МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Нижегородский государственный университет им. Н.И. Лобачевского»**

**Национальный исследовательский университет**

**Институт информационных технологий, математики и механики**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

**«Структура хранения данных: плекс»**

**Выполнил:** студент группы 381706-1

Силенко Дмитрий Игоревич

\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

аспирант каф. МОСТ ИИТММ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лебедев И.Г.

Нижний Новгород

2019.

Содержание

[1. Введение 3](#_Toc533027637)

[2. Постановка задачи 4](#_Toc533027638)

[3. Руководство пользователя 5](#_Toc533027639)

[4. Руководство программиста 6](#_Toc533027640)

4.[1. Описание структуры программы 6](#_Toc533027641)

4.[2. Описание структур данных 6](#_Toc533027642)

4.[3. Описание алгоритмов 7](#_Toc533027643)

[5. Заключение 8](#_Toc533027644)

[6. Литература 9](#_Toc533027645)

# **Введение**

**Основная цель данной работы** – разработка динамической структуры данных – плекса. Но для начала необходимо разобраться, что вообще такое плекс и как он работает.

Под **плексом** в рамках этой работы называют графическое изображение моделируемого объекта, характеризующее его форму и размеры и выполненное по определенным правилам проектирования с применением общепринятых соглашений по изображению и обозначению графических элементам.

В качестве плекса будем рассматривать плоский геометрический объект, состоящий из отрезков прямых (линий) и граничных точек этих линий (представление других геометрических элементов может рассматриваться как тема дополнительных заданий). Набор линий, образующих плекс, должен быть связным, т.е. любая линия плекса должна иметь общую точку хотя бы с одной другой линией плекса**.**

# **Постановка задачи**

Выполнение работы предполагает решение следующих задач:

1. Разработка и реализация классов точка, отрезок, круг, плекс – TPoint,TSection, TCircle, TPlex.
2. Пример программы, демонстрирующей работу класса TPlex.
3. Написание набора автоматических тестов с использованием Google C++ Testing Framework и проверка работоспособности методов классов.

# **Руководство пользователя**

При запуске программы, пользователю предлагается ввести последовательно координаты двух точек, которые будут в дальнейшем активно использоваться. После ввода каждой из них, сформированные точки выводятся на экран для проверки.

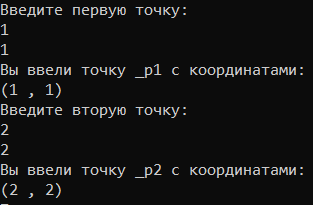


Рисунок 1Ввод точек

Далее, на основе введенных точек формируется отрезок и также выводится на экран. После чего, выводится одна из стандартных точек и пользователю предлагается ввести радиус будущего круга с центром в этой точке.

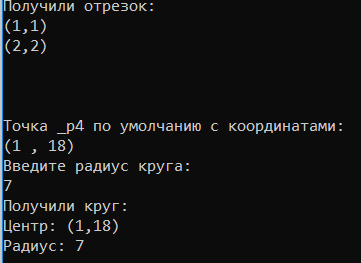


Рисунок 2 Формирование отрезка и круга

Затем, начинается построение плекса. В него последовательно добавляются две введенные в начале пользователем точке, две стандартные точки, круг, созданный пользователем, а так же еще одна точка, которую пользователю необходимо ввести.

Полученный плекс выводится на экран.

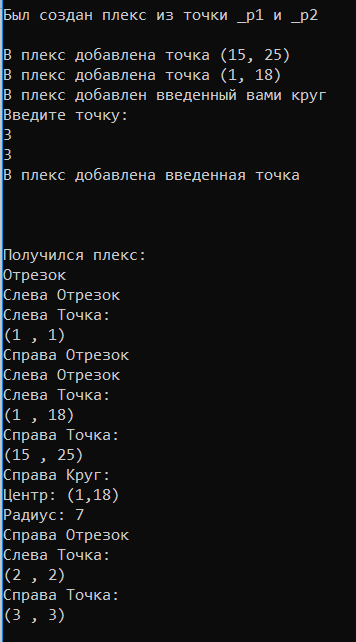


Рисунок 3 Создание плекса и вывод его на экран

# **Руководство программиста**

## **Описание структуры программы**

Программа состоит из следующих модулей:

* Модуль PlexLib. Статическая библиотека. Включает в себя заголовочные файлы PointLib.h, SectionLib.h, CircleLib.h, PlexLib.h, в которых описаны методы с реализаций шаблонных классов *TPoint, TSection, TCircle, TPlex.*
* Модуль PlexTest. Набор тестов для класса TPlex. Включает в себя файл test\_plex*.cpp.* Разработаны они с помощью использования Google C++ Testing Framework.
* Модуль Plex. Пример использования списка. Включает в себя файл с реализацией mainPlex*.cpp*

