

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 6**

з дисципліни “Основи програмування”

тема “Формат Даних XML”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент I курсу  групи КП-92  Мовчан Максим  варіант №9 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Використати наслідування класів, віртуальні функції та абстрактний клас.  
Опанувати формат даних XML та навчитись його зчитувати і формувати.  
Використати поліморфізм для заміни способу роботи з файловим сховищем даних.

**Загальні вимоги до завдання**

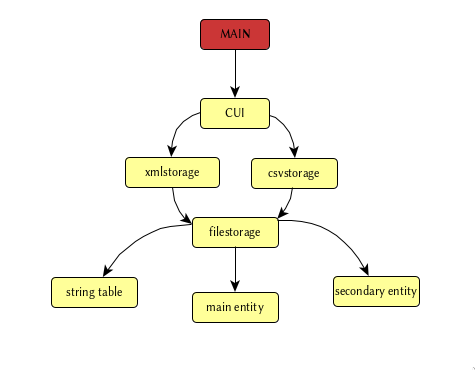
Додати до консольної програми з попередньої лабораторної роботи модуль сховища для роботи з файлами даних формату XML так, щоби це ніяк не вплинуло на код і функціональність модуля консольного інтерфейсу.

**Методичні вказівки до виконання**

Кроки виконання:

1. Qt проект:
   1. Скопіювати у директорію завдання Qt проект з лабораторної роботи №5 і переназвати файл налаштувань проект на lab6.pro.
   2. Відкрити файл налаштувань проекту (lab6.pro) і задати версію C++17.
   3. Перенести CSV файли з директорії data у data/csv.
2. Абстрактний клас і наслідування.
   1. Виконати інструкції щодо модифікації класів сховищ.
   2. Змінити спосіб створення сховища з використанням upcasting (**див. Додаток A**).
3. Сховище даних XML:
   1. Створити у проекті директорію data/xml та додати у неї два XML файли (з розширенням .xml).
   2. Використати будь-який онлайн сервіс для перетворення з CSV в XML і заповнити створені файли даними основних та додаткових сутностей (приклад **див. Додаток B**).
   3. Створити клас XML сховища відповіно до інструкцій.
   4. Реалізувати XmlStorage з використанням Qt XML класів:
      1. Додати у файл налаштувань проекту (.pro) QT += xml як перший рядок .
      2. Підключити у модулі xml\_storage заголовочний файл #include <QtXml>
   5. При створенні об'єкта сховища замінити його тип на XmlStorage, а шлях до файлу на ../lab6/data/xml (**див. Додаток C**).

**Діаграма** **класів**

****

**Інтерфейси**

|  |
| --- |
| **xmlstorage.hpp** |
| **#ifndef XMLSTORAGE\_HPP**  **#define XMLSTORAGE\_HPP**  **#include "filestorage.hpp"**  **class XmlStorage: public FileStorage {**  **vector<Organisation> loadOrgs();**  **void saveOrgs(const vector<Organisation>& orgs);**  **int getNewOrgId();**  **vector<Founder> loadFndrs();**  **void saveFndrs(const vector<Founder>& fndrs);**  **int getNewFndrId();**  **public:**  **explicit XmlStorage(const string& dir\_name\_ = ""): FileStorage(dir\_name\_) {}**  **};**  **#endif // XMLSTORAGE\_HPP** |

|  |
| --- |
| **csvstorage.hpp** |
| **#ifndef CSVSTORAGE\_HPP**  **#define CSVSTORAGE\_HPP**  **#include "filestorage.hpp"**  **#include <csvlab.hpp>**  **class CsvStorage: public FileStorage {**  **vector<Organisation> loadOrgs();**  **void saveOrgs(const vector<Organisation>& orgs);**  **int getNewOrgId();**  **vector<Founder> loadFndrs();**  **void saveFndrs(const vector<Founder>& orgs);**  **int getNewFndrId();**  **public:**  **explicit CsvStorage(const string& dir\_name\_ = ""): FileStorage(dir\_name\_) {}**  **};**  **#endif // CSVSTORAGE\_HPP** |

