Лабораторна робота№4-5

ДОСЛІДЖЕННЯ РОБОТИ КОНСТРУКТОРІВ ТА ДЕСТРУКТОРІВ

**Мета:** продемонструвати роботу конструкторів та деструкторів

**Індивідуальне завдання:** написати програму, що працює з конструкторами та деструкторами. Реалізувати поведінку в конструторах та деструкторах, що викликає виняткову ситуацію. Описати поведінку, зокрема створення та видалення об’єктів з участю поліморфізму; можливість перехоплення виключної ситуації з метою виправлення помилок. В розробленої програми дослідити та навести результати досліджень (алгоритм), як саме виконується процес неявного виклику конструкторів на деструкторів (створення та руйнування об’єктів на рівні стеку; виклик методу delete)

ХІД РОБОТИ

Дана лабораторна робота проводилася у сфері розробки Eclipse на основі старої лабораторної роботи. На рис.1 наведено зміст main.cpp, де проводиться робота з двома сутностями (рис.2-3).

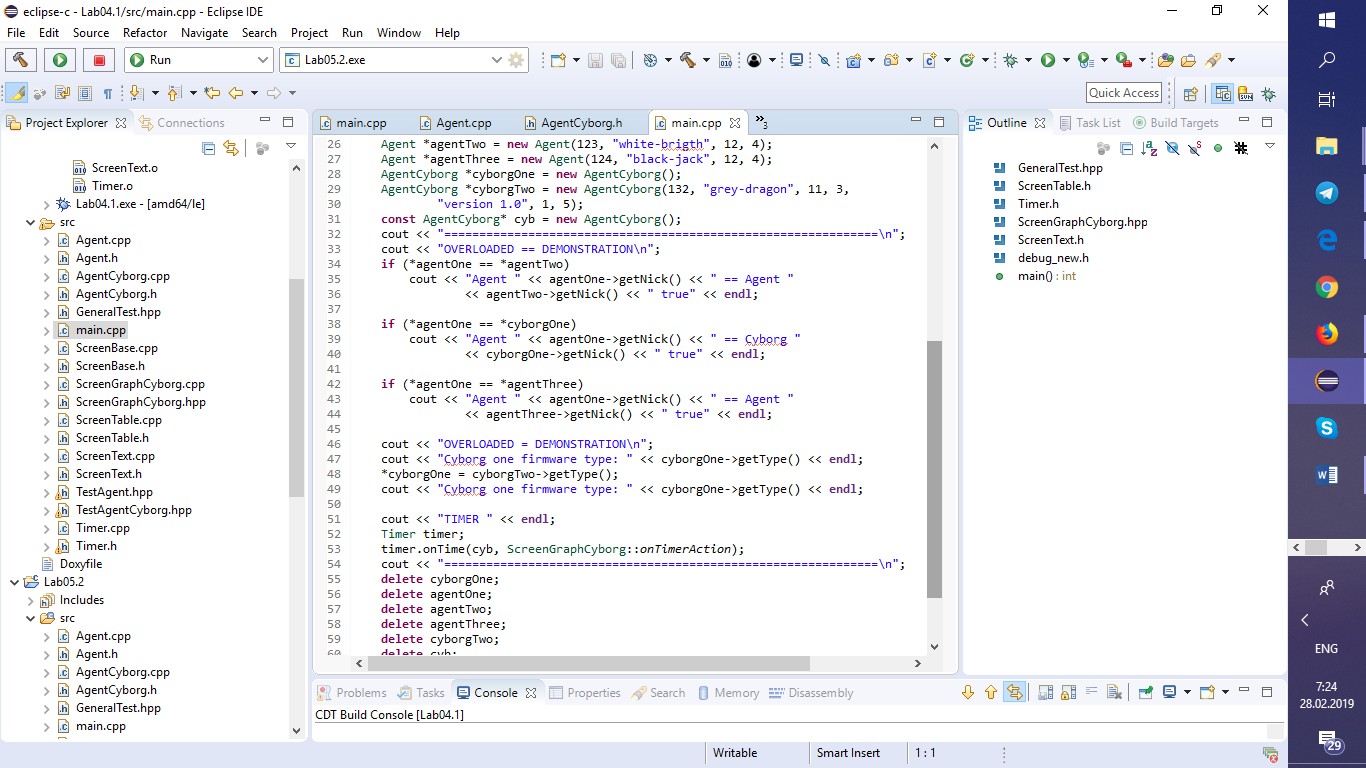


Рисунок 1 — Робота з сутностями

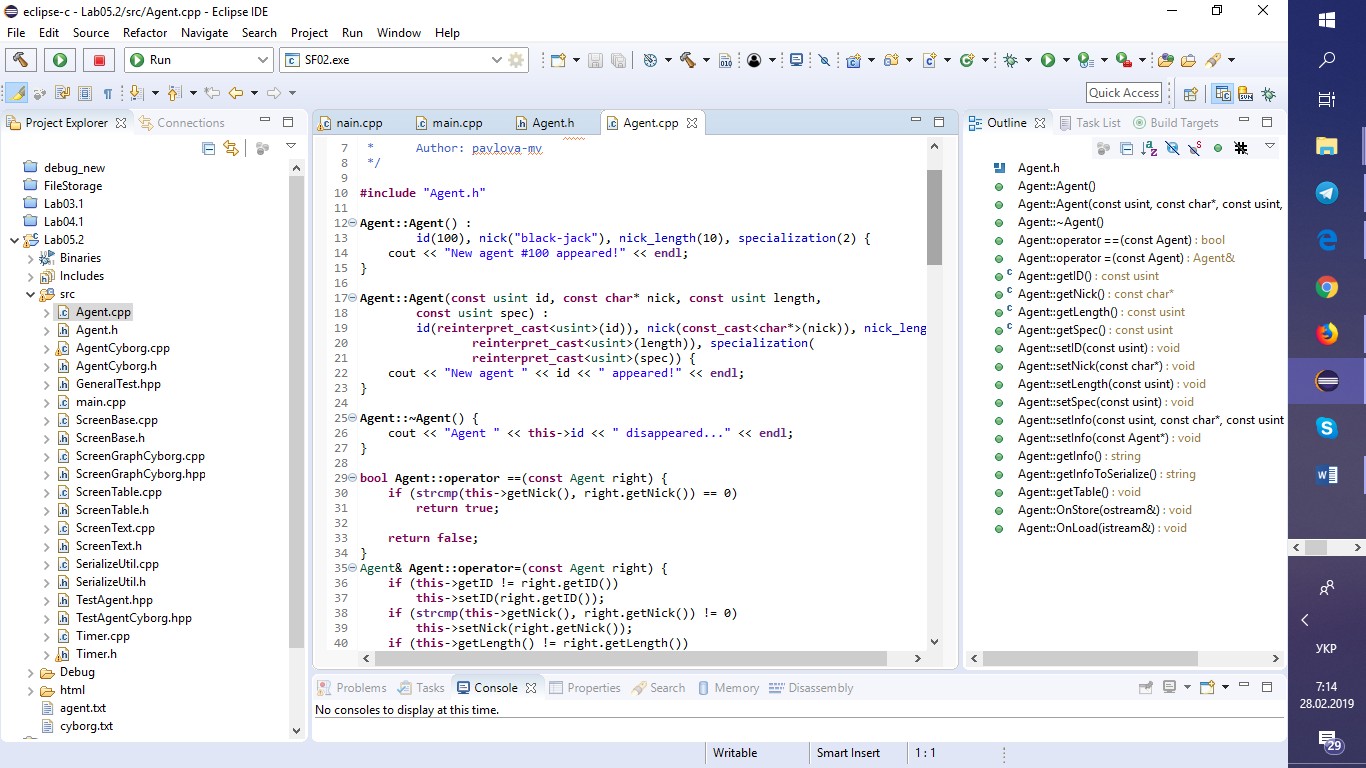


Рисунок 2 — Конструктори та деструктори першої сутності

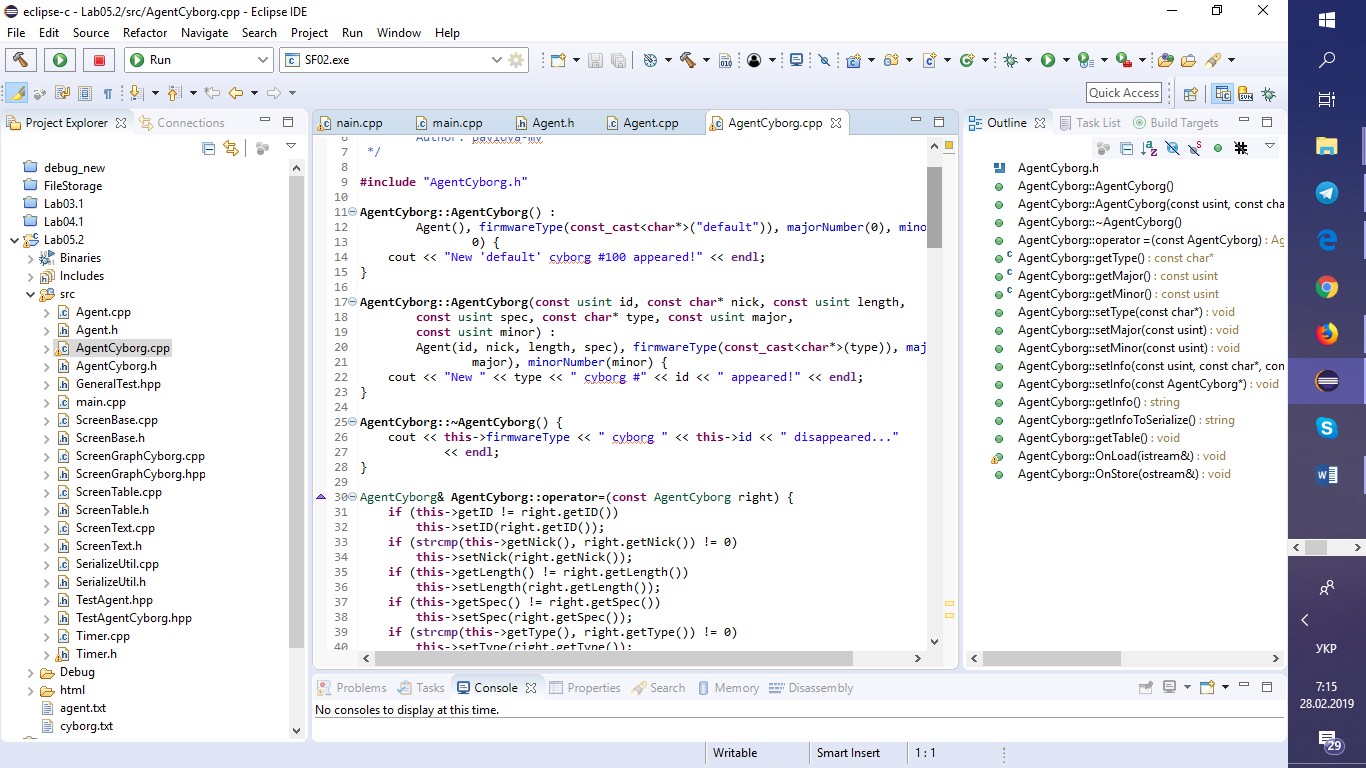


Рисунок 3 — Конструктори та деструктори другої сутності

В результаті виконання коду програми було отримано наступний результат (рис.4-5).

