**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЁТ**

**По лабораторной работе №2**

**По дисциплине «Программирование»**

Студент гр. 3351 \_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Рыжов Н.Г.

Санкт-Петербург

2024

цель работы

Используя указатели написать программу, формирующую динамический список, содержащий координаты точек на экране. Длина списка (количество точек) и координаты точек задаются пользователем. Программа должна позволять удалять любую точку с заданным пользователем номером и вставлять новую точку на заданную позицию в списке. Совпадение координат точек не допускается. Сформировать с помощью библиотеки **SFML** на экране изображение списка (до и после удаления точки), состоящее из окружностей произвольного радиуса с центрами, совпадающими с координатами точек, и последовательно соединенными линиями кратчайшей длины. Линии должны быть нарисованы от края окружности, а не от центра. Выполнить задание без использования контейнерных классов.

**Текст программы**

#include <iostream>

#include <SFML/Graphics.hpp>

using namespace std;

using namespace sf;

const int WIDTH = 1280, HEIGHT = 800; // Ширина и высота окна

const int RADIUS = 20; // Радиус окружностей

const Color LINE\_COLOR = Color::White, BACKGROUND = Color::Black, OUTLINE\_COLOR = Color::Yellow; // Цвета линий, фона и контура окружностей

const int OUTLINE\_SIZE = 2; // Толщина контура окружностей

// Структура точки

struct Point {

float x; // Координата x

float y; // Координата y

Point\* next; // Указатель на следующую точку

};

Point\* getElement(unsigned int index, Point\* head) {

Point\* current = head; // Текущая точка

for (int i = 0; i < index; i++) {

current = current->next; // Переход к следующей точке

}

return current; // Возвращение точки по индексу

}

bool isDuplicate(Point\* head, float x, float y) { // Функция проверки повторения точки

for (Point\* current = head; current != NULL; current = current->next) { // Перебор всех точек

if (current->x == x && current->y == y) return true; // Если найдена повторяющаяся точка, возвращаем true

}

return false; // Если повторений не найдено, возвращаем false

}

void drawCircles(RenderWindow& window, Point\* head) { // Функция рисования окружностей

for (Point\* current = head; current != NULL; current = current->next) { // Перебор всех точек

CircleShape circle(RADIUS); // Создание окружности

circle.setPosition(current->x - RADIUS, current->y - RADIUS); // Установка позиции окружности

circle.setFillColor(BACKGROUND); // Установка цвета заливки окружности

circle.setOutlineThickness(OUTLINE\_SIZE); // Установка толщины контура окружности

circle.setOutlineColor(OUTLINE\_COLOR); // Установка цвета контура окружности

window.draw(circle); // Рисование окружности

}

}

void drawLines(RenderWindow& window, Point\* head) { // Функция рисования линий между точками

for (Point\* fixedPoint = head; fixedPoint != NULL; fixedPoint = fixedPoint->next) { // Перебор всех точек

for (Point\* current = head; current->next != NULL; current = current->next) { // Перебор всех точек, кроме последней

VertexArray line(Lines, 2); // Создание линии

line[0].color = LINE\_COLOR; // Установка цвета первой вершины линии

line[1].color = LINE\_COLOR; // Установка цвета второй вершины линии

line[0].position = Vector2f(fixedPoint->x, fixedPoint->y); // Установка позиции первой вершины линии

line[1].position = Vector2f(current->next->x, current->next->y); // Установка позиции второй вершины линии

window.draw(line); // Рисование линии

}

}

}

void printSummary(Point\* head) { // Функция вывода сводки данных

int index = 0;

for (Point\* current = head; current != NULL; current = current->next) { // Перебор всех точек

cout << "Точка " << index << ": (" << current->x << ", " << current->y << ")" << endl; // Вывод координат точки

index++;

}

}

int main() {

setlocale(LC\_ALL, "rus");

int numPoints;

cout << "Введите количество точек: " << endl;

cin >> numPoints;

Point\* head = NULL; // Указатель на первую точку

Point\* tail = NULL; // Указатель на последнюю точку

for (int i = 0; i < numPoints;) {

float x, y;

cout << "Введите координаты точки " << i << ":\nX = ";

cin >> x;

cout << "Y = ";

cin >> y;

if (!isDuplicate(head, x, y)) { // Если точка не повторяется

Point\* newPoint = new Point; // Создание новой точки

newPoint->x = x; // Установка координаты x

newPoint->y = y; // Установка координаты y

newPoint->next = NULL; // Сброс указателя на следующую точку

if (tail != NULL) tail->next = newPoint; // Если есть предыдущая точка, установка указателя на новую точку

tail = newPoint; // Установка указателя на новую точку как на последнюю

if (i == 0) head = newPoint; // Если это первая точка, установка указателя на нее как на первую

i++;

