Фаховий коледж ракетно-космічного машинобудування

Дніпровського національного університету імені Олеся Гончара

ЗВІТ

з лабораторних робіт

з дисципліни «Технології (Програмування)»

Спеціальність 123 Комп’ютерна інженерія

Група КС-21-1

Виконав К.І. Антіпов

Перевірив М.М. Гапоненко

2022-2023

ЗМІСТ

Лабораторна робота № 21

Лабораторна робота №21

*Перевірки даних на допустимість. Проектування конструкторів та деструктора. Конструктор за замовчанням. Розробка класу з елементом-масивом та його конструкторів. Розробка методів класу для одномірного елементу-масиву*

*Мета: закріпити навички з реалізації перевірок даних на допустимість в класі, навчитись проектувати конструктори, в тому числі за замовчуванням, та деструктор, розробляти клас з членом класу одномірним масивом та відповідні методи для роботи з ним*

*В завданні необхідно виконати наступне. Користувальницький клас Array повинен працювати з одновимірним числовим масивом і містити поля , одне з яких - значення розмірності масиву, друге поле задає масив. Поля повинні бути закритими. Масив створюється в пам'яті динамічно. Клас містить методи : конструктори за замовчуванням та з параметрами, де створюється динамічний масив ( операція new для створення) , деструктор , в якому виконується звільнення пам'яті. Крім цього, клас містить також методи :*

*виведення на екран поточного стану масиву: void print (... ) ;*

*установлення ( переустановлення ) поточного стану та кiлькості елементiв масиву: void set (... ) ;*

*методи, які виконують дії з масивами, зазначені для кожного варіанта (якщо завдання вирішити не можна, то повідомте програмно про це).*

*Запропонувати користувачу можливість повторного виконання програми з іншою розмірністю масиву.*

*У звіті продемонструвати роботу програми у випадку, якщо є відповідні елементи в масиві, якщо є один, якщо немає, тощо.*

