename);

cout << data << endl;

int action;

while (true){

cout<<"1 - пошук запису за введеним кодом товару\n2 - зберігання довідника у заданий текстовий файл\n3 - додавання нового запису в довідник\n4 - вилучення заданого запису\n5 - завершення роботи"<< endl;

cin >> action;

switch (action){

case 1: {

int productNumber;

cout << "Введіть номер продукта для пошуку: ";

cin >> productNumber;

printEntryByProductNumber(filename, productNumber);

break;

}

case 2: {

string targetFilename;

cout << "Введіть назву файлу для зберігання довідника: ";

cin >> targetFilename;

createFileIfNotExists(targetFilename);

copyDirectoryToFile(filename, targetFilename);

cout << "Довідник збережено в файл " << targetFilename << endl;

break;

}

case 3: {

Edirectory entry;

cout << "Введіть номер розділу: ";

cin >> entry.directoryPart;

cout << "Введіть назву розділу: ";

cin >> entry.directoryPartName;

cout << "Введіть номер группи: ";

cin >> entry.groupArticle;

cout << "Введіть номер товару: ";

cin >> entry.numberArticle;

cout << "Введіть назву товару: ";

cin >> entry.articleName;

writeToFile(filename, entry.directoryPart, entry.directoryPartName, entry.groupArticle, entry.numberArticle, entry.articleName);

break;

}

case 4:{

int productNumber;

cout << "Введіть номер продукта для видалення: ";

cin >> productNumber;

deleteEntryByProductNumber(filename, productNumber);

break;

}

case 5:{

return 0;

}

}

}

return 0;

}

**Висновок.** У ході виконання лабораторної роботи, мені довелося працювати з іншими студентами для виконання спільного проєкту. Під час роботи ми разом визначали архітектуру програми, аналізували поставлені задачі та шукали кращі проєктні рішення. У даному коді я реалізувала простий інтерфейс для управління довідником товарів. Користувач має можливість додавати нові записи до довідника, вказуючи номер розділу, назву розділу, код групи товару та назву товару. Також користувач може видаляти записи, вказавши код групи товару для вилучення.

Код вміло працює з файлами у форматі CSV, зчитуючи дані з файлу "products.txt" при запуску програми і зберігаючи зміни у файлі "output.txt".

Кожен крок взаємодії з користувачем забезпечується зрозумілими повідомленнями про помилки або успішне виконання операцій.

Цей код може бути корисним як основа для подальшого розвитку програм, які вимагають управління даними з використанням текстових файлів. Він демонструє простий, але ефективний підхід до роботи зі структурованими даними.