Чернівецький національний університет імені Юрія Федьковича

Навчально-науковий інститут фізико-технічних та комп’ютерних наук

Кафедра математичних проблем управління і кібернетики

**Звіт**

про виконання лабораторної роботи №4

«Основи Об’єктно – орієнтоване програмування мовою С++. Класи. Протокол

класу. Конструктори та деструктори.»

з дисципліни

«Об’єктно-орієнтоване програмування»

Виконав: студент 141 групи

Фрасинюк О.Б.

Перевірив: канд.фіз.-мат. наук,

доцент Лазорик В.В.

Оцінка:

Дата захисту:

Чернівці 2024

Варіант 12

Завдання 1\_2.

Створити тип даних - клас вектор VectorFloat (вектор дійсних чисел), який має

вказівник на float, число елементів size і змінну стану codeError.У класі визначити

o конструктор без параметрів( виділяє місце для одного елемента та інінціалізує

його в нуль);

o конструктор з одним параметром - розмір вектора( виділяє місце та інінціалізує

масив значенням нуль);

o конструктор із двома параметрами - розмір вектора та значення

ініціалізації(виділяє місце (значення перший аргумент) та інінціалізує значенням

другого аргументу).

o деструктор звільняє пам'ять;

o перевантаження операцій (операції перевантажувати через функції класу або

дружні функції, якщо не вказано яким чином це робити):

 унарних префіксних та постфіксних ++ та --: одночасно збільшує

(зменшує) значення елементів масиву на 1.0;

 унарної логічної ! (заперечення): повертає значення true, якщо

елементи якщо size не дорівнює – нулю, інакше false;

 унарний арифметичний - (мінус) : повертає всі елементи масиву класу

вектор з протилежним знаком;

 присвоєння =: копіює вектор(перевантажити через функцію класу);

 присвоєння з (перевантажити через функцію класу)

o += - додаванням векторів;

o -= - відніманням векторів;:

o \*= - множення, вектора на число;

o /= - ділення, вектора на число;

o %= - остача від ділення, вектора на число;

 арифметичних бінарні:

o + - додавання векторів;

o – - віднімання векторів;

o \* - множення, вектора на число;

o / - ділення, вектора на число;

o % - остача від ділення, вектора на число типу int;

 побітові операції зсувів, як дружні операції введення та виведення

вектора у потік (перевантажувати через дружні функції)

o введення >> (побітовий зсув право) ;

o введення << (побітовий зсув ліво);

 рівності == та нерівності!= , функція-операція виконує певні дії над

кожною парою елементів векторів за індексом;

 операцію індексації [] – функцію, яка звертається до елементу масиву

за індексом, якщо індекс невірний функція вказує на останній елемент

масиву та встановлює код помилки у змінну codeError;

 розподілу пам’яті new та delete;

 виклику функції ();

 порівняння (функція-операція виконує певні дії над кожною парою

елементів векторів за індексом), які повертають true або false:

o > (більше) для двох векторів;

o >= (більше рівне) для двох векторів;

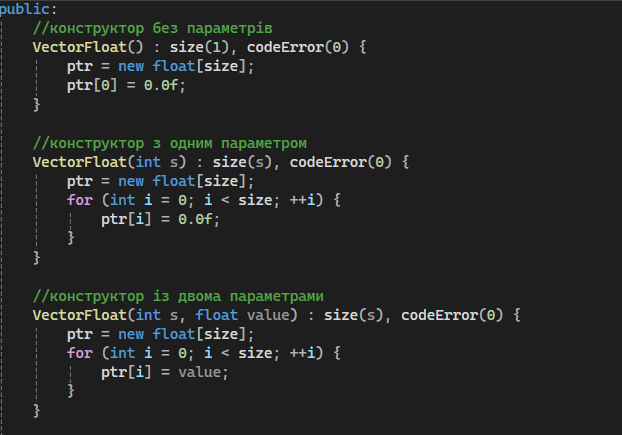
o < (менше) для двох векторів;

o <=(менше рівне) для двох векторів.

