ЛАБОРАТОРНА РОБОТА №16. РОБОТА З ДИНАМІЧНОЮ ПАМ’ЯТТЮ

Тема. Системна робота з динамічною пам’яттю.

Мета. Дослідити особливості мови С++ при роботі з динамічною пам’яттю.

1 ВИМОГИ

**1.1 Розробник**

- Котенко Сергій Миколайович;

- Студент групи КІТ 102.8(а);

- 12-06-2019р..

**1.2 Загальне завдання**

Загальне завдання.

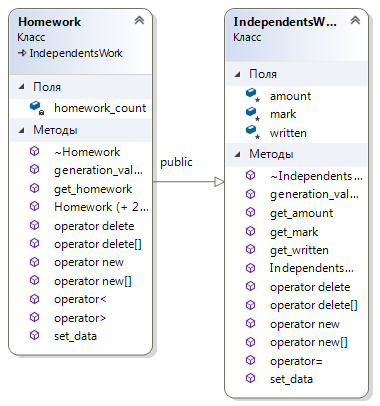
Маючи класи з прикладної області РЗ (тільки базовий клас та клас/класи спадкоємці), перевантажити оператори new / new [] та delete / delete []. Продемонструвати їх роботу і роботу операторів розміщення new / delete при розробці власного менеджера пам’яті (сховища). Детальна інформація про власне сховище: є статично виділений масив заданого обсягу. Організувати виділення і звільнення пам’яті елементів ієрархії класів тільки в рамках даного сховища.

Додаткові умови виконання завдання:

- продемонструвати відсутність витоків пам’яті

2 ОПИС ПРОГРАМИ

**2.1 Опис логічної структури**



*Діаграма класу IndependentsWork*:

* ~IndependentsWork - Деструктор класу;
* generation\_values – Генерація випадкових значень;
* get\_amount, get\_mark, get\_written - Отримання даних;
* IndependentsWork - Конструктор класу;
* operator delete, operator delete[] , operator new, operator new[] , operator= - Перевантаження операторів;
* set\_data - Встановлення значень .

*Діаграма класу Homework*:

* ~Homework - Деструктор класу;
* generation\_values – Генерація випадкових значень;
* get\_homework - Отримання даних;
* Homework - Конструктор класу;
* operator delete, operator delete[] , operator new, operator new[] , operator= , operator<, operator> - Перевантаження операторів;
* set\_data - Встановлення значень .

**2.2 Фрагменти коду**

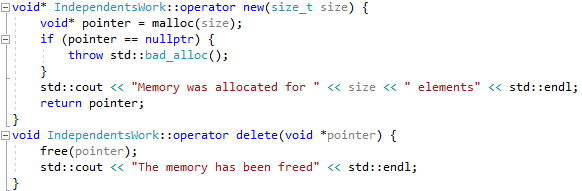


Рисунок 2.1 – Перевантаження операторів new та delete

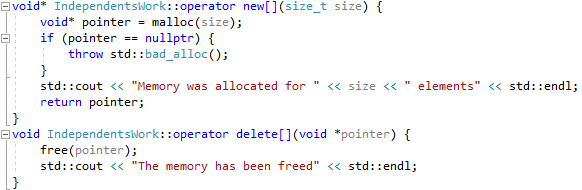


Рисунок 2.2 – Перевантаження операторів new[] та delete[]



Рисунок 2.3 – Перевірка працездатності

**3 ВАРІАНТИ ВИКОРИСТАННЯ**

**3.1 Ілюстрація роботи програми**



Рисунок 3.1 – Результат виділення та очистки пам’яті

**ВИСНОВОК**

В інтегрованому середовищі *Visual Studio* розроблена програма мовою С++. Виконання програми дозволяє продемонструвати коректність роботи перевантаження операторів new / new [] та delete / delete [].