

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 № 5**

з дисципліни “Основи програмування”

тема “Статична програмна бібліотека”

|  |  |  |
| --- | --- | --- |
| 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**

Винести частину коду консольної програми у проект статичної бібліотеки.

Навчитись користуватись Qt Creator і створювати у ньому консольні проекти та проекти статичних бібліотек

**Загальні вимоги до завдання**

Створити програму з Консольним Інтерфейсом Користувача, що взаємодіє з Файловим Сховищем Даних за допомогою власної програмної бібліьотеки формату даних CSV.

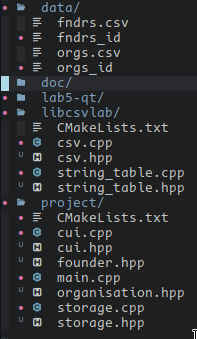
Додати у реалізацію файлового сховища даних (FileStorage) роботу із новим типом сутностей за варіантом.

Розширити консольний інтерфейс користувача (CUI) меню для роботи з сутностями нового типу.

**Методичні вказівки до виконання**

1. Додаткова сутність (ExtraEntity):
   1. Тип даних. Створити у окремому модулі структуру даних для ExtraEntity з декількома полями (різного типу), серед яких обов'язково має бути поле id - ідентифікатор сутності: унікальне ціле число.
   2. Дані. Створити у директорії data новий CSV файл (приклад, courses.csv), у якому перечислити дані всіх екземплярів цього додаткового типу даних (приклад, Course), де значення стовпця id довільні (але унікальні).
   3. Сховище. Реалізувати другу версію класу Storage, що включає поля і методи для роботи з об'єктами додаткового типу даних.
   4. CUI. Додати до CUI головне меню, у якому користувач може зробити вибір з яким типом сутності він хоче працювати (основна чи додаткова). Додати підменю для роботи з додатковими сутностями аналогічно існуючим.
2. CMake проекти:
   1. Статична бібліотека:
      1. Створити у корені завдання директорію з назвою libcsvlab.
      2. Перенести модуль csv та його залежності (модуль string\_table) у директорію libcsvlab.
      3. Створити у директорії libcsvlab файл налаштувань CMake проекту для збірки статичної бібліотеки libcsvlab.
      4. Бібліотеку не встановлювати в систему.
   2. Головний проект:
      1. Створити у корені завдання директорію з назвою project.
      2. Перенести всі інші модулі консольної програми у директорію project.
      3. Налаштувати у проекті головної програми шляхи підключення заголовочних файлів і архіву статичної бібліотеки libcsvlab.
3. Qt проекти (відео інструкція):
   1. Обов'язково додати у .gitignore в корені репозиторія додаткові рядки ігнорування для Qt.
   2. Встановити Qt i Qt Creator і відкрити Qt Creator.
   3. Створити у Qt Creator проект типу Qt Subdirs, який розмістити всередині кореня директорії завдання.
   4. Додати у кореневий проект проекти консольного додатку (project) та статичної бібліотеки (csvlab).
   5. Підключити статичну бібліотеку до консольного додатку та скопіювати весь код проектів та файли з CSV даними.
   6. Змінити шлях до сховища даних використовуючи назву проекту, наприклад ../lab5/data.
   7. Це пояснюється тим, що Qt за замовчуванням збирає executable файл програми у директорії, сусідній директорії з проектом.
   8. Тепер проект має збиратись, а зібрана програма запускатись і працювати аналогічно початковому рішенню.

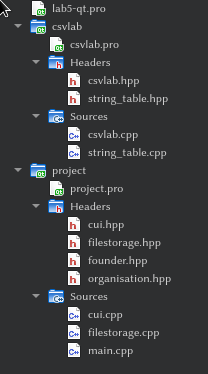
**Файлова структура**



|  |
| --- |
| **CMakeLists.txt(libcsvlab)** |
| cmake\_minimum\_required(VERSION 3.16.4)  project (csvlab)  include\_directories(.)  file(GLOB CPP\_SOURCES "\*.cpp")  add\_library(${PROJECT\_NAME} STATIC ${CPP\_SOURCES}) |

|  |
| --- |
| **CMakeLists.txt(project)** |
| cmake\_minimum\_required(VERSION 3.16.4)  set(CMAKE\_CXX\_STANDARD 17)  set(CMAKE\_CXX\_STANDARD\_REQUIRED ON)  project(lab5)  include\_directories(../libcsvlab)  link\_directories(../libcsvlab/build)  file(GLOB CPP\_SOURCES "\*.cpp")  add\_executable(${PROJECT\_NAME}.out ${CPP\_SOURCES})  target\_link\_libraries(${PROJECT\_NAME} csvlab) |