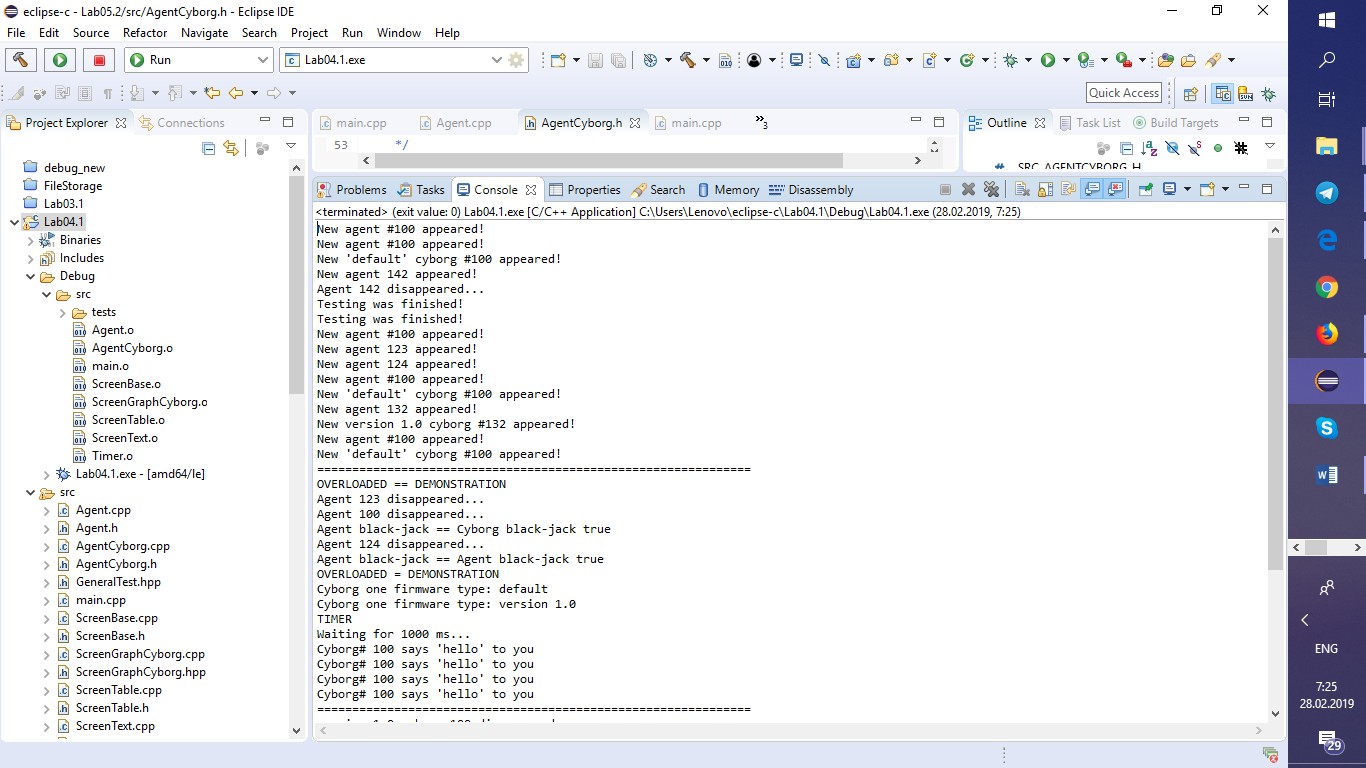


Рисунок 4 — Кількість конструкторів для 6 об’єктів

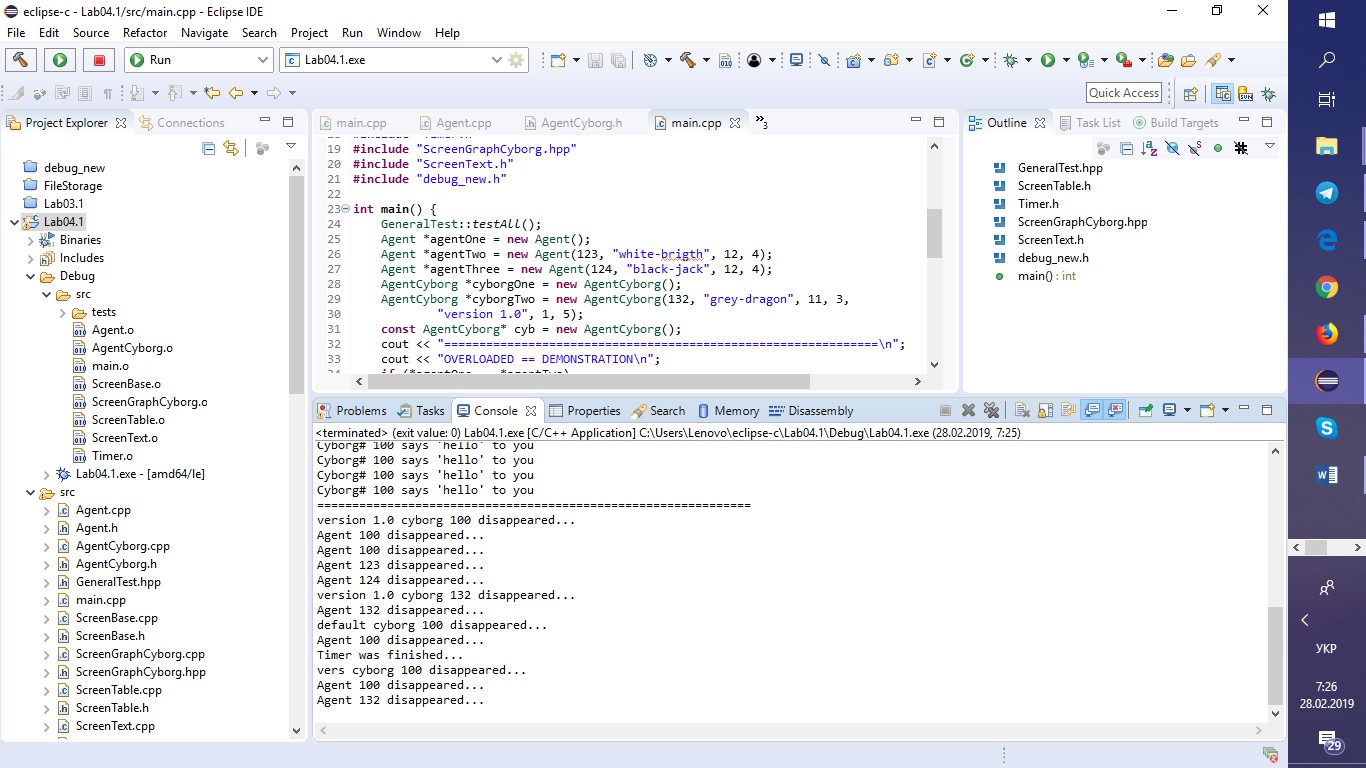
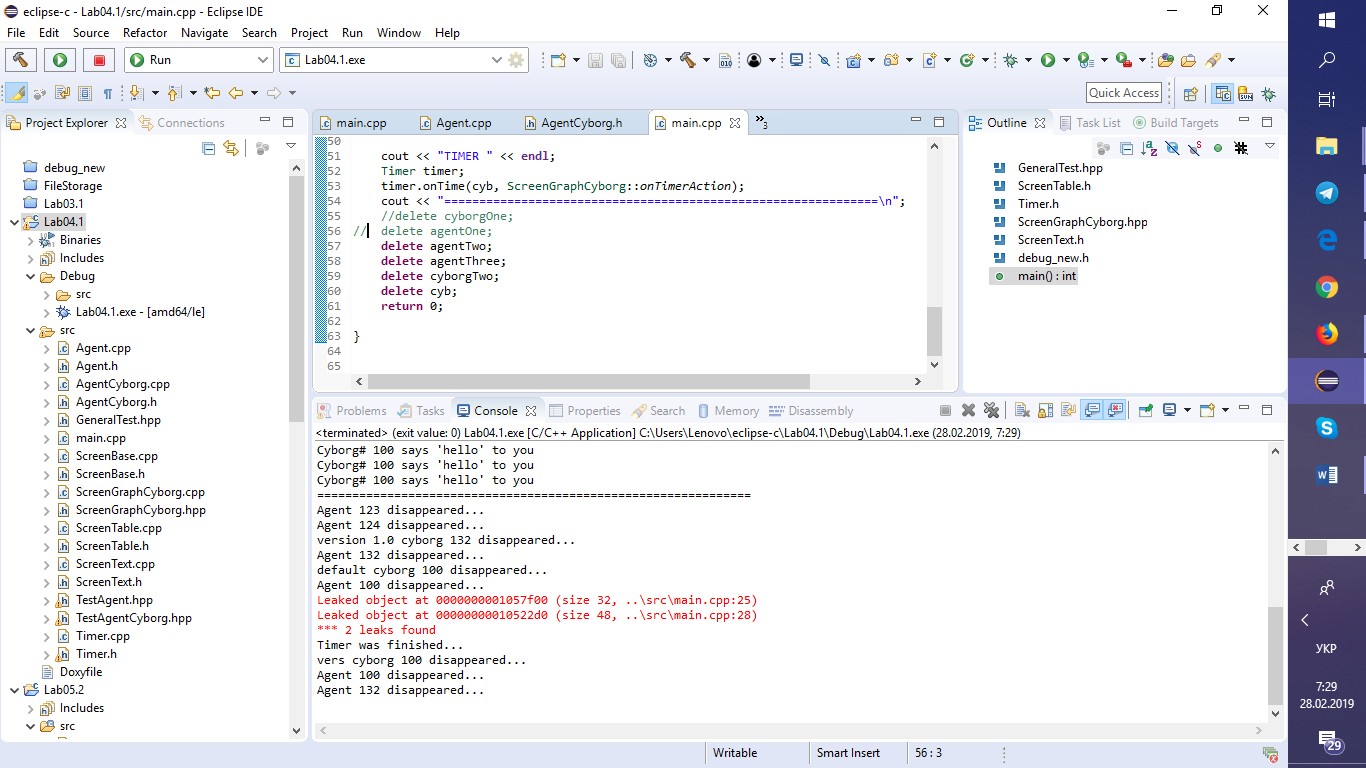


Рисунок 5 - Кількість деструкторів для 6 об’єктів

Тож можна зробити висновок, що конструктори та деструктори можуть неявно викликатися у ієрархіях таким чином, щоб виконати конструктор або деструктор суперкласу.