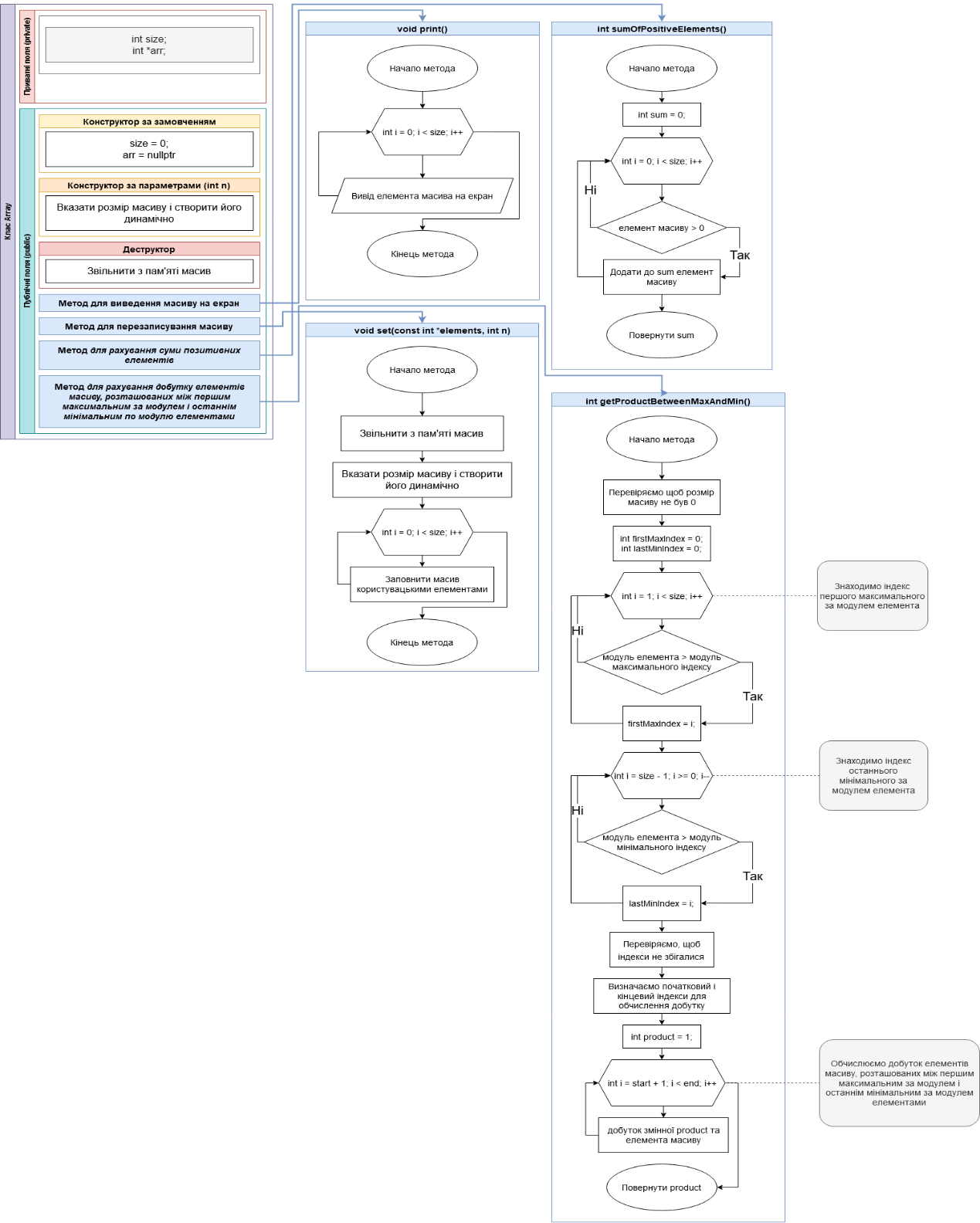
3. Варіант

*Обчислити:*

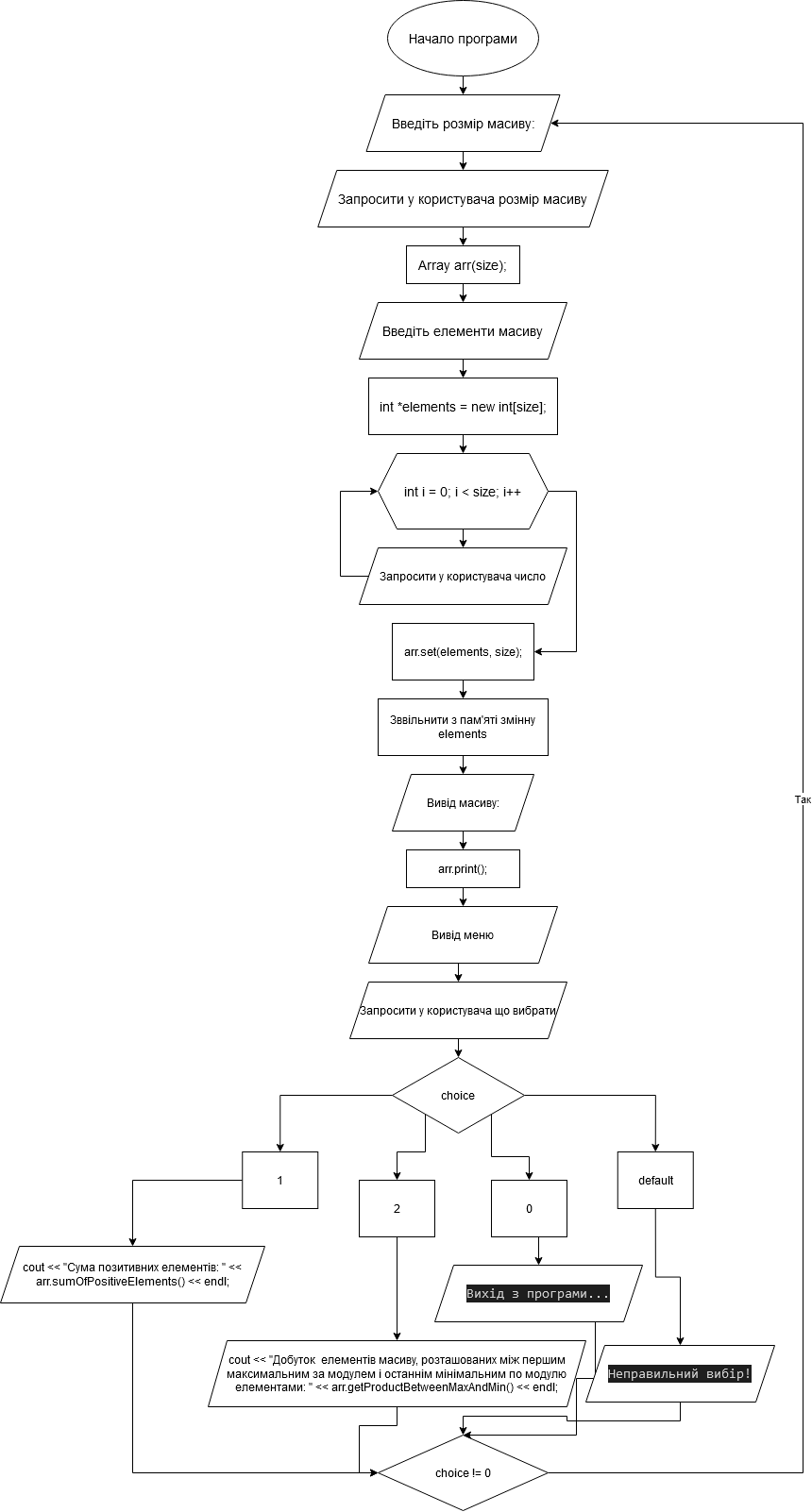
*1. Суму позитивних елементів масиву.*

*2. Добуток елементів масиву, розташованих між першим максимальним за модулем і останнім мінімальним по модулю елементами.*

2 Блок-схема алгоритму рішення задачі.



Блок-схема 1 – Клас Array



Блок-схема 2 – програма

3 Текст програми мовою програмування С++.

#include <iostream>

#include <cmath>

#include <windows.h>

using namespace std;

class Array

{

private:

    int size;

    int \*arr;

public:

    Array()

    {

        size = 0;

        arr = nullptr;

    }

    Array(int n)

    {

        size = n;

        arr = new int[size];

    }

    ~Array()

    {

        delete[] arr;

    }

    void print()

    {

        for (int i = 0; i < size; i++)

        {

            cout << arr[i] << " ";

        }

        cout << endl;

    }

    void set(const int \*elements, int n)

    {

        delete[] arr;

        size = n;

        arr = new int[size];

        for (int i = 0; i < size; i++)

        {

            arr[i] = elements[i];

        }

    }

    int sumOfPositiveElements()

    {

        int sum = 0;

        for (int i = 0; i < size; i++)

        {

            if (arr[i] > 0)

            {

                sum += arr[i];

            }

        }

        return sum;

    }

    int getProductBetweenMaxAndMin()

    {

        // Перевіряємо щоб розмір масиву не був 0

        if (size == 0)

        {

            return 0;

        }

        int firstMaxIndex = 0;

        int lastMinIndex = 0;

        // Знаходимо індекс першого максимального за модулем елемента

        for (int i = 1; i < size; i++)

        {

            if (abs(arr[i]) > abs(arr[firstMaxIndex]))

            {

                firstMaxIndex = i;

            }

        }

        // Знаходимо індекс останнього мінімального за модулем елемента

        for (int i = size - 1; i >= 0; i--)

        {

            if (abs(arr[i]) < abs(arr[lastMinIndex]))

            {

                lastMinIndex = i;

