

Лабораторна робота №5. Агрегація та композиція

Тема. Класи. Агрегація. Композиція. Ключові слова typedef та auto.

Мета. Порівняти поняття агрегація та композиція. Отримати знання про призначення ключових слів typedef та auto.

1.Вимоги

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

1.1 Основне завдання

Дослідити заздалегідь визначені типи даних з бібліотеки `<cstdlib>` / `<stdlib.h>`. Модернізувати розроблені у попередній роботі класи наступним чином:

- замінити типи даних, що використовуються при індексуванні на типи з вказаної бібліотеки;
 - створити власний синонім типу, визначивши його необхідність;
 - створити/оновити функцію сортування масиву, де крім поля, по якому виконується сортування, передається і вказівник на функцію, яка визначає напрям сортування;
 - в базовий клас додати два поля, що мають кастомний тип даних (тип даних користувача) та які будуть відображати відношення «агрегація» та «композиція», при цьому оновити методи читання та запису об'єкта;
 - ввести використання ключового слова auto як специфікатор зберігання типу змінної.
- Визначити плюси та мінуси цього використання.

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 “базового класу”.

2.2.Опис логічної структури:

TestsInfo отримав поля з прикладом агрегації(graph) та композиції(plan)

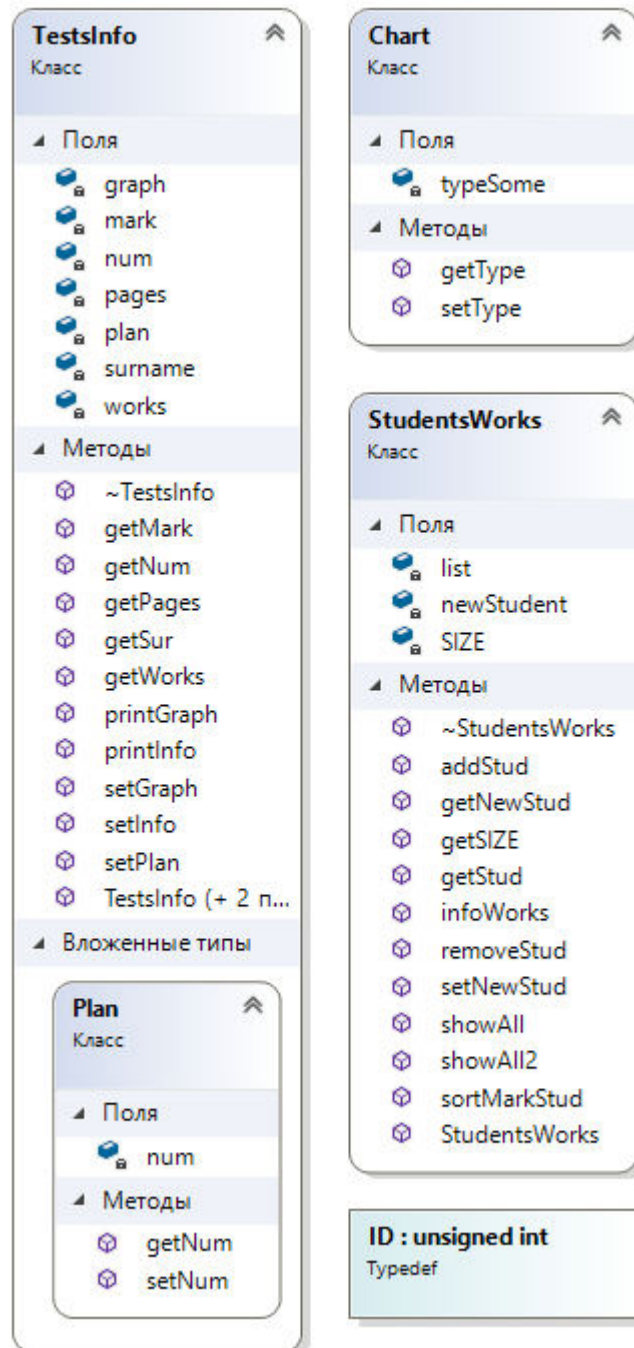


Рис.3.1 Діаграми класів

```

Student number:1
Number of completed works:6
Average volume of work:2
GPA:3
Surname :Zaycev!NONE!

Student number:2
Number of completed works:7
Average volume of work:5
GPA:5
Surname :Volkov Ivan!NONE!

Student number:3
Number of completed works:7
Average volume of work:5
GPA:5
Surname :Volkov Ivan 2!NONE!
Student who wants to join the group:
The copy constructor worked here.

```

Рис.3.2 Приклад використання плану котрого у студентів немає _(ツ)_/

```

> /*-----Composition example-----*/
> class Plan {
> private:
>     ID num = 0; → // Number of points plan → Using ID type
> public:
>     void setNum(int number) { ... }
>     ID getNum() { return num; }
> };
>
> void setPlan() { ... }
>
> /*-----Aggregation example-----*/
> void setGraph();
> void printGraph();
>
private:
>     int num; → // ID
>     int works; → // Number of completed works
>     int pages; → // The average number of pages in the work
>     int mark; → // Mark )
>     string surname; → // Surname))
>     Chart graph; → // Graphics
>     Plan plan; → // Plan
> };

```

Рис.3.3 Фрагменти коду, які містять агрегацію і композицію
+ використання синоніму до *unsigned int* = ID

```

class Chart{
private:
    string typeSome = "!NONE!"; // Type of graphics
public:
    void setType(string chart){
        typeSome = chart;
    }

    string getType(){return typeSome;}
};

```

Рис.3.4 Клас-приклад зберігаючий у собі тип діаграми(у випадку студента - графік)

```

string TestsInfo::printInfo()const {
    stringstream ss;
    ss << "\n\n Student number:" << this->num << endl << "Number of completed works:" << this->works << endl
    << "Average volume of work:" << this->pages << endl << "GPA:" << this->mark << endl << "Surname :"
    << this->surname;
    auto s = ss.str(); //An example of using a variable of type auto
    return s;
}

```

Рис.3.5 Приклад використання авто-змінної ,котра стала типом string .

Висновки

На лабораторній роботі отримано знання про базові вирази та добутий досвід роботи по застосуванню їх на практиці. У рамках лабораторної роботи розроблені методи, які перевіряють рядок на наявність цифр, латинських символів, пропусків, розділових знаків, відсутність маленького символу на початку та наявність з щонайменше з 2ма словами.