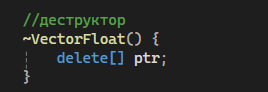
У змінну стани встановлювати код помилки, коли не вистачає пам'яті, виходить за межі масиву.

Передбачити можливість підрахунку числа об'єктів даного типу. Перевірити роботу цього класу.

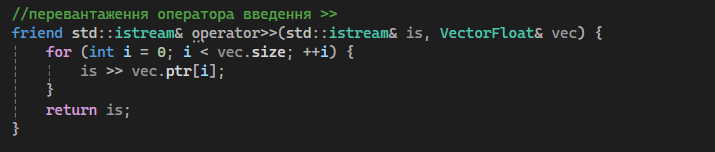
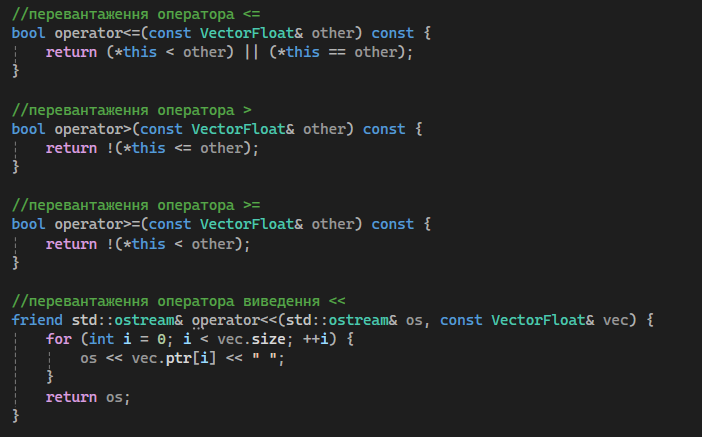
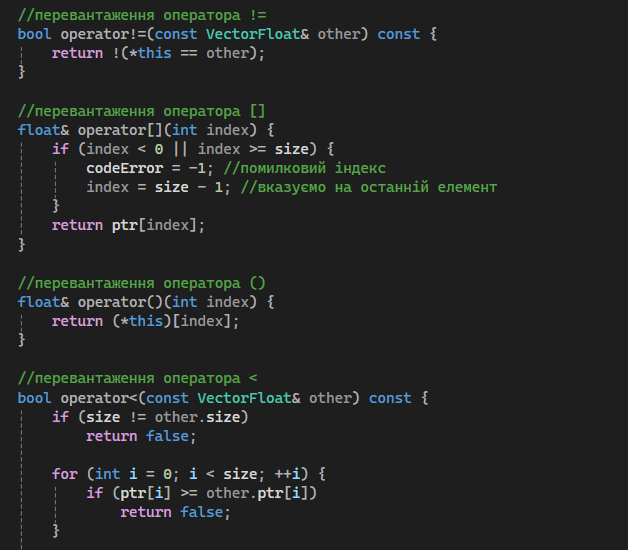
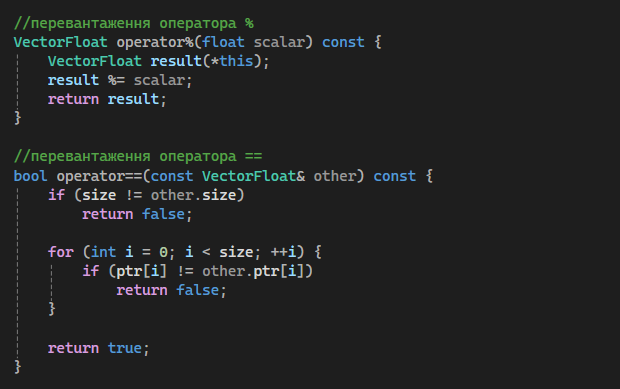
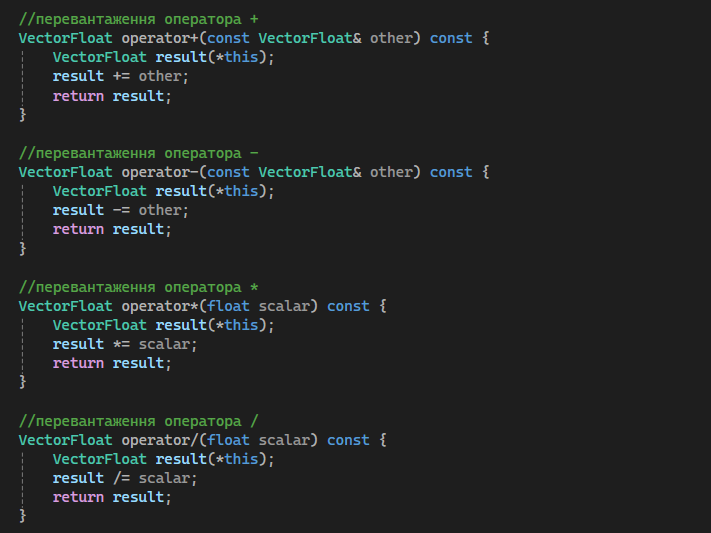
