

# Лабораторна робота №6. Спадкування

Тема. Класи. Спадкування.

Мета. Отримати знання про парадигму ООП – спадкування.

Навчитися застосовувати отримані знання на практиці.

## 1.Вимоги

Розробник : Кабак О.Р. ,НТУ “ХП” ,KIT102.8a

### 1.1 Основне завдання

Модернізувати попередню лабораторну роботу шляхом:

- додавання класу-спадкоємця, котрий буде поширювати функціонал «базового класу» у відповідності до індивідуального завдання;
- додавання ще одного класу-списку, що буде керувати лише елементами класу-спадкоємця;
- в функціях базового класу та класу-спадкоємця обов’язкове використання ключових слів final та override.

### 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 проєкті не повинні використовуватися бібліотеки введення / виведення мови C, 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 спадкоємця “базового класу”.

### 2.2.Опис логічної структури:

```
virtual void setInfo(int num, int works, int pages, string surname);

virtual int getNum()const final;
virtual int getWorks()const final;
virtual int getPages()const final; + ...//exemple using final
virtual int getMark()const final;
virtual string getSur()const final;

virtual void setMark(int mark) final;

virtual void printInfo()const; +
```

Рис.3.1 Приклад методів \*батька\* з віртуальними та деякими фінальними методами

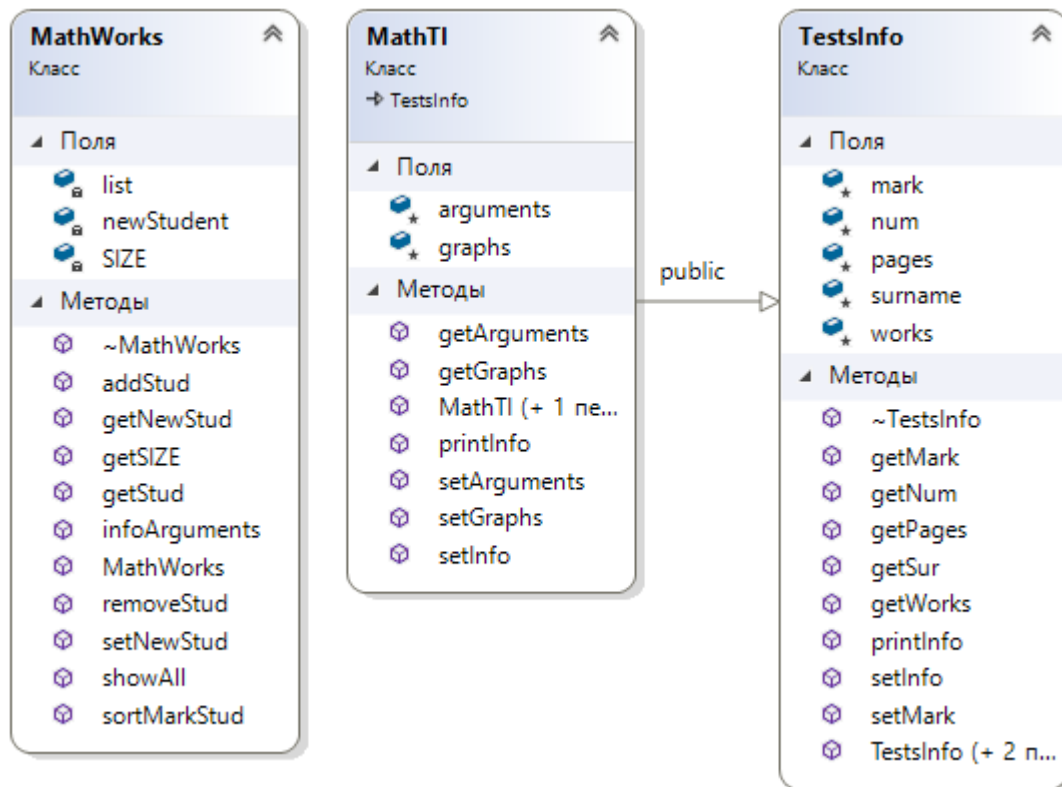


Рис.3.2 Діаграми класу-масиву спадкоємців(MathWorks), клас-спадкоємець(MathTI)--(public)-->TestsInfo

```

class MathTI :
    public TestsInfo
{
public:
    MathTI() : TestsInfo(), graphs(0), arguments(0) {};
    MathTI(int num, int works, int pages, string surname, int graphs, int arguments) :
        TestsInfo(num, works, pages, surname), graphs(graphs), arguments(arguments) {
        if (graphs == 5 || arguments == 3) { setMark(5); };
    };

    virtual void setInfo(int num, int works, int pages, string surname)override;

    int getGraphs();
    int getArguments();

    void setGraphs(int graphs);
    void setArguments(int arguments);

    virtual void printInfo()const override; //exemple using override
protected:
    int graphs;
    int arguments;
};
  
```

Рис 3.3 Приклад коду класу спадкоємця MathTI ->TestsInfo з public доступом

```
Your list for Math froup
Current student journal:

Student number:1
Number of completed works:8
Average volume of work:3
Number of graphs in work:3
Number of strong arguments:3
GPA:5
Surname :Surname1

Student number:2
Number of completed works:3
Average volume of work:3
Number of graphs in work:5
Number of strong arguments:3
GPA:5
Surname :Surname2

Student number:3
Number of completed works:8
Average volume of work:2
Number of graphs in work:5
Number of strong arguments:2
GPA:5
Surname :Surname3

Student number:4
Number of completed works:4
Average volume of work:2
Number of graphs in work:3
Number of strong arguments:2
GPA:3
Surname :Surname4

Student who wants to join the group:

Student number:5
Number of completed works:7
Average volume of work:4
Number of graphs in work:3
Number of strong arguments:2
GPA:5
Surname :Surname5
```

Рис.3.4 Результат виводу списку математиків на екран

#### Висновки

На лабораторній роботі отримано знання про парадигму ООП - Спадкування.

Отримані навички застосування отриманих знань на практиці шляхом написання класу-спадкоємця та класу контролера.