## **Описание структур данных**

*Класс TPoint:*

Поля:

double x, y – координаты точки

Конструкторы:

TPoint(double \_x = 0, double \_y = 0);

TPoint(TPoint &point);

Методы:

void SetX(double \_x); - задать координату X

void SetY(double \_y); - задать координату Y

double GetX(); - получить координату X

double GetY(); - получить координату Y

virtual void show(); - вывести точку на экран

bool operator == (TPoint &point); - проверка на равенство

TPoint& operator = (TPoint &point); - присваивание

*Класс TSection:*

Поля:

TPoint point1, point2 – Точки на концах отрезка

Конструкторы:

TSection(TPoint \_point1, TPoint \_point2);

TSection(TSection &sect);

Методы:

void SetPoint1(TPoint \_point1); - задать первую точку

void SetPoint2(TPoint \_point2); - задать вторую точку

TPoint GetPoint1(); - получить первую точку

TPoint GetPoint2(); - получить вторую точку

virtual void show(); - вывести отрезок на экран

*Класс TCircle:*

Поля:

TPoint mid; – центр круга

Double r; - радиус круга

Конструкторы:

TCircle(TPoint \_mid, double \_r);

TCircle(TCircle &circle);

Методы:

void SetMid(TPoint \_mid); - задать центр круга

void SetRadius(double \_r); - задать радиус круга

TPoint GetMid(); - получить центр круга

double GetRadius(); - получить радиус круга

virtual void show(); - вывести круг на экран

*Класс TPlex:*

Поля:

TPoint \*l, \*r; - указатель на левую и правую точки плекса

double rad;

Конструкторы:

TPlex();

TPlex(TPoint \*\_l, TPoint \*\_r);

TPlex(TPlex& plex);

Методы:

void SetL(TPoint \*\_l); - задать левую точку плекса

void SetR(TPoint \*\_r); - задать правую точку плекса

void SetRad(double \_rad);

TPoint\* GetL(); - получить левую точку плекса

TPoint\* GetR(); - получить правую точку плекса

double GetRad();

virtual void Show(); - вывести плекс на экран

void Add(TPoint \*\_point1, TPoint \*\_point2); - добавить точку в плекс

void Add(TPoint \*mid, double rad, TPoint \*point);

TPoint\* Search(TPoint \*point); - найти точку в плексе

## **Описание алгоритмов**

Метод **Search**:

Если левая точка равна искомой, возвращаем указатель на неё. Аналогично поступаем с правой точкой. Иначе рекурсивно выполняем метод Search для левого и правого плекса. В случае если точка не будет найдена, бросаем исключение.

Метод **Add**:

Если один из параметров функции равен левой точке – левая точка становится плексом (новая точка добавляется справа). В случае с точкой справа, новая точка добавляется слева. Иначе рекурсивно выполняем метод Add для левого и правого плексов. В случае если точка не будет найдена, бросаем исключение.

Метод **Show:**

Начинаем вывод плекса с отрезка. Если слева находится плекс, то рекурсивно выполняем метод Show для него. Если слева находится точка, то выводим её координаты на экран и переходим на правую ветку плекса.

# **Заключение**

Эта лабораторная работа дала возможность более детально разобраться с устройством работы плекса. В ходе выполнения, был реализован ряд вспомогательных шаблонных классов: точки TPoint, отрезка TSection, круга TCircle, а также основной шаблонный класс плекса TPlex, описанный в специально разработанной библиотеке PlexLib. Все это позволило нам реализовать ряд методов для работы с плексами.

Помимо этого, для проверки работоспособности всех методов, были написаны автоматические тесты, реализованные с использованием Google C++ Testing Framework. Пример использования списка для пользователя также написан и успешно работает.

# **Литература**

1. Лабораторный практикум. Составители: Барышева И.В., Мееров И.Б., Сысоев А.В., Шестакова Н.В. Под редакцией Гергеля В.П. Учебно-методическое пособие. – Нижний Новгород: Нижегородский госуниверситет, 2017. – 105с.

URL: <http://www.unn.ru/books/met_files/Pract_ADS.pdf>

1. Национальный открытый университет «Интуит». Курс «Основы программирования». Лекция 11: Структуры данных: общее понятие, реализация. Простейшие структуры данных: очередь, стек. Использование стека и обратная польская запись.

URL: <https://www.intuit.ru/studies/courses/2193/67/lecture/1980?page=3>