}

else {

cout << "Копировать точку" << endl;

}

}

// Печатаем сводку после ввода всех координат

printSummary(head);

RenderWindow window(VideoMode(WIDTH, HEIGHT), L"Circles"); // Создание окна SFML

// Главный цикл взаимодействия с точками

while (true) {

int action;

cout << "\nВыберите действие:\n1. Перезаписать координаты\n2. Добавить точку\n3. Удалить точку\n4. Вывести сводку\n5. Выйти\nВаш выбор: ";

cin >> action;

if (action == 1) {

int index;

float newX, newY;

cout << "Введите номер точки, которую хотите перезаписать: ";

cin >> index;

if (index >= numPoints) {

cout << "Неверный ввод" << endl;

continue;

}

Point\* point = getElement(index, head); // Получение точки по индексу

cout << "Введите новые координаты:\nX = ";

cin >> newX;

cout << "Y = ";

cin >> newY;

if (isDuplicate(head, newX, newY)) { // Если новые координаты повторяются

cout << "Копировать точку" << endl;

continue;

}

point->x = newX; // Установка новых координат

point->y = newY;

}

else if (action == 2) {

float x, y;

cout << "Введите координаты новой точки:\nX = ";

cin >> x;

cout << "Y = ";

cin >> y;

if (!isDuplicate(head, x, y)) { // Если новые координаты не повторяются

Point\* newPoint = new Point; // Создание новой точки

newPoint->x = x; // Установка координаты x

newPoint->y = y; // Установка координаты y

newPoint->next = NULL; // Сброс указателя на следующую точку

if (tail != NULL) tail->next = newPoint; // Если есть предыдущая точка, установка указателя на новую точку

tail = newPoint; // Установка указателя на новую точку как на последнюю

numPoints++; // Увеличение количества точек

}

else {

cout << "Копировать точку" << endl;

continue;

}

}

else if (action == 3) {

int index;

cout << "Введите номер точки, которую необходимо удалить: ";

cin >> index;

if (index >= numPoints) {

cout << "Неверный ввод" << endl;

continue;

}

if (index == 0) {

Point\* temp = head; // Временный указатель на первую точку

head = head->next; // Установка указателя на первую точку как на следующую

delete temp; // Удаление первой точки

}

else {

Point\* prev = getElement(index - 1, head); // Получение предыдущей точки

Point\* current = prev->next; // Получение текущей точки

prev->next = current->next; // Установка указателя на следующую точку

if (current == tail) tail = prev; // Если удаляется последняя точка, установка указателя на предыдущую точку как на последнюю

delete current; // Удаление текущей точки

}

numPoints--; // Уменьшение количества точек

}

else if (action == 4) {

printSummary(head); // Вывод сводки данных

}

else if (action == 5) {

break;

}

else {

cout << "Неверное действие" << endl;

continue;

}

window.clear(BACKGROUND); // Очистка окна

drawLines(window, head); // Рисование линий

drawCircles(window, head); // Рисование окружностей

window.display(); // Отображение окна

}

return 0;

}

**Описание программы**

Программное обеспечение – Microsoft Visual Studio 2022. Язык программирования – C++. Программа принимает на вход количество точек и их координаты, а также строит окружности заданного радиуса и цвета с помощью библиотеки SFML. Информация о всех точках хранится в односвязном списке, информация о каждой точке задана структурой Point. После ввода точек программа выдаёт сводку данных, которая содержит компактную информацию о всех точках. Далее программа показывает меню, в котором можно создать новую точку, удалить точку, перезаписать координаты точки, повторно вывести сводку данных. Функции построения окружностей и линий между ними разбиты по отдельности для простоты восприятия программы.

**Руководство пользователя**

Программа просит пользователя ввести количество точек для построения окружностей. После этого пользователю необходимо задать координаты X и Y для каждой точки. После ввода всех данных программа выводит сводку, содержащую компактную информацию о всех точках. Создаётся окно SFML, в котором строятся окружности и линии между ними. (Рис. 1)