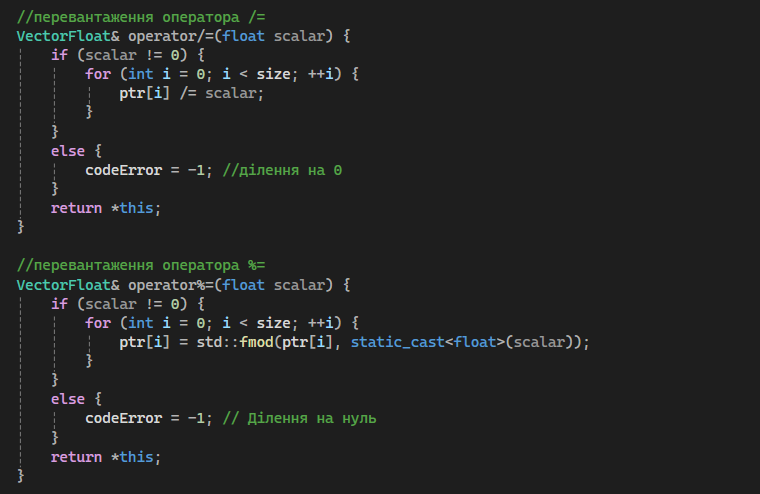
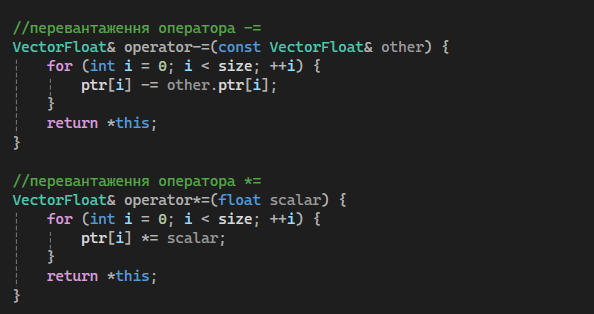
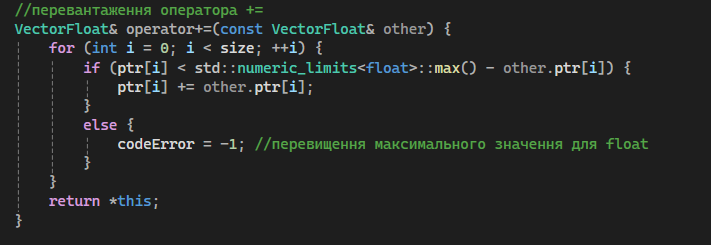
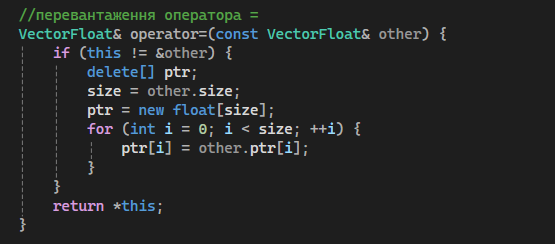
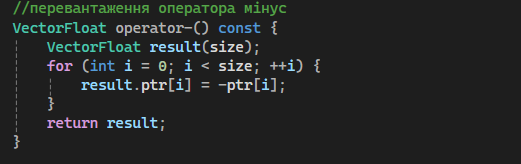
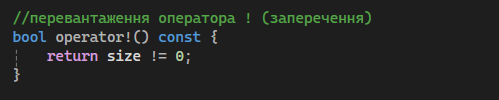
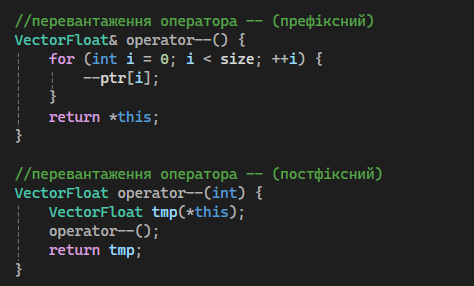
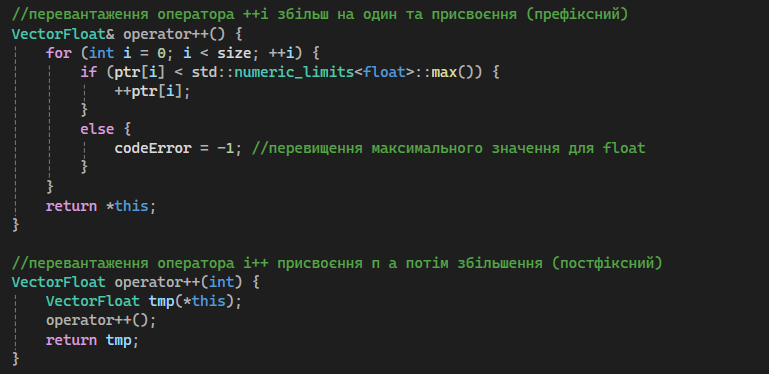
1. Створюємо коеструктори без параметрів, з 1 та 2 параметрами