            }

        }

        // Перевіряємо, щоб індекси не збігалися

        if (firstMaxIndex == lastMinIndex)

        {

            return 0;

        }

        // Визначаємо початковий і кінцевий індекси для обчислення добутку

        int start = min(firstMaxIndex, lastMinIndex);

        int end = max(firstMaxIndex, lastMinIndex);

        // Обчислюємо добуток елементів масиву, розташованих між першим максимальним за модулем і останнім мінімальним за модулем елементами

        int product = 1;

        for (int i = start + 1; i < end; i++)

        {

            product \*= arr[i];

        }

        return product;

    }

};

int main()

{

    int choice;

    int size;

    SetConsoleCP(65001);       // встановлюємо кодування

    SetConsoleOutputCP(65001); // встановлюємо кодування

    do

    {

        cout << "Введіть розмір масиву: ";

        cin >> size;

        Array arr(size);

        cout << "Введіть елементи масиву: ";

        int \*elements = new int[size];

        for (int i = 0; i < size; i++)

        {

            cin >> elements[i];

        }

        arr.set(elements, size);

        delete[] elements;

        cout << "Вивід масиву: ";

        arr.print();

        cout << "Меню:" << endl;

        cout << "1. Сума позитивних елементів" << endl;

        cout << "2. Добуток  елементів масиву, розташованих між першим максимальним за модулем і останнім мінімальним по модулю елементами" << endl;

        cout << "Введіть свій вибір (0 для виходу з програми): ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            cout << "Сума позитивних елементів: " << arr.sumOfPositiveElements() << endl;

            break;

        case 2:

            cout << "Добуток  елементів масиву, розташованих між першим максимальним за модулем і останнім мінімальним по модулю елементами: " << arr.getProductBetweenMaxAndMin() << endl;

            break;

        case 0:

            cout << "Вихід з програми..." << endl;

            break;

        default:

            cout << "Неправильний вибір!" << endl;

            break;

        }

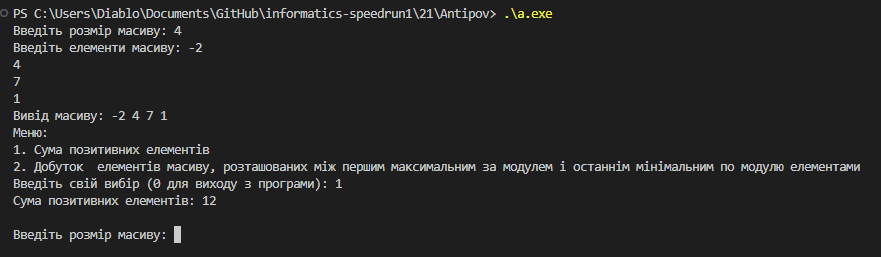
        cout << endl;

    } while (choice != 0);

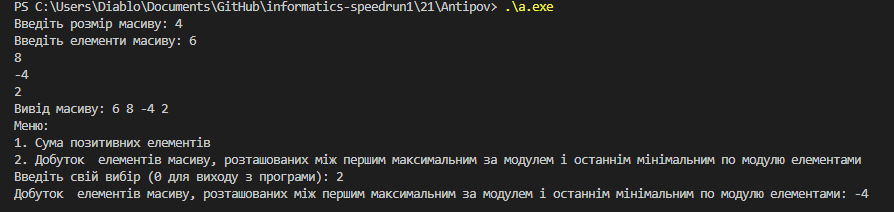
    return 0;

}

4 Копія вікна виконання програми



Виконання програми 1 – Сума позитивних елементів



Виконання програми 2 – Добуток елементів масиву

5 Висновок

Під час виконання лабораторної роботи закріпили навички з реалізації перевірок даних на допустимість в класі, навчитись проектувати конструктори, в тому числі за замовчуванням, та деструктор, розробляти клас з членом класу одномірним масивом та відповідні методи для роботи з ним.