**Структура Qt проектів**

****

**fndrs.csv**

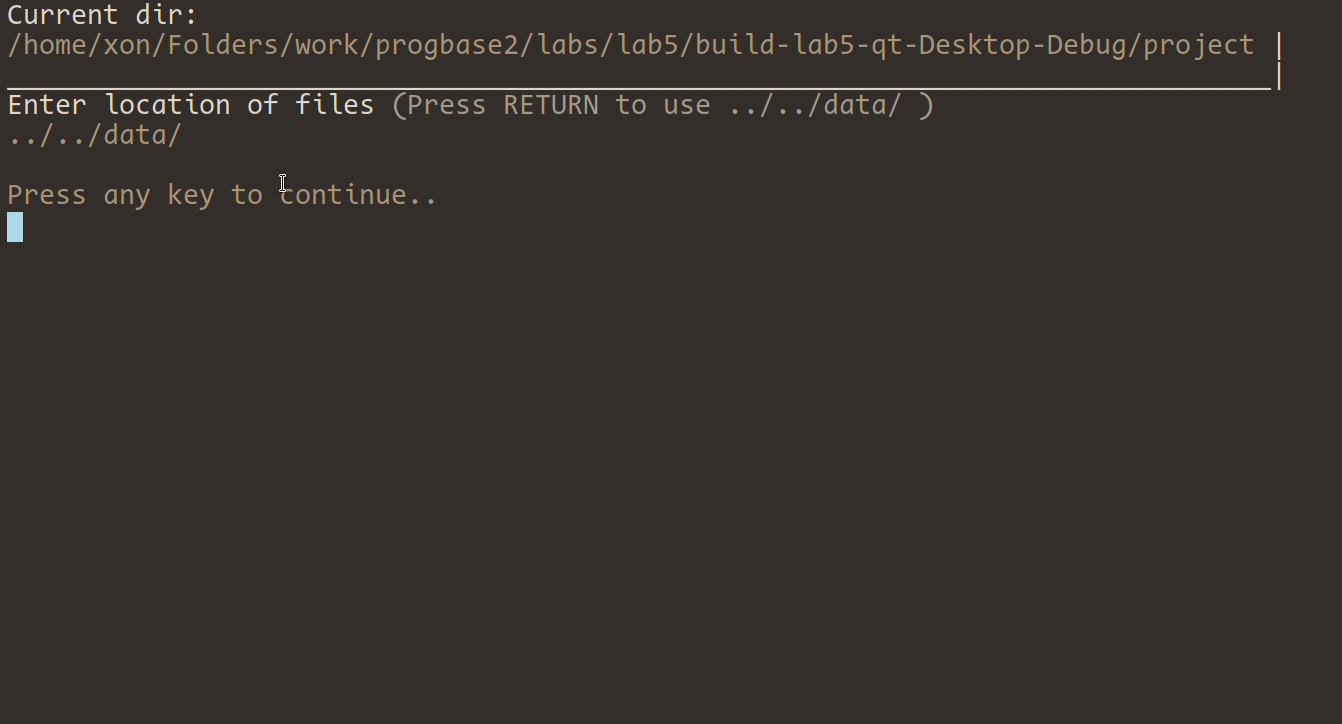
|  |  |  |  |
| --- | --- | --- | --- |
| id | name | age | wealth |
| 11 | Larry Page | 47 | 48bil |
| 42 | Bill Gates | 64 | 105bil |
| 30 | Paul Allen | 65 | 20bil |
| 12 | Steve Wozniak | 69 | 100mil |
| 96 | Ricardo Milos | 1 | 999guglplex |

