

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Західноукраїнський національний університет

Факультет комп'ютерних інформаційних технологій

Кафедра ІОСУ

Лабораторна робота № 7

З дисципліни “Сучасні парадигми програмування”

Виконав

Студент групи КН-11

Стрижак В. М.

Тернопіль
2024 рік

Тема: Віртуальні функції . Шаблони класів

Мета роботи: ознайомитися із базовими механізмами використання шаблонів функцій та шаблонів класів, навчитися створювати та використовувати параметризовані функції та параметризовані контейнерні класи.

Завдання: 7.1 (8)

Контейнерний клас описує та забезпечує набір дій над даними параметризованого масиву, розмірність якого визначається під час роботи програми. Усі обчислення та перетворення повинні бути реалізовані у вигляді функцій членів класу.

Завдання: 7.2 (8)

Реалізувати контейнерний клас та необхідні функції- маніпулятори над його елементами.

Хід роботи

7.1: Дана прямокутна матриця. Визначити:

- кількість рядків, які не містять жодного нульового елемента;
- максимальне із чисел, що зустрічається в заданій матриці більше одного разу.

Перетворити матрицю таким чином, щоб всі елементи, рівні нулю, розташовувались на початку всіх інших.

Код програми:

```
#include <iostream>
#include <vector>
#include <unordered_map>
#include <algorithm>

class Matrix {
public:
    Matrix(const std::vector<std::vector<int>>& data) : data(data) {}

    int countNonZeroRows() const {
```

```

    int count = 0;
    for (const auto& row : data) {
        if (std::all_of(row.begin(), row.end(), [](int x) { return x != 0; })) {
            count++;
        }
    }
    return count;
}

int maxRepeatedElement() const {
    std::unordered_map<int, int> elementCount;
    for (const auto& row : data) {
        for (int element : row) {
            elementCount[element]++;
        }
    }

    int maxElement = -1;
    for (const auto& elem : elementCount) {
        int element = elem.first;
        int count = elem.second;
        if (count > 1 && (maxElement == -1 || element > maxElement)) {
            maxElement = element;
        }
    }
    return maxElement;
}

void moveZerosToFront() {
    for (auto& row : data) {
        for (auto& row : data) {
            size_t zeroCount = std::count(row.begin(), row.end(), 0);
            std::vector<int> nonZeroElements;
            nonZeroElements.reserve(row.size() - zeroCount);
            for (int element : row) {
                if (element != 0) {
                    nonZeroElements.push_back(element);
                }
            }
            std::fill(row.begin(), row.begin() + zeroCount, 0);
            std::copy(nonZeroElements.begin(), nonZeroElements.end(), row.begin() +
zeroCount);
        }
    }

    void print() const {
        for (const auto& row : data) {
            for (int element : row) {
                std::cout << element << " ";
            }
            std::cout << "\n";
        }
    }

private:
    std::vector<std::vector<int>> data;
};

int main() {
    std::vector<std::vector<int>> matrixData = {
        {1, 0, 3, 4},
        {5, 6, 0, 0},
        {7, 8, 9, 0},
        {0, 0, 0, 0}
    };

    Matrix matrix(matrixData);

    std::cout << "Number of rows without any zero elements: " <<
matrix.countNonZeroRows() << "\n";

```

```

        std::cout << "Maximum element repeated more than once: " <<
matrix.maxRepeatedElement() << "\n";

        matrix.moveZerosToFront();
        std::cout << "Matrix after moving zeros to the front:\n";
        matrix.print();

        return 0;
}

```

Результат коду:

```

Number of rows without any zero elements: 0
Maximum element repeated more than once: 0
Matrix after moving zeros to the front:
0 1 3 4
0 0 5 6
0 7 8 9
0 0 0 0

```

7.2: Описати клас "гаражна стоянка", що має одну лінію для стоянки автомашин. Вїзд та виїзд відбувається з одного кінця лінії. Моделювання виїзду машини з автопарку відбувається так, що техніка, яка заважає виїзду, вилучається, а потім, після виведення потрібного авто зі стоянки, повертається на місце у тому ж порядку слідування. Програма повинна виводити повідомлення про прибуття та виїзд будь-якої машини, видавати довідку про наявність конкретної машини в гаражі та відображати стан лінії для стоянки машин у поточний момент.

Код програми:

```

#include <iostream>
#include <vector>
#include <algorithm>
#include <stack>

class Garage {
public:
    // Додаємо автомобіль на стоянку
    void arrive(const std::string& car) {
        parkingLine.push_back(car);
        std::cout << "Car " << car << " has arrived." << std::endl;
    }

    // Вилучаємо автомобіль зі стоянки
    void depart(const std::string& car) {
        auto it = std::find(parkingLine.begin(), parkingLine.end(), car);
        if (it == parkingLine.end()) {
            std::cout << "Car " << car << " is not in the garage." << std::endl;
        }
    }
};

```

```

        return;
    }

    std::stack<std::string> tempStack;
    while (parkingLine.back() != car) {
        tempStack.push(parkingLine.back());
        parkingLine.pop_back();
    }
    parkingLine.pop_back();
    std::cout << "Car " << car << " has departed." << std::endl;

    while (!tempStack.empty()) {
        parkingLine.push_back(tempStack.top());
        tempStack.pop();
    }
}

// Перевіряємо наявність автомобіля на стоянці
bool isPresent(const std::string& car) const {
    return std::find(parkingLine.begin(), parkingLine.end(), car) !=
parkingLine.end();
}

// Відображаємо поточний стан стоянки
void display() const {
    std::cout << "Current parking line: ";
    for (const auto& car : parkingLine) {
        std::cout << car << " ";
    }
    std::cout << std::endl;
}

private:
    std::vector<std::string> parkingLine; // Контейнер для зберігання автомобілів
};

int main() {
    Garage garage;

    garage.arrive("Car1");
    garage.arrive("Car2");
    garage.arrive("Car3");

    garage.display();

    garage.depart("Car2");
    garage.display();

    std::cout << "Is Car3 in the garage? " << (garage.isPresent("Car3") ? "Yes" :
"No") << std::endl;

    garage.arrive("Car4");
    garage.display();

    garage.depart("Car1");
    garage.display();

    return 0;
}

```

Результат коду:

```
Car Car1 has arrived.  
Car Car2 has arrived.  
Car Car3 has arrived.  
Current parking line: Car1 Car2 Car3  
Car Car2 has departed.  
Current parking line: Car1 Car3  
Is Car3 in the garage? Yes  
Car Car4 has arrived.  
Current parking line: Car1 Car3 Car4  
Car Car1 has departed.  
Current parking line: Car3 Car4
```

Висновок: Я ознайомився із базовими механізмами використання шаблонів функцій та шаблонів класів, навчився створювати та використовувати параметризовані функції та параметризовані контейнерні класи.