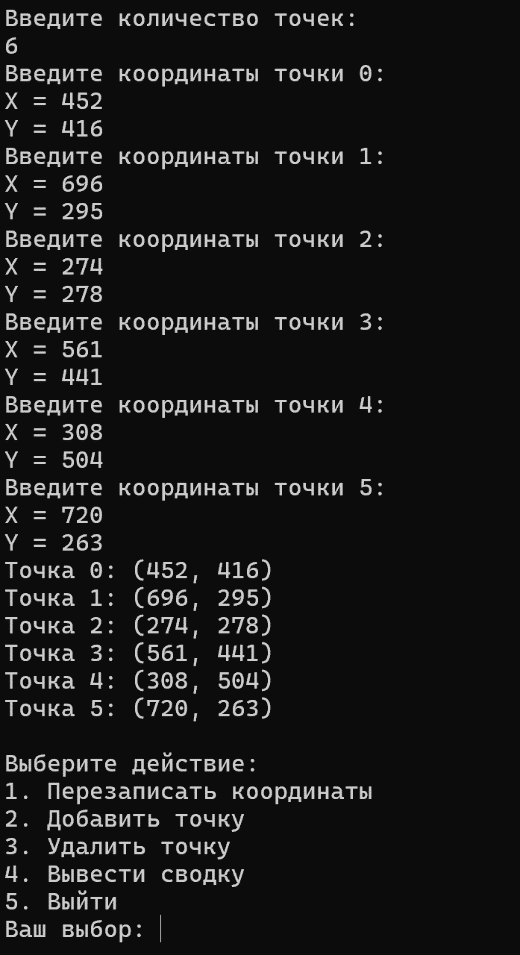


Рис. 1 – Задание шести точек

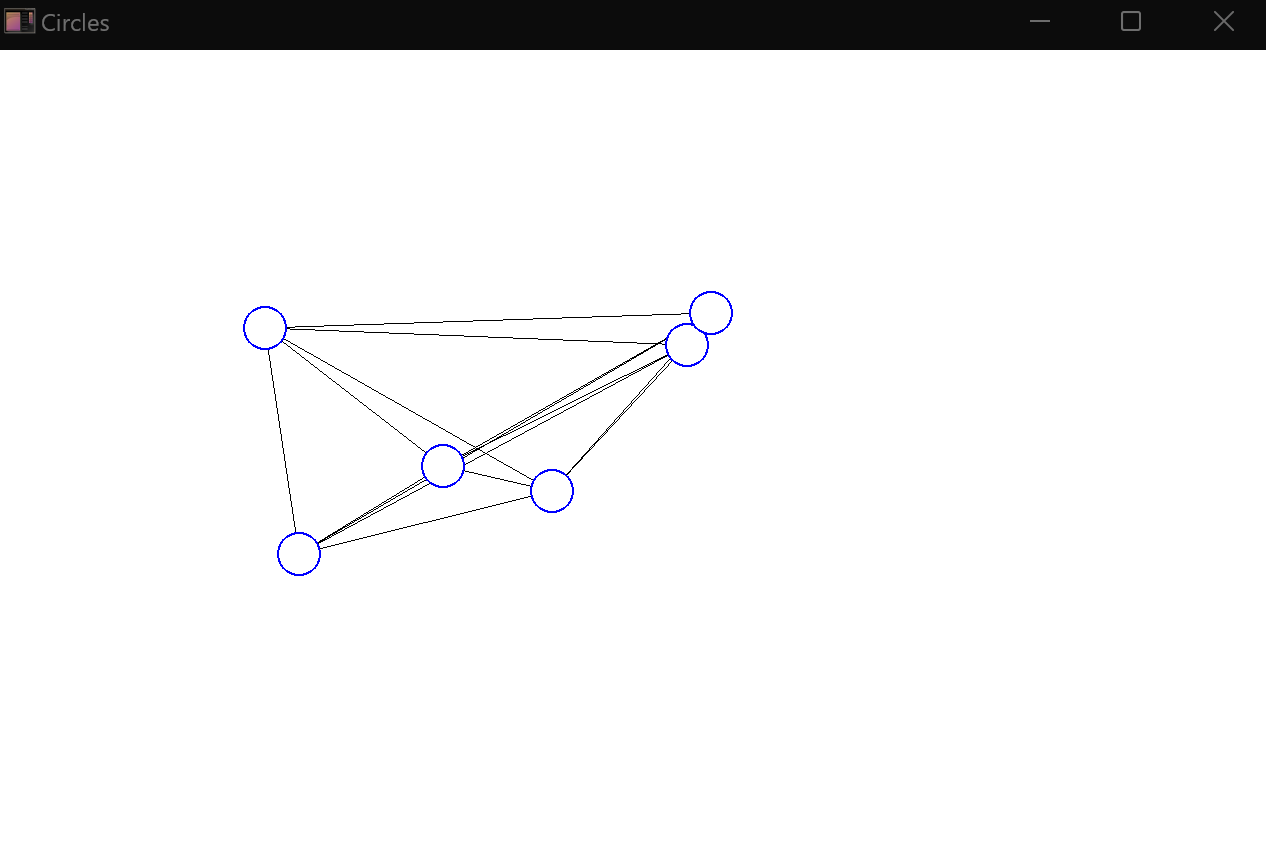


Рис. 2 – Отображение точек

В конце программа выводит меню работы с точками. Координаты каждой точки можно перезаписать, также позиции можно добавить или удалить. Ещё можно повторно вывести сводку данных и выйти из программы. (Рис. 2)