**Текст коду програми**

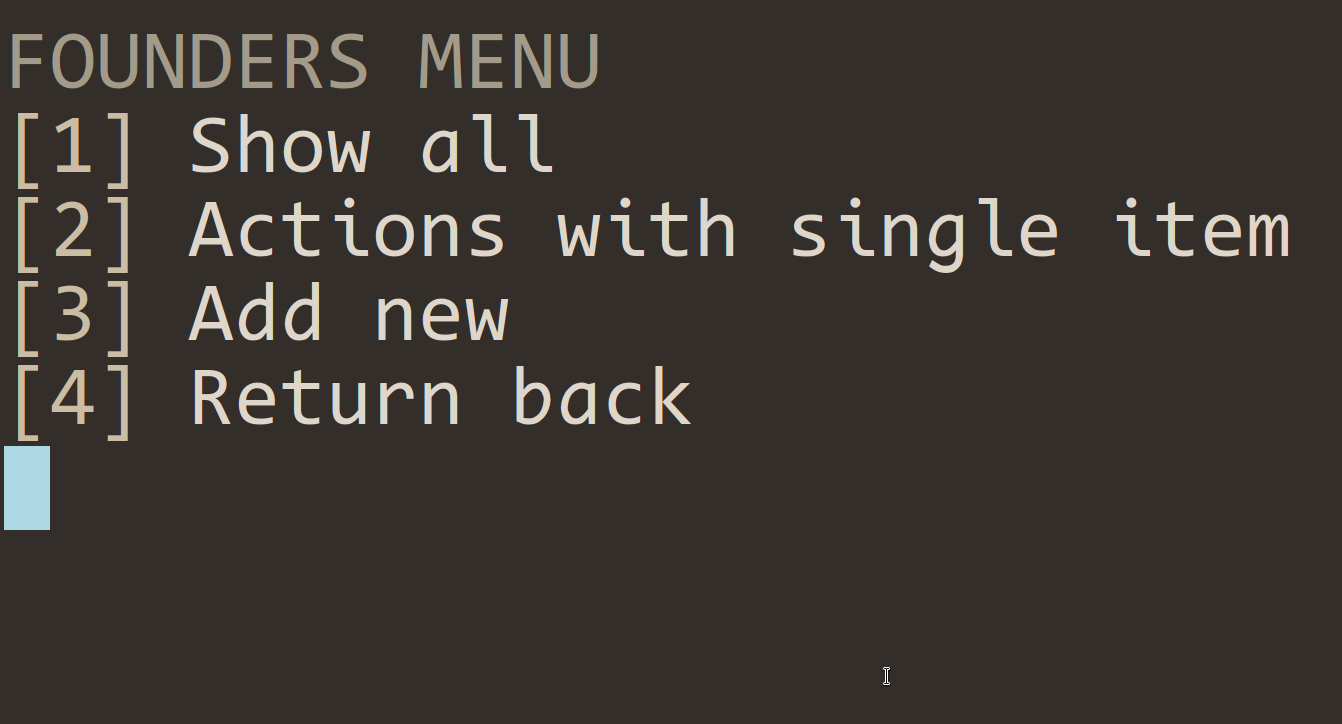
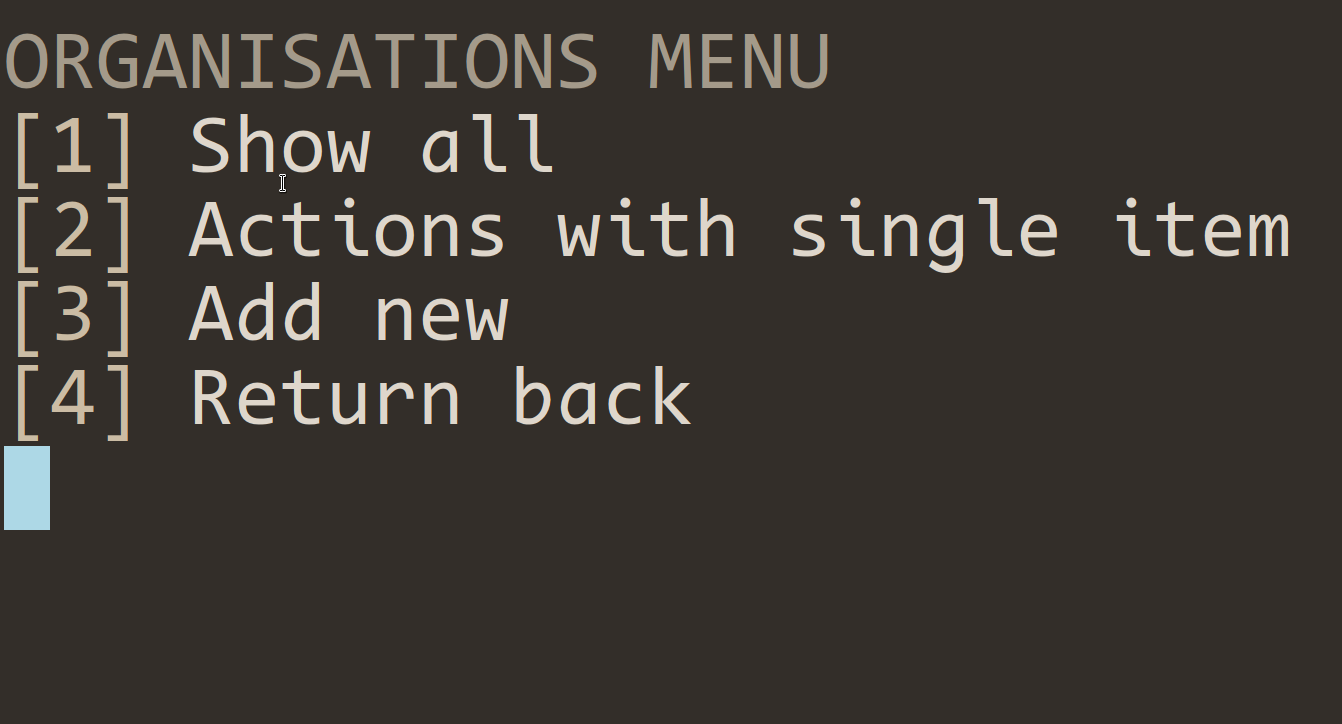
|  |
| --- |
| **main.cpp** |
| **#include "cui.hpp"**  #include <string>  #include <stdio.h> /\* defines FILENAME\_MAX \*/  #ifdef \_\_unix\_\_  # include <unistd.h>  # define GetCurrentDir getcwd  #elif defined(\_WIN32) || defined(WIN32)  # include <direct.h>  # define GetCurrentDir \_getcwd  #endif  #include<iostream>  using namespace std;  auto **main**() -> int {  char buff[FILENAME\_MAX];  getcwd( buff, FILENAME\_MAX );  string wdir(buff);  cout << "Current dir: " << endl;  cout << blu\_f + wdir + res " |" << endl;  for (int i = 0; i < (int)wdir.length() + 1; i++) cout << '\_';  cout << '|' << endl;  cout << "Enter location of files " yel\_f "(Press RETURN to use ../../data/ )" res << endl << blu\_f;  string str = "";  getline(*cin*, *str*);  cout << res;  if (str == "") str.append("../../data/");  cout << cont << endl;  getch();  system("clear");  FileStorage fs {str};  Cui cui{&fs};  cui.show();  } |

**Console**

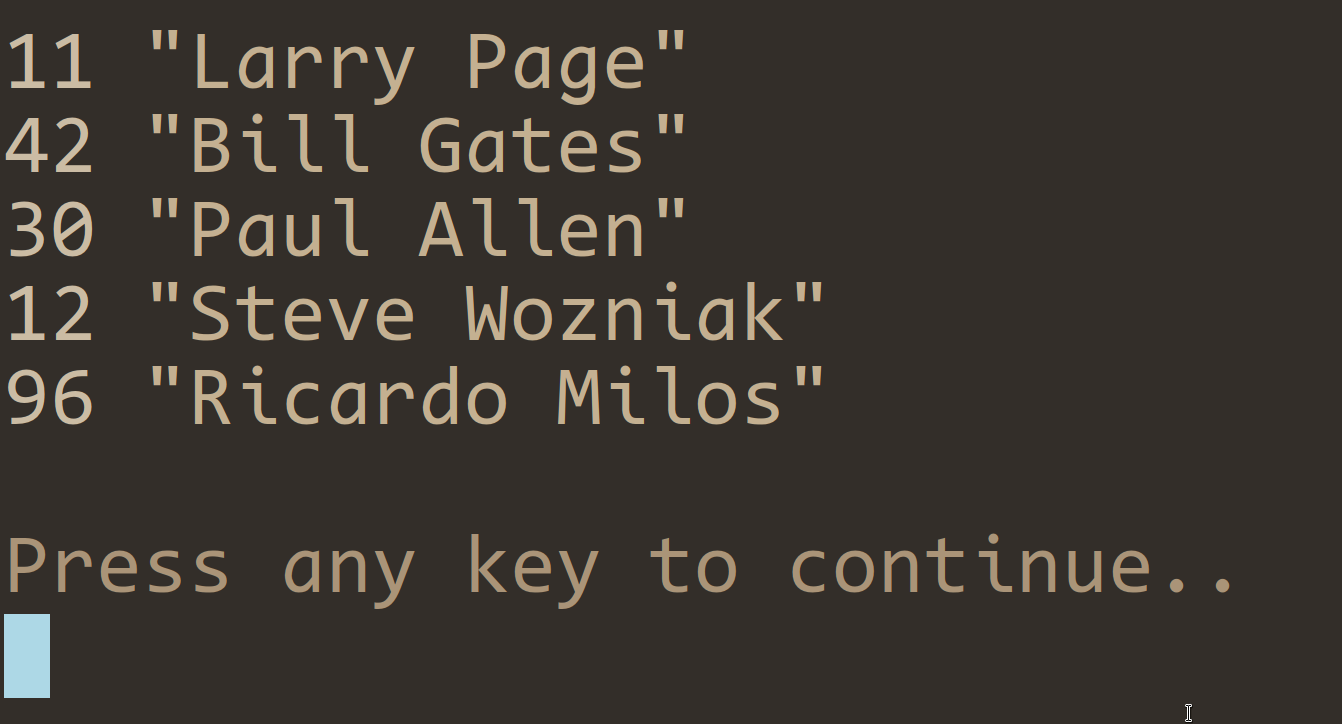
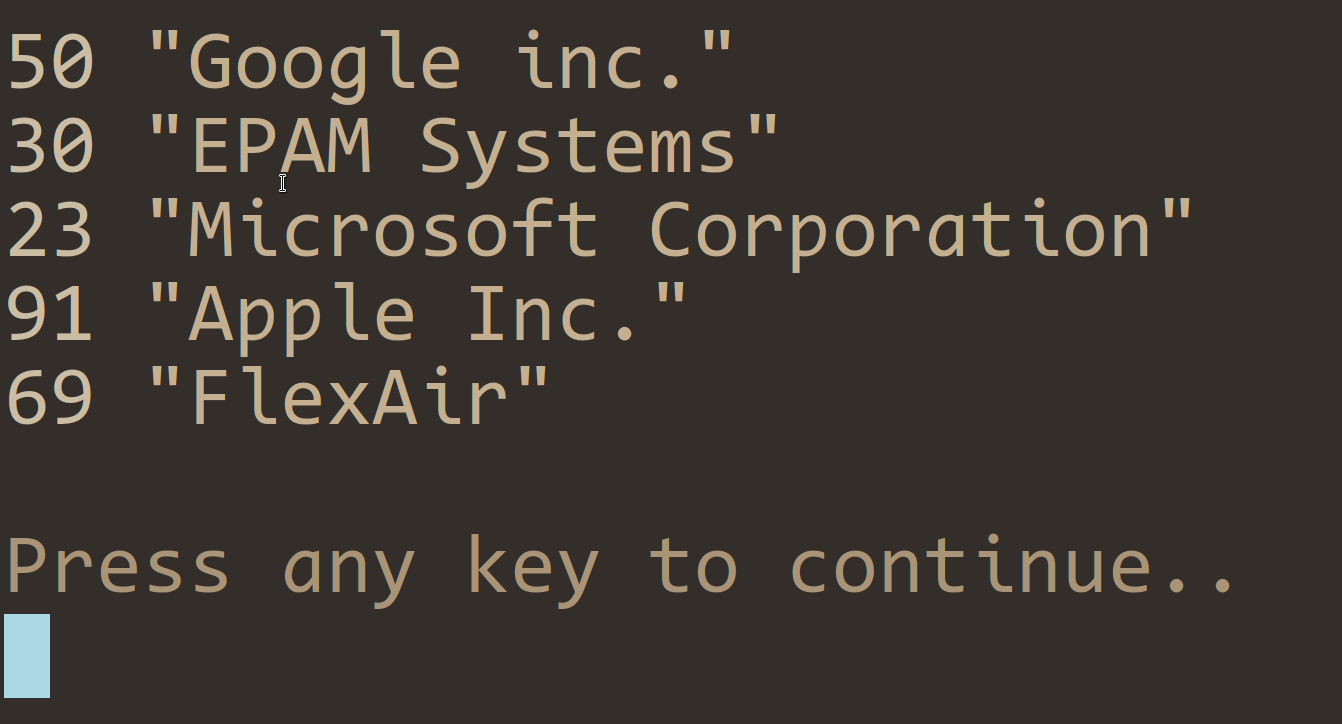
**Вибір директорії з даними Головне меню**



