Приложение

к договору № \_\_\_\_

|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** |  |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В.Старостин**  **«\_\_\_» \_\_\_\_\_\_\_\_\_ 2022 г.** |  |

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на опытно-конструкторскую работу**

**«Разработка базового функционала библиотеки 2D геометрического ядра»**

**№ 02068143.00221**

г. Н. Новгород, 2022

# 1. Наименование НИОКР

Наименование темы НИОКР — «разработка базового функционала библиотеки 2D геометрического ядра».

Сокращённое наименование — «gkernel2d».

# 2. Срок выполнения НИОКР

1. Начало — 03.10.2022.
2. Окончание — 15.01.2023.

# 3. Исполнитель

Студенты группы 3821М1ПИ Нижегородского государственного университета им. Н. И. Лобачевского» (ННГУ): Желтов Сергей, Лямин Роман, Шеин Илья, Новрузов Ильяс.

# 4. Основания для разработки

Данная работа выполняется в рамках дисциплины «Информационные технологии в области принятия решений», а также в рамках гранта передовой инженерной школы.

# 5. Требования к программе

**5.1. Общие требования**

Необходимо разработать программную библиотеку, содержащую реализацию геометрических объектов, контейнеров и алгоритмов, необходимых для решения задач плоской геометрии.

Библиотека должна содержать реализацию следующих геометрических объектов:

* точка;
* отрезок.

Для хранения этих объектов необходимо реализовать контейнеры:

* цепь;
* контур;
* набор отрезков;
* набор контуров;
* слой отрезков;
* слой контуров.

Отличие слоя от набора заключается в отсутствии топологических пересечений объектов, хранящихся в слое.

В библиотеке должна быть реализована вспомогательная структура на базе красно-чёрного дерева, которая будет использоваться в алгоритмах библиотеки.

Библиотека должна содержать алгоритмы:

* поиск точки пересечения пары отрезков;
* поиск точек пересечений набора отрезков.

Библиотека должна содержать набор функций преобразования:

* из набора отрезков в слой отрезков;
* из набора контуров в слой контуров;
* из слоя отрезков в набор отрезков;
* из слоя контуров в набор контуров;
* из набора контуров в слой отрезков.

Библиотека должна содержать функционал, позволяющий конструировать логические функции для наборов и слоёв отрезков и контуров. В качестве такого функционала выступает паттерн pipeline — представление операций в виде последовательности блоков, выполняющих одну подзадачу, и связанных между собой передаваемыми данными.

Для демонстрации работы библиотеки, используя pipeline, необходимо сконфигурировать две логические операции:

* логическая операция «и» для слоёв отрезков;
* логическая операция «и» для слоёв контуров.

**5.2. Требования к геометрическим объектам**

Точка — структура, описывающая геометрическую точку в двумерном пространстве.

Структура должна содержать:

1. Две координаты – x, y, тип данных — число двойной точности.
2. Функции, возвращающие значения координат для чтения.

Отрезок – геометрический объект, часть прямой, ограниченная двумя точками и имеющее направление.

Структура должна содержать:

1. Две точки: начальную и конечную.
2. Функции, возвращающие точки для чтения.

**5.3. Требования к контейнерам**

Общим требованием к контейнерам, перечисленным в пунктах 5.3.3–5.3.6 ТЗ является возможность задавать набор меток.

Тип метки представляет из себя целое число. Пользователю доступны типы меток от 0 до 255.

Набор меток для контейнера задаётся один раз и в дальнейшем не меняется.

Набор меток для контейнера задаётся функцией установки меток.

Набор значений меток задаётся функцией задания набора меток.

**5.3.1. Требования к цепи**

Цепь — последовательность непересекающихся отрезков, в котором конец каждого отрезка совпадает с началом последующего.

**5.3.2. Требования к контуру**

Контур — цепь, в которой начальная точка первого отрезка совпадает с конечной точкой последнего отрезка.

**5.3.3. Требования к набору отрезков**

Набор отрезков — контейнер для хранения отрезков.

Набор отрезков должен содержать функции:

1. Добавление нового отрезка.
2. Функция, возвращающая отрезок набора по порядковому номеру для чтения.
3. Функция установки типов меток.
4. Функция задания значений меток.