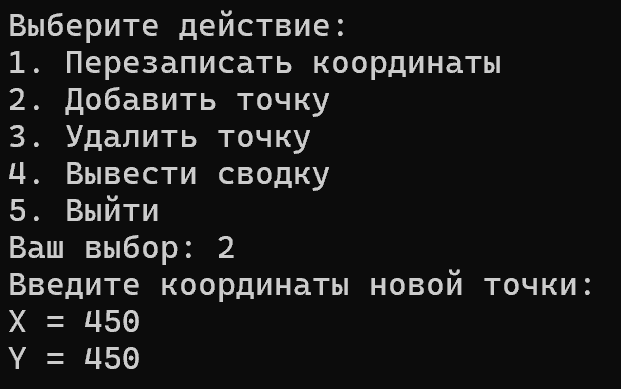


Рис. 3 – Создание новой точки

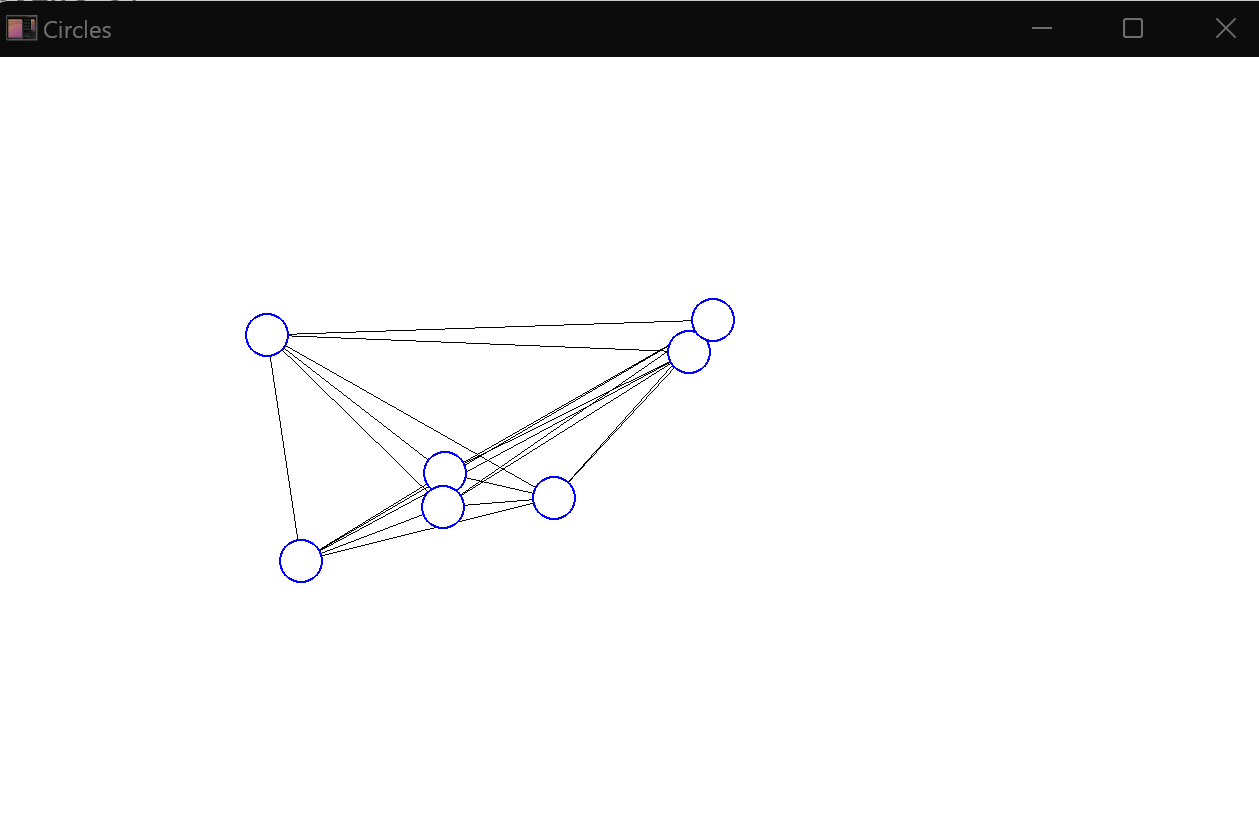


Рис. 4 – Отображение новой формации окружностей

Меню для редактирования точек будет появляться, пока пользователь не выйдет из программы. Соответственно вносить изменения можно большое количество раз.

**Пути дальнейшего улучшения программы**

1. Улучшение визуальной части программы
2. Создание новых фигур