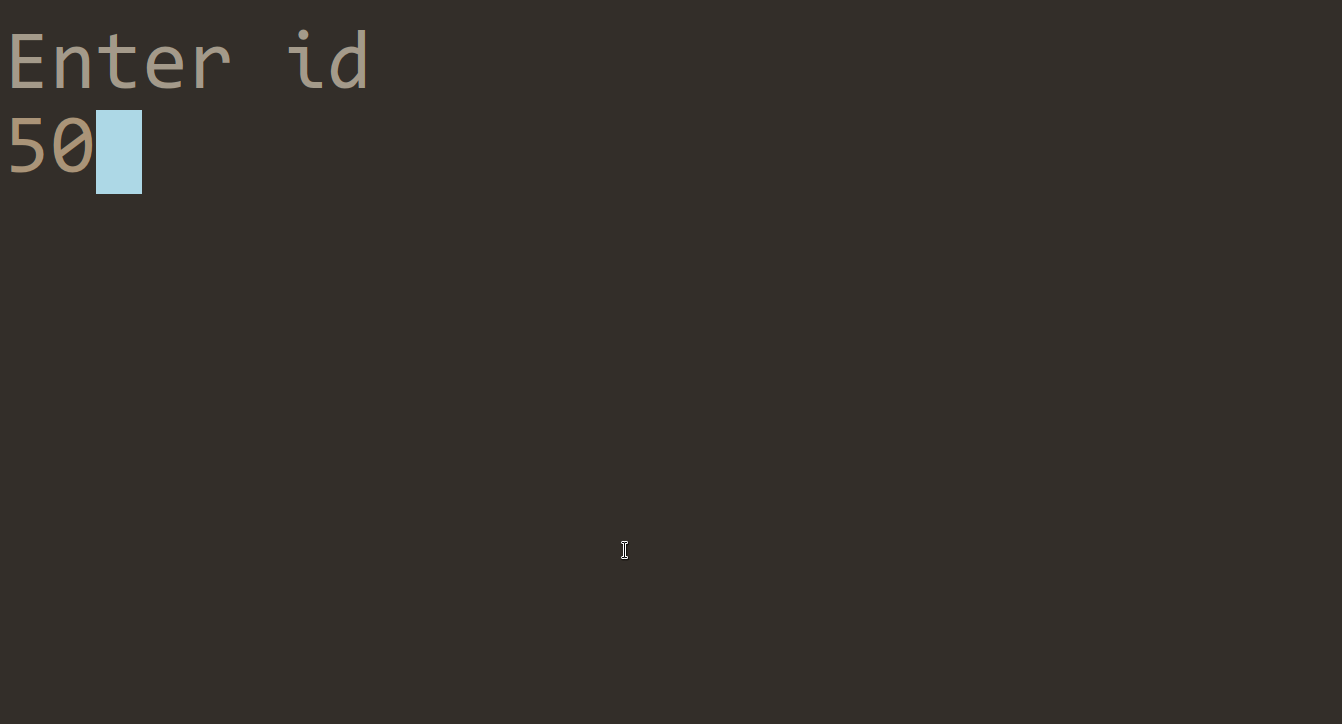
**Меню основної сутності Меню додаткової сутності**

