**Лабораторна робота №7. Поліморфізм**

*Тема.* Класи. Поліморфізм. Абстрактні класи.

*Мета.* Отримати знання про парадигму ООП – поліморфізм.

Навчитися застосовувати отримані знання на практиці.

**1.Вимоги**

*Розробник :* Кабак О.Р. ,НТУ “ХПІ” ,КІТ102.8а

**1.1 Основне завдання**

Модернізувати попередню лабораторну роботу шляхом:

-додавання ще одного класу-спадкоємця до базового класу. Поля обрати самостійно;

-базовий клас зробити абстрактним. Додати абстрактні поля;

-розроблені класи-списки поєднуються до одного таким чином, щоб він міг працювати як з базовим класом, так і з його спадкоємцями. При цьому, серед полів класу-списку повен бути лише один масив, що містить усі типи класів ієрархії. Оновити методи, що працюють з цим масивом.

**1.2 Додаткові умови виконання завдання.**

-продемонструвати відсутність витоків пам’яті;

-продемонструвати роботу розроблених методів за допомогою

модульних тестів;

-не використовувати конструкцію «using namespace std;», замість

цього слід роботи «using» кожного необхідного класу:using std::string,

using std::cout;

-в проекті не повинні використовуватися бібліотеки введення /

виведення мови С, а також не повинні використовуватися рядки типу

char\*.

**2.Опис програми**

**2.1. Функціональне призначення**

Програма створена для генерування динамічного масиву самостійних робіт студента з сутністю спадкоємців “базового класу”.

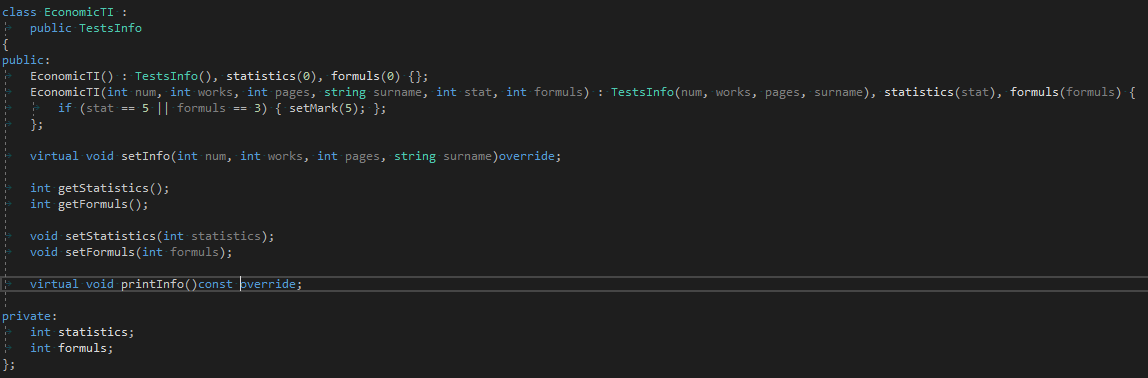


Рис3.1 Новий спадкоємець EconomicTI--(public)-->TestsInfo

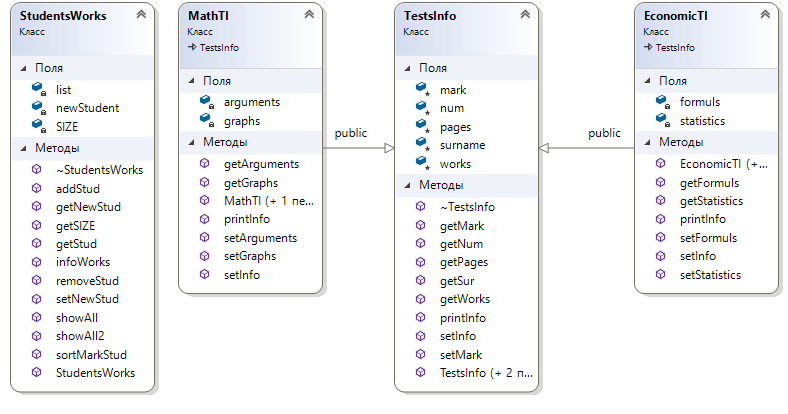


Рис 3.2. Діаграми класів, новий класс-спадкоємець, перероблений контейнер.