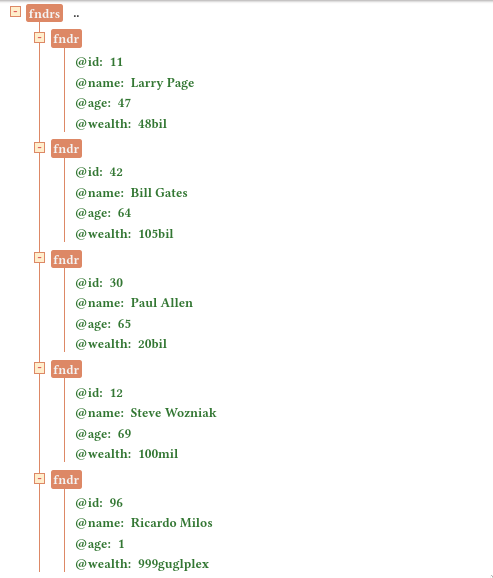
|  |
| --- |
| **filestorage.hpp** |
| **#ifndef FILESTORAGE\_H**  **#define FILESTORAGE\_H**  **#include "founder.hpp"**  **#include "organisation.hpp"**  **#include "string\_table.hpp"**  **#include <fstream>**  **#include <optional>**  **#include <vector>**  **using namespace std;**  **class FileStorage {**  **string dir\_name\_;**  **protected:**  **fstream orgs\_file\_;**  **fstream fndrs\_file\_;**  **virtual vector<Organisation> loadOrgs() = 0;**  **virtual void saveOrgs(const vector<Organisation>& orgs) = 0;**  **virtual int getNewOrgId() = 0;**  **virtual vector<Founder> loadFndrs() = 0;**  **virtual void saveFndrs(const vector<Founder>& fndrs) = 0;**  **virtual int getNewFndrId() = 0;**  **public:**  **explicit FileStorage(const string& dir\_name = "");**  **virtual ~FileStorage() {}**  **void setName(const string& dir\_name);**  **string name() const;**  **bool isOpen() const;**  **bool open(bool type = 0);**  **void close();**  **vector<Organisation> getAllOrgs();**  **optional<Organisation> getOrgById(int org\_id);**  **bool updateOrg(const Organisation& org);**  **bool removeOrg(int org\_id);**  **int insertOrg(const Organisation& org);**  **// courses**  **vector<Founder> getAllFndrs(void);**  **optional<Founder> getFndrById(int fndr\_id);**  **bool updateFndr(const Founder& fndr);**  **bool removeFndr(int fndr\_id);**  **int insertFndr(const Founder& fndr);**  **};**  **#endif // FILESTORAGE\_H** |

**Дані**

|  |
| --- |
| **Founders.xml** |
| **<?xml version="1.0"?>**  **<fndrs>**  **<fndr id="11" name="Larry Page" age="47" wealth="48bil"></fndr>**  **<fndr id="42" name="Bill Gates" age="64" wealth="105bil"></fndr>**  **<fndr id="30" name="Paul Allen" age="65" wealth="20bil"></fndr>**  **<fndr id="12" name="Steve Wozniak" age="69" wealth="100mil"></fndr>**  **<fndr id="96" name="Ricardo Milos" age="1" wealth="999guglplex"></fndr>**  **</fndrs>** |

|  |
| --- |
| **Orgs.xml** |
| **<?xml version="1.0"?>**  **<orgs>**  **<org id="50" country="US" label="Google inc." founders="Larry Page, Sergey Brin"></org>**  **<org id="30" country="UK" label="EPAM Systems" founders="Arkadiy Dobkin, Leo Lozner"></org>**  **<org id="23" country="US" label="Microsoft Corporation" founders="Bill Gates, Paul Allen"></org>**  **<org id="91" country="UK" label="Apple Inc." founders="Steve Jobs, Steve Wozniak, Ronald Wayne"></org>**  **<org id="69" country="UG" label="FlexAir" founders="Ricardo Milos"></org>**  **</orgs>** |

**Дерева**



**Висновки**

Було винесено частину реалізації обробки csv у окремий клас. Ми також створили окремий клас для роботи з xml файлами. Ми використовували вбудований в qt функціонал для зчитування його і форматування. Було використано Cmake/Qmake для збірки проекту з багатьох вихідних файлів. Ми розбили проект на модулі та створили власні заголовочні файли. Компіляція всього коду відбувалась за допомогою компілятора g++.