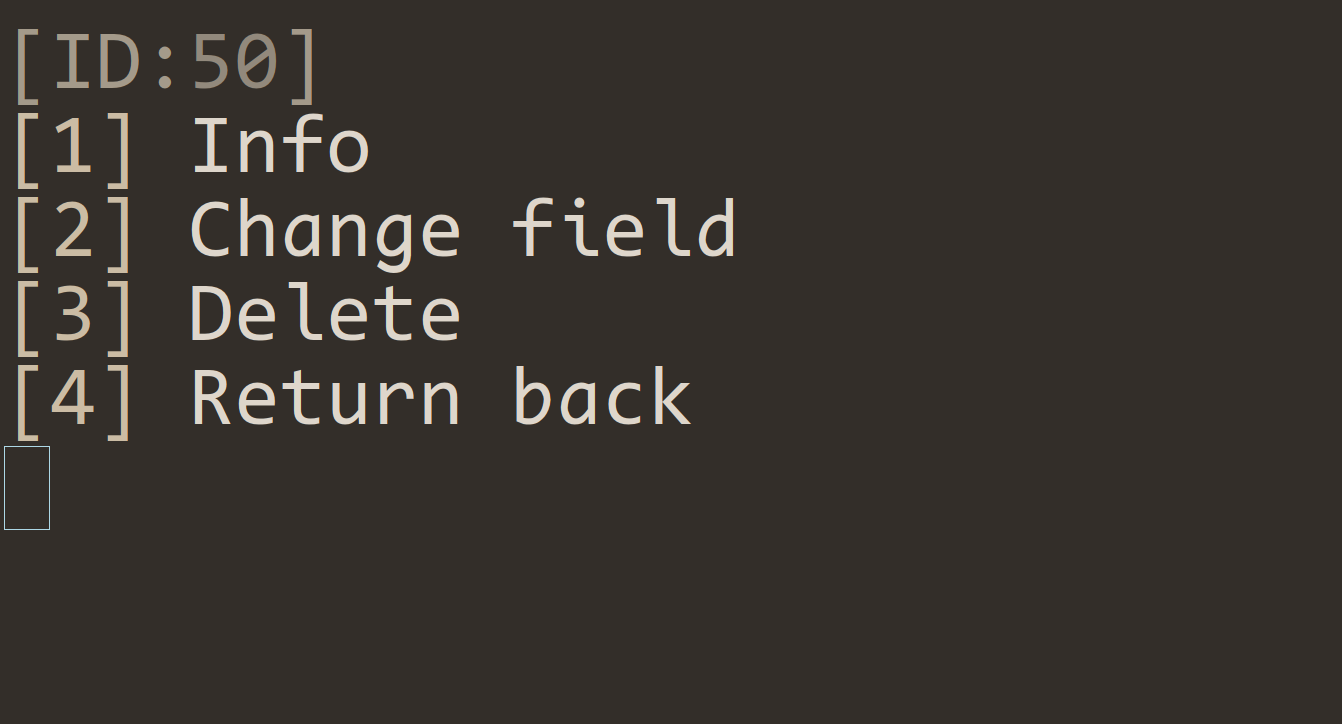
****

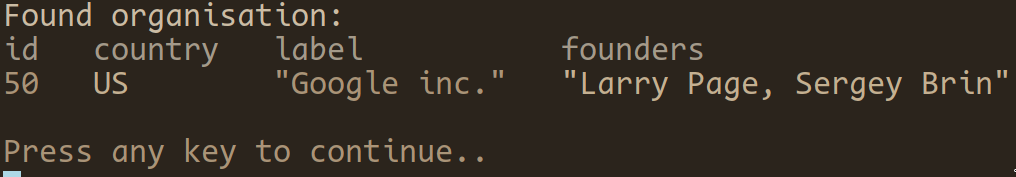
**Список основних сутностей Список додаткових сутностей**