**5.3.4. Требования к набору контуров**

Набор контуров — контейнер, хранящий в себе набор произвольных контуров.

Набор контуров должен содержать функции:

1. Добавление нового контура.
2. Функция, возвращающая контур набора по порядковому номеру для чтения.
3. Функция установки типов меток.
4. Функция задания значений меток.

**5.3.5. Требования к слою отрезков**

Слой отрезков — набор отрезков, которые могут касаться друг друга в одной или обеих крайних точках, но не могут пересекаться.

Слой отрезков может быть получен только в результате применения функции преобразования, представленной в пунктах 5.1 и 5.5.3 ТЗ.

Слой отрезков должен содержать функции:

1. Функция, возвращающая отрезок слоя по порядковому номеру для чтения.
2. Функция установки типов меток.
3. Функция задания значений меток.

**5.3.6. Требования к слою контуров**

Слой контуров — множество непересекающихся контуров, для которых задан лексикографический порядок на основе меток.

Слой контуров может быть получен только в результате применения функции преобразования, представленной в пунктах 5.1 и 5.5.3 ТЗ.

Отрезки, формирующие слой контуров, должны удовлетворять правилам:

1. Каждый отрезок принадлежит какому-то контуру.
2. Отрезки упорядочены.
3. Последний отрезок контура ссылается на первый отрезок следующего контура.

Слой контуров должен содержать функции:

1. Функция, возвращающая контур слоя по порядковому номеру для чтения.
2. Функция установки типов меток.
3. Функция задания значений меток.

### 5.4. Требования к дереву поиска

Дерево поиска должно быть реализовано на базе красно-чёрного дерева и выполнять следующие функции:

1. Добавление элемента с гарантией абсолютного порядка.
2. Поиск минимального/максимального элемента.
3. Удаление элемента с гарантией абсолютного порядка.
4. Поиск следующего элемента для заданного.
5. Поиск предыдущего элемента для заданного.

Все перечисленные операции должны иметь сложность O(log(n)).

### 5.5. Требования к алгоритмам

### 5.5.1. Функция поиска пересечения двух отрезков

Вход: отрезок, отрезок

Выход: отрезок, маркер типа пересечения

Функция осуществляет поиск пересечения двух отрезков. Результатом поиска является отрезок (пункт 5.2 ТЗ) и маркер, указывающий на тип пересечения:

* пересечения нет;
* пересечение в точке;
* пересечение во множестве точек (наложение).

Если пересечения нет, то оба конца результирующего отрезка имеют нулевые координаты.

Если отрезки пересекаются в одной точке, то оба конца результирующего отрезка имеют одни и те же координаты.

Если произошло наложение, то концевые точки результирующего отрезка содержат границы области наложения.

Алгоритм должен иметь сложность O(1).

### 5.5.2. Функция поиска пересечений набора отрезков

Для работы алгоритма поиска пересечений набора отрезков необходимо использовать функцию обратного вызова и функцию разметки рёбер.

**Функция обратного вызова**

Вход: отрезок, отрезок, отрезок

Выход: булево значение

Функция корректирует процесс поиска точек пересечений. Получив два пересекающихся отрезка и их область пересечения, возвращает булево значение, указывающее нужно ли продолжать поиск: true — продолжить, false —завершить.

**Функция разметки рёбер**

Вход: порядковый номер отрезка, ссылка на набор отрезков

Расставляет значения меток для отрезков.

**Функция поиска пересечений набора отрезков**

Вход: набор отрезков, функция разметки рёбер, функция обратного вызова

Осуществляет поиск пересеченийнабора отрезков. Реализация алгоритма должна удовлетворять требованиям:

* 1. Алгоритм поиска должен быть реализован на базе алгоритма заметающей прямой и использовать в работе дерево поиска из пункта 5.4.1 ТЗ.
  2. При идентификации пересечения алгоритм должен передать управление функции обратного вызова.
  3. Алгоритм должен завершить работу, когда функция обратного вызова вернёт false.
  4. Сложность алгоритма O(n\*log(n)).

**5.5.3. Требования к алгоритмам преобразования структур**

Библиотека должна содержать следующие функции преобразования:

1. Из набора отрезков в слой отрезков.
2. Из набора контуров в слой контуров.
3. Из слоя отрезков в набор отрезков.
4. Из слоя контуров в набор контуров.
5. Из набора контуров в слой отрезков.

