**Лабораторна робота №3. Використання функцій та структур**

**Мета*:*** *навчитись складати й реалізовувати алгоритми та програми мовою С++ для обробки структур, використовуючи оператори введення та виведення до файлу.*

***Завдання:***

1. Ознайомитися з теоретичними відомостями та врахувати їх при виконанні завдання і в подальшій роботі.
2. Запустити середовище програмування С++ Builder (або інше доступне при виконанні вдома)
3. Створіть новий документ. Збережіть під іменем **<Номер групи><Номер лабораторної><Прізвищеанглійською>.срр**. Наприклад, 102buts.cpp. В подальших лабораторних роботах (ЛР) цей файл може використовуватися для нарощування функціональності та створення додатку.
4. У вікні редагування введіть текст програми, відповідно до наданої постановки задачі.
5. В першому рядку програми записати в коментарі номер групи та прізвище, а також номер ЛР
6. Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді cpp-файлу з іменем у форматі

**<Номер групи> <Номер лабораторної><Прізвище англійською>**

Наприклад, 31-01Ivanov.cpp.

Тему в заголовку листа записати

**ООП<Номер групи>-ЛР <Номер лабораторної>-<Прізвищеанглійською>**

**Строк відсилки ЛР ІПЗ-31 03.10.2020**

**ІПЗ-32 05.10.2020**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, Тему в заголовку листа записати

**ООП<Номер групи>-Запитання-<Прізвище англійською>**.

**Постановка задачі:**

1. Створюємо додаток, який буде оброблювати інформацію про викладачів та студентів.

Інформація про викладача (атрибути):

* Код (порядковий номер),
* прізвище та ініціали,
* стать (1 – жіноча, 2 чоловіча),
* телефон,
* кафедра,
* посада.

Інформація про студента (атрибути):

* Код (порядковий номер),
* прізвище та ініціали,
* стать (1 – жіноча, 2 чоловіча),
* телефон,
* група,
* поточна кількість балів по предметах фізика, математика, програмування.

Для роботи з цією інформацією створити масив структур, кількість екземплярів задати глобальною змінною.

1. На першому етапі створити меню, до якого в подальшому будуть додаватися нові функції. Для цієї ЛР позиції меню «***interactive data entry about the teacher***» (інтерактивне введення даних про викладача) (1), «***interactive data entry about the student***» (інтерактивне введення даних про студента) (2), «***calculation of the average score***» (розрахунок середнього балу) (3), "***Exit***" ("Вихід"). Повинна надаватися можливість багаторазового звернення до меню і виходу з програми за позицією "***Exit***".
2. Позиції меню (1) та (2) реалізуються аналогічно: На початку виводиться запит щодо режиму запису даних до файлу: формувати новий файл або дозаписати в існуючий. Потім вводиться інформація потоковим введенням і записується до файлу, де дані атрибутів відділяємо знаком ";", а кінець позначаємо знаком "#", після якого записуємо поточну дату у вигляді РРММДД (де РР – дві останні цифри року, ММ – номер місяця, ДД – номер дня). Відповідно, введені дані потім потрібно буде аналізувати та "розбирати" на складові, щоб присвоїти значення відповідним складовим програмного об’єкта. На цьому етапі ім’я файлу надаєте літералом у форматі FT\_**<Прізвищеанглійською>.txt** для викладачів, FS\_**<Прізвищеанглійською>.txt**. Наприклад, FR\_buts. **txt** для студентів.
3. В позиції меню (3) розраховується середній бал студента по всіх предметах та виводиться інформація про нього, у т.ч. розрахований бал. Розраховується середній бал по предметах по всій групі (сукупності) студентів.
4. Дії в позиціях меню оформлюються викликом відповідної функції, де вони реалізуються.

**Теоретичні відомості.**

Файловий ввід / вивід аналогічний стандартному вводу / виводу, єдина відмінність - це те, що введення / виведення виконаються не на екран, а в файл. Якщо введення / виведення на стандартні пристрої виконується за допомогою об'єктів cin і cout, то для організації файлового введення / виводу досить створити власні об'єкти, які можна використовувати аналогічно операторам cin іcout.

Для роботи з текстовими файлами потрібно

1. Підключити потрібну бібліотеку.

#include <fstream>

1. Описати файл для виводу даних

ofstream fout;

1. Відкрити файл (в прикладі Results.txt), який буде збережено в папці з проектом: Якщо такий уже існує, то він буде змінений, а попередня інформація буде втраченою.

fout.open(“results.txt”);

або можна зв‘язати об"єкт класу ofstream з вихідним файлом при при описі

ofstream fout("results.txt", ios::out);

1. Вивести дані:

fout<< “Output Data.”;

1. Закрити файл:

fout.close();

Для введення даних використовується об‘єкт ifstream. Дії по використанню аналогічні, але потрібно перевіряти чи відкрився файл.

Розглянемо приклад.

#include <fstream>

#include <string>

#include <iostream>

using namespace std;

int main()

{ ofstream fout; // створюємо об"єкт класу ofstream для запису

string fam="Fam6"; // Прізвище³

int codos=0; // код

int rikn=2005; // рік народження

string telef="+380442233445"; //телефон

int gender=1; // стать

string inptLine; // для зчитування даних

ifstream finpt; // створюємо об"єкт класу ifstream для читання

fout.open("MyFile.txt",ios:: app); // **Для дозапису**

fout<< fam<< ";"<<codos<<";"<<rikn<<";"<<telef<<";"<<gender<<"#"<<endl;

fout<< "New"<< ";"<<codos<<";"<<rikn<<";"<<telef<<";"<<gender<<"#"<<endl;

fout.close(); // закриваемо файл

cout << "Reading in C++"<<endl;

finpt.open("MyFile.txt",ios::in);

if (! finpt.is\_open()) // Якщо файл не відкрився

cout << "File not be opened!\n"; // Повідомлення про помилку

else

{ while (!finpt.eof())

{finpt >> inptLine; // читаємо дані

cout << inptLine << endl; //виводимо прочитане на консоль

}

finpt.close(); // закриваємо файл

}

system("pause");

return 0;

}

Режими відкриття файлів встановлюють характер використання файлів. Для встановлення режиму в класі **ios** передбачені константи, які визначають режим відкриття файлів.

| Константа | Опис |
| --- | --- |
| ios::in | Відкрити файл для читання |
| ios::out | Відкрити файл для запису |
| ios::ate | При відкритті перемістити вказівник у кінець файлу |
| ios::app | Відкрити файл для запису в кінець файлу |
| ios::trunc | Видалити вміст файлу, якщо він існує |
| ios::binary | Відкриття файлу в двійковому режимі |