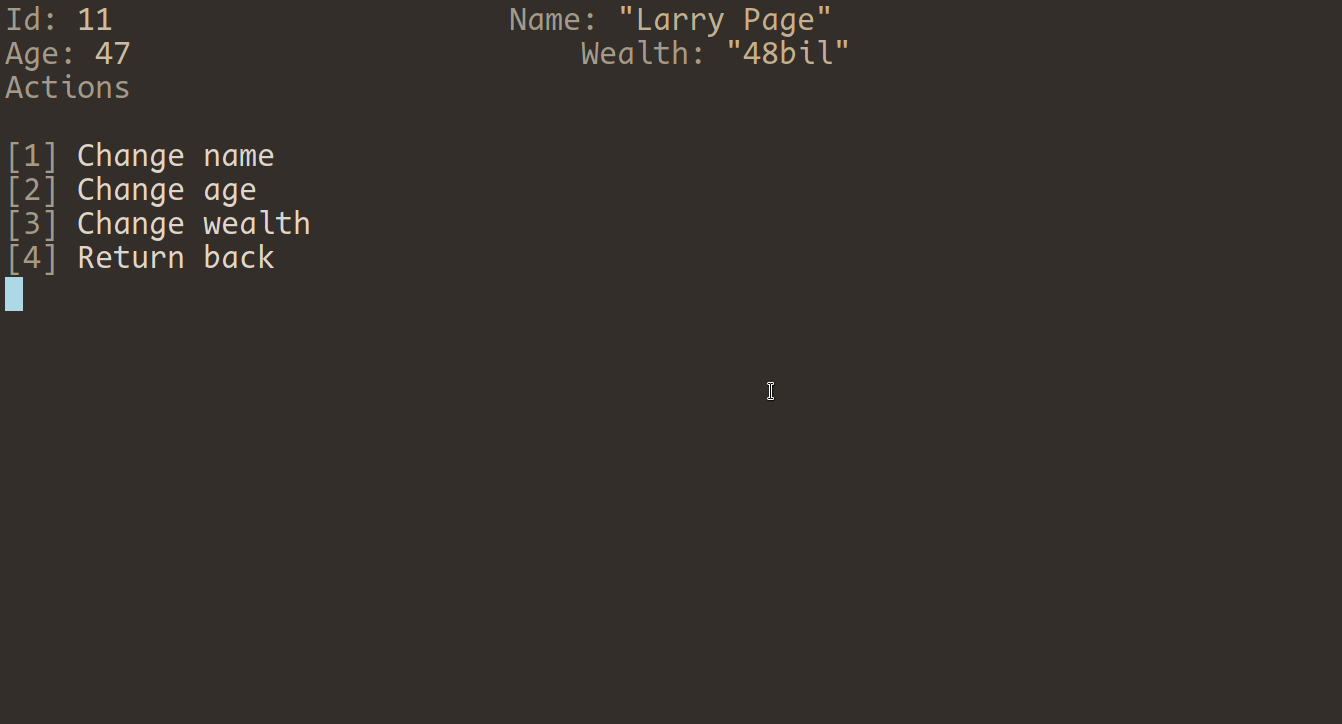
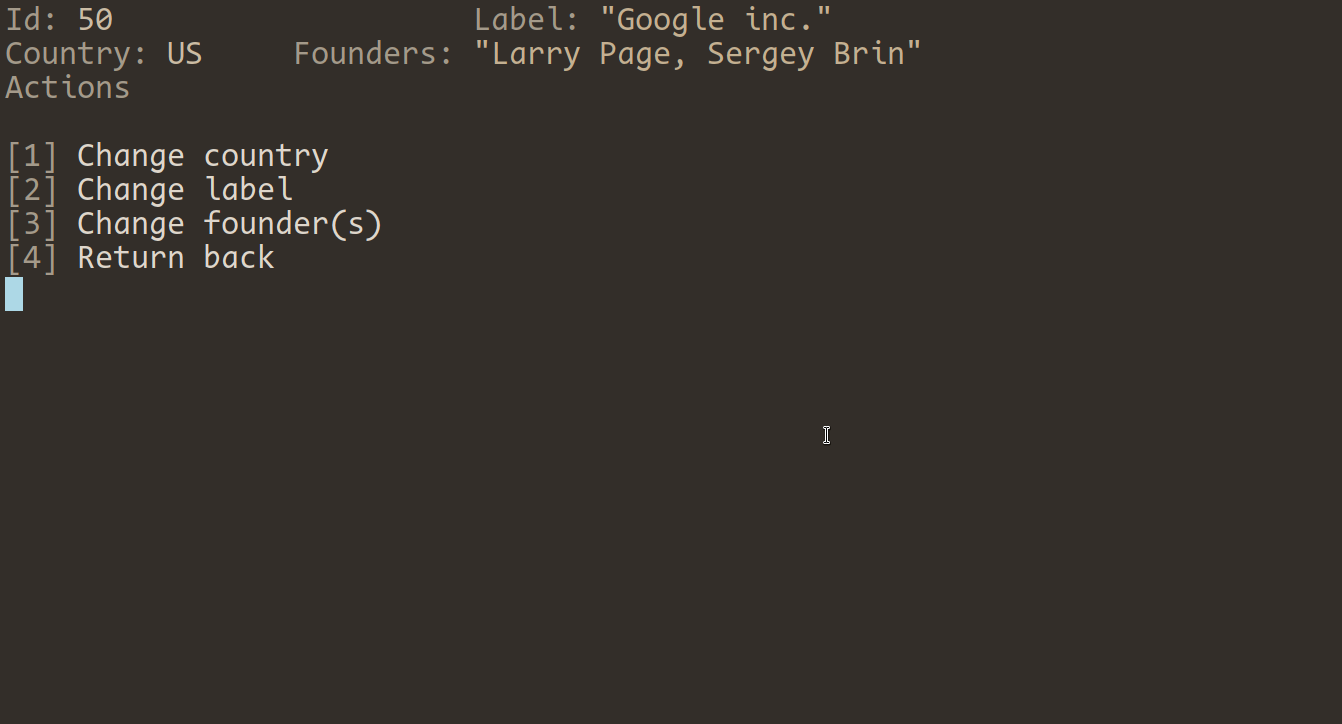
**Робота з окремою сутністю ( для дод. сутності так само )**