**Функция преобразования**

Вход: ссылка на структуру

Выход: структура

Результат преобразования — сгенерированный внутри функции объект, который не влияет на состояние входной структуры.

Функции преобразования должны удовлетворять следующим требованиям:

1. Структура, полученная в результате преобразования, сохраняет метки исходной структуры.
2. Алгоритмы 1, 2 должны иметь сложность O(n\*log(n)).
3. Алгоритмы 3, 4 должны иметь сложность O(n).
4. Алгоритм 5 должен иметь сложность O(n).

**5.6. Требования к паттерну pipeline**

Паттерн pipeline — организация операций над слоями и контурами отрезков и слоёв в виде конвейера, состоящего из функциональных блоков (узлов), где каждый блок специализируется на одной подзадаче. Каждый узел обрабатывает данные от предыдущего узла или внешнего кода, если узел стоит в начале цепи, и передаёт результат следующему узлу или внешнему коду, если узел завершает цепь.

Список функциональных блоков:

1. Блок установки меток.
2. Блок поиска точек пересечения пары отрезков.
3. Блок поиска точек пересечения множества отрезков.
4. Блок фильтрации.
5. Блок преобразования (переход от набора отрезков к слою отрезков и т. д.).

Каждый из этих блоков также является конфигурируемым.

**5.7. Требования к демонстрационным функциям**

**5.7.1. Логическая операция «и» для слоёв отрезков**

Вход: слой отрезков, слой отрезков

Выход: слой отрезков

Используя паттерн pipeline, строит слой отрезков из отрезков, принадлежащих одновременно двум входным слоям.

**5.7.2. Логическая операция «и» для слоёв контуров**

Вход: слой контуров, слой контуров

Выход: слой контуров

Используя паттерн pipeline, строит слой контуров, в котором каждый контур ограничивает часть плоскости, находящуюся на пересечении областей, описываемых контурами входных слоёв.

**6. График релизов**

Таблица 1. График релизов

|  |  |  |  |
| --- | --- | --- | --- |
| № релиза | Содержание релизов | Пункты ТЗ | Дата релиза |
| 1 | * Геометрические объекты | 5.2 | 07.11.2022 |
| * Контейнеры (цепь, контур, набор отрезков, набор контуров) | 5.3.1-5.3.4 |
| * Дерево поиска | 5.4.1 |
| * Функция поиска пересечения двух отрезков | 5.5.1 |
| * Функция поиска пересечений набора отрезков | 5.5.2 |
| 2 | * Слой отрезков | 5.3.5 | 28.11.2022 |
| * Слой контуров | 5.3.6 |
| * Функции преобразования базовых сущностей | 5.5.3 |
| * Паттерн pipeline | 5.6 |
| 3 | * Пример логической операции «и» для слоя отрезков | 5.7.1 | 19.12.2022 |
| * Пример логической операции «и» для слоя контуров | 5.7.2 |

**7. Ведомость исполнения**

Этапы НИОКР, ответственные за их выполнение, сроки и виды отчётности приведены в таблице 1.

Таблица 1. Этапы разработки и виды отчётности

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Этап | Содержание | Срок исполнения | | Ответственный | Вид отчётности |
| начало | окончание |
| 1 | Разработка концепции библиотеки |  |  | Исполнитель | Описание концепции |
| 2 | Разработка архитектуры библиотеки |  |  | Исполнитель | Описание архитектуры |
| 3 | Разработка кода библиотеки |  |  | Исполнитель | Исходные коды библиотеки |
| 4 | Разработка демонстрационных примеров |  |  | Исполнитель | Исходные коды примеров |
| 5 | Разработка программной документации |  |  | Исполнитель | Руководство программиста |
| 6 | Формирование верификационного базиса |  |  | Заказчик, Исполнитель | Unit-тесты |
| 7 | Разработка научно-технического отчёта |  | 26 декабря | Исполнитель | Научно-технический отчёт |
| 8 | Разработка методики испытаний |  | 26 декабря | Исполнитель | Программная методика испытаний |
| 9 | Испытания системы |  | 15 января | Заказчик, Исполнитель | Протокол |
| 10 | Приёмка НИОКР |  | 15 января | Заказчик | Акт приёмки |