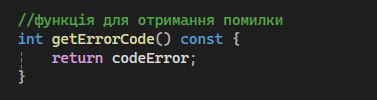
2. Створюємо деструктор



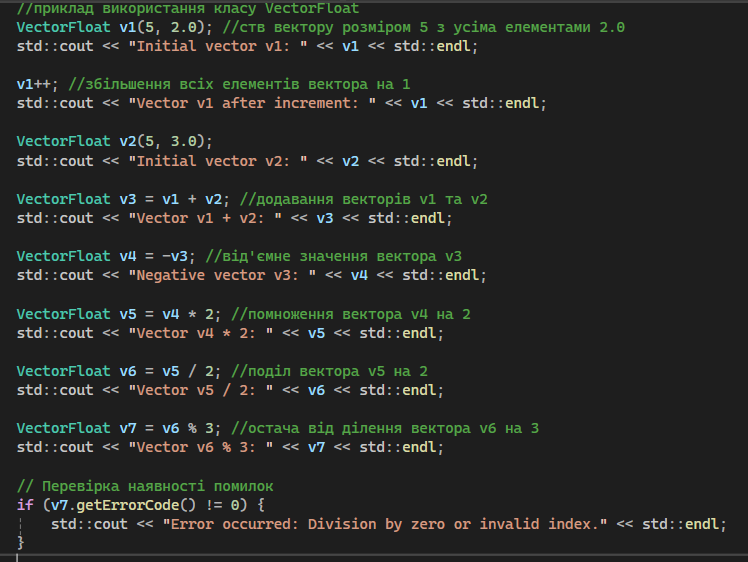
3. Створюємо перевантаження операторів **+**, **-**, **\***, **/**, **==**, **!=**, префіксний та постфіксний оператори **++** і **--**, та інші наведені у методичці



4. Функція для виведення помилки



5. Перевірка



Завдання 2\_2.

Побудувати асоційований клас збереження

двох сутностей. В завданні створити клас, який представляє собою асоціативний

масив між двома сутностями. Написати функцію створення набору

асоціативних сутностей. Перевантажити операцію індексації [] – функцію, яка

звертається до об’єкта класу, за однією сутністю, якщо індекс, повертає

асоціативну сутність, якщо відповідної сутності немає в встановлює код

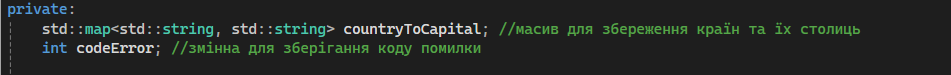
помилки у змінну CodeError, альтернативні звернення через перевантаження

операції виклику функції(); перевантажити дружні операції введення та

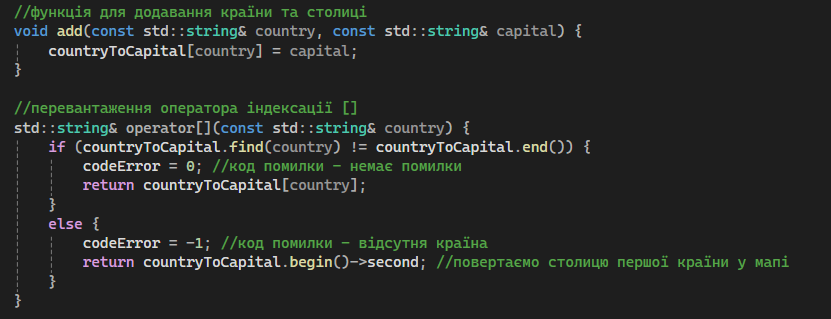
виведення.

Назви країни та столиця

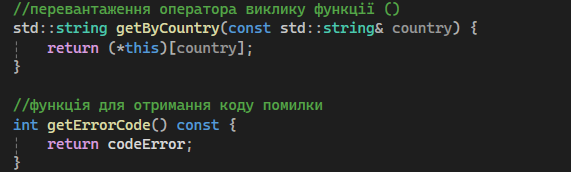
1. Створюєм масив та змінну CodeError



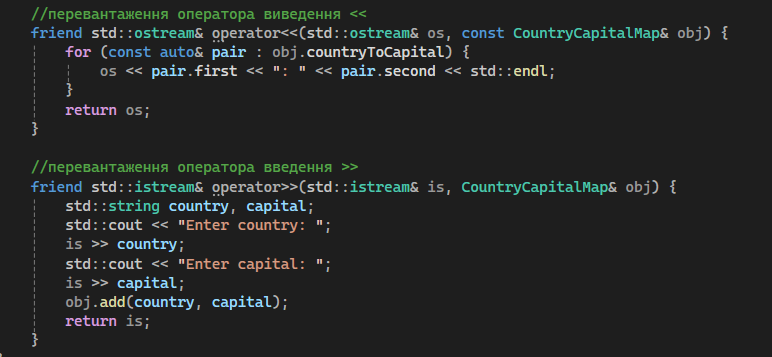
2. Створюєм функцію для додавання країни та столиці й описуємо перевантаження оператора індексації



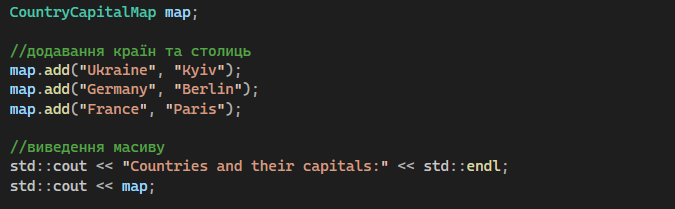
3. Виклик функції перевантаження та створення функції для отримаання коду помилки



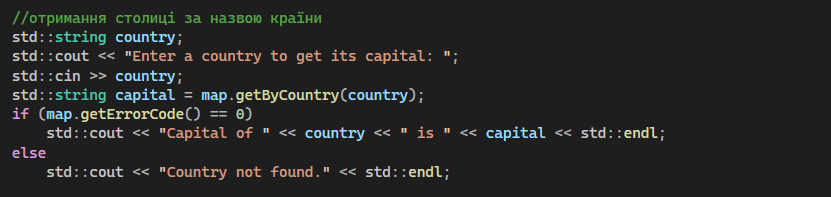
4. Перевантаження операторів введення та виведення



5. Для перевірки додаєм декілька країн та столиць до масиву та виводимо масив



6. Виведення країни за її столицею



7. Додавання країни та столиці введеною користувачем та виведення нового масиву