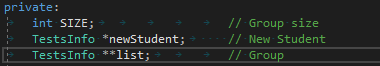


Рис 3.3 Тепер поля класу-контейнера змінені на вказівник на елемент і масив вказівників на елемент

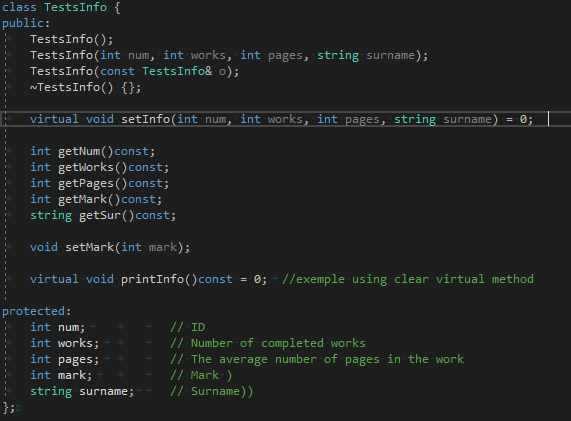


Рис 3.4 Базовий клас \*батько\* перетворений на абстрактний з чистими віртуальними методами

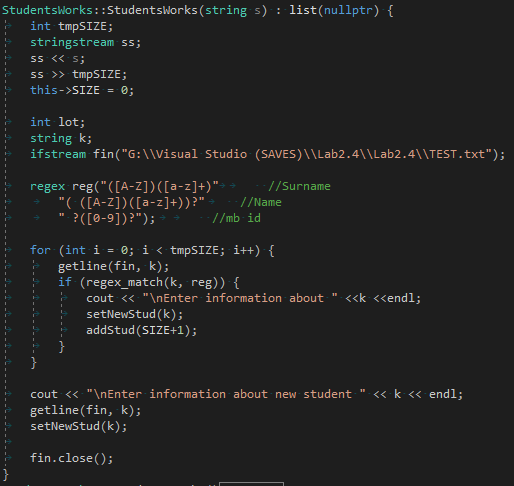


Рис 3.5 Змінено підхід створення масиву студентів(тепер ця група складатиметься з математиків та економістів)

Код для створення Економіста\Математика:

void StudentsWorks::setNewStud(string s) {

regex reg("([A-Z])([a-z]+)" //Surname

"( ([A-Z])([a-z]+))?" //Name

" ?([0-9])?"); //mb id

if (regex\_match(s, reg)) {

int works, pages, a, b;

int i = 2;

while (i > 1) {

cout << "Enter the number of student works(0-10):";

cin >> works;

if (works < 0 || works > 10) {

cout << "You entered an invalid value, try again\n";

}

else { i = 1; }

}

while (i > 0) {

cout << "Enter the number of pages in works(1-5):";

cin >> pages;

if (pages < 1 || pages >5) {

cout << "You entered an invalid value, try again\n";

}

else { i = 0; }

}

cout << "From which group is the student?(Math-0,Economic-1)\n";

bool z;

cin >> z;

if (z) {

i = 2;

while (i > 1) {

cout << "Enter the number of statistics(1-5):";

cin >> a;

if (a < 0 || a > 5) {

cout << "You entered an invalid value, try again\n";

}

else { i = 1; }

}

while (i > 0) {

cout << "Enter the number of formuls(1-3):";

cin >> b;

if (b < 1 || b >3) {

cout << "You entered an invalid value, try again\n";

}

else { i = 0; }

}

newStudent = new EconomicTI(SIZE + 1, works, pages, s, a, b);

}

else {

i = 2;

while (i > 1) {

cout << "Enter the number of graphs (1-5):";

cin >> a;

if (a < 0 || a > 5) {

cout << "You entered an invalid value, try again\n";

}

else { i = 1; }

}

while (i > 0) {

cout << "Enter the number of arguments(1-3):";

cin >> b;

if (b < 1 || b >3) {

cout << "You entered an invalid value, try again\n";

}

else { i = 0; }

}

newStudent = new MathTI(SIZE + 1, works, pages, s, a, b);

}

}

else {

cout << "Surname entered incorrectly\n";

}

}

Висновки

В даній лабораторній роботі отримані знання про парадигму ООП - Поліморфізм.Отримані навички застосування отриманих знань на практиці шляхом написання класу-спадкоємця та класу контролера.