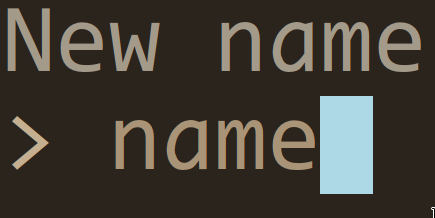




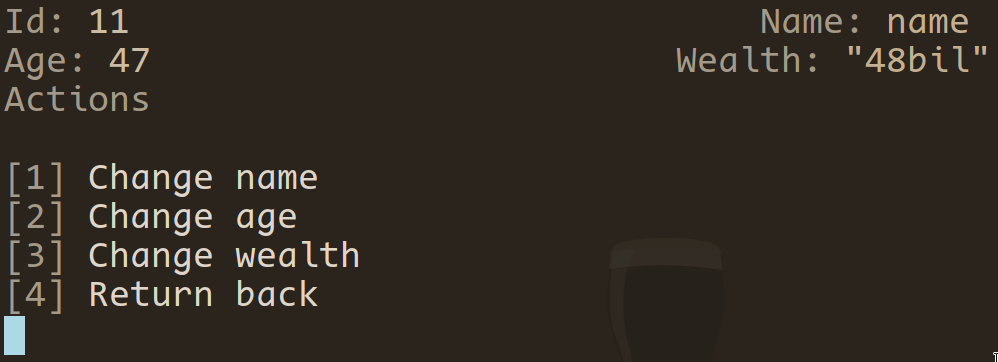
**Меню зміни поля**

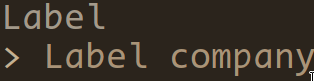
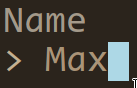
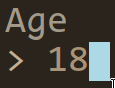
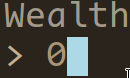
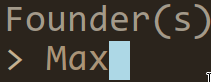
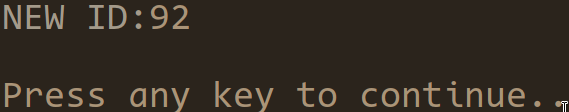
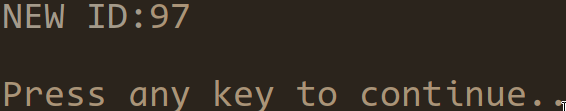


**Зміна поля сутності ( однаково для всіх )**

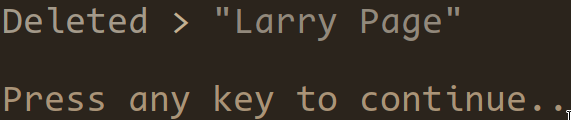


**Результат**

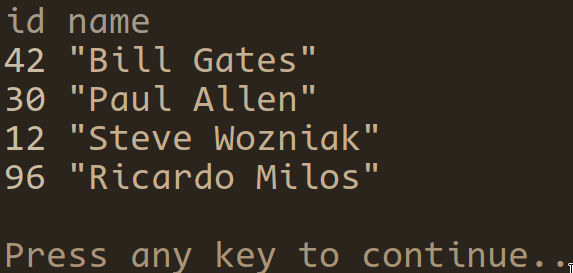




**Видалення сутності**

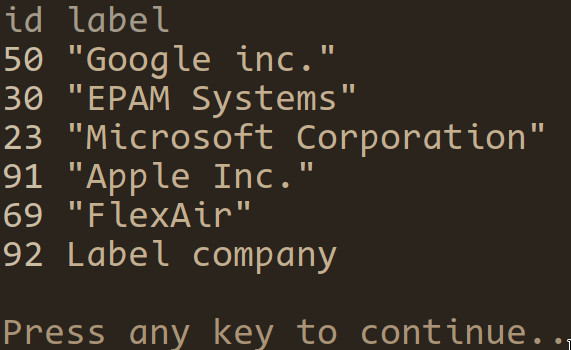
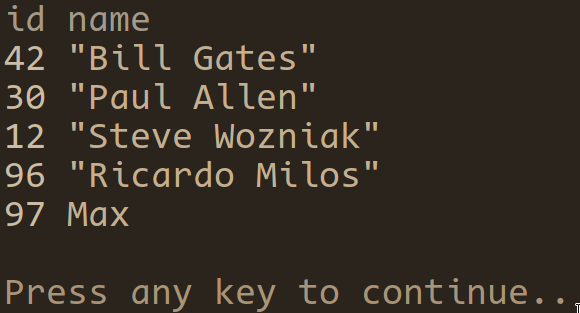


**Результат**



**Додавання нової сутності**

**Результат**



**Висновки**

Було винесено частину коду консольної програми у проект статичної бібліотеки libcsvlab. Ми створили консольний проект та проект статичної бібліотеки у Qt Creator. Ми додали додаткові методи у FileStorage для роботи з додатковою сутністю Founder. Ми використовували формат даних CSV , зчитували його і форматували. 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++.