У випадку не викликання delete матимемо втічки пам’яті (рис.6).



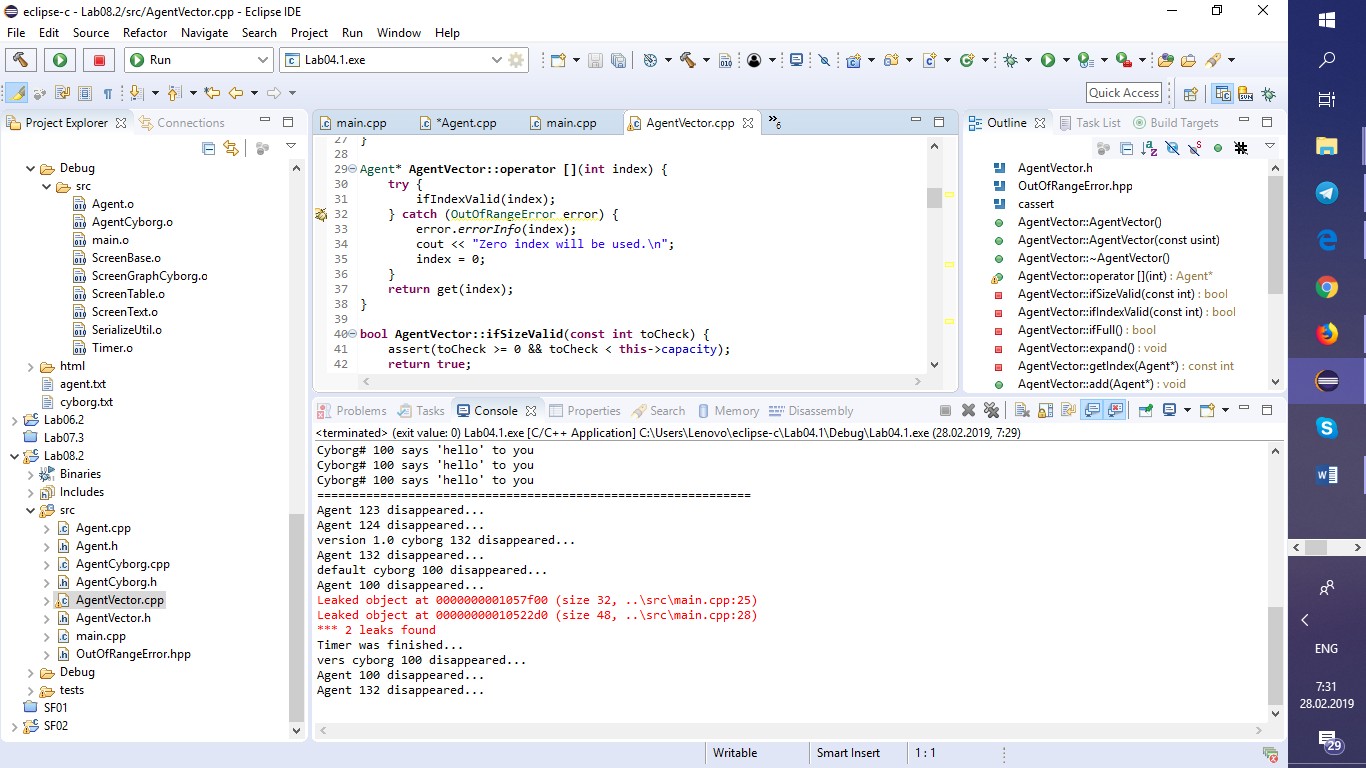
Рисунок 6 — Втічки пам’яті Обробка виключної ситуації наведена на рис.7.

Рисунок 7 — Виключна ситуація

У разі не виконання цього блоку програма мала б небезпеку завершитися з NullPointerException.

ВИСНОВКИ

В ході лабораторної роботи було досліджено конструктори та деструктори.