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 — 10.01.2023.

# 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

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
* **пайплайн (дописать)**

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

Архитектура библиотеки должна быть построена таким образом, чтобы с помощью передачи функций обратного вызова в алгоритм поиска точек пересечений отрезков можно было сконфигурировать новые алгоритмы для выполнения основных операций, таких как: логические операции над отрезками, логические операции над контурами.

Для демонстрации работы библиотеки необходимо сконфигурировать два алгоритма:

* 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 **(написать про тип данных)**
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 (добавить требования к интерфейсам)**

Общим требованием к контейнерам, перечисленным в пунктах 5.3.1–5.3.6 ТЗ является возможность задавать набор меток.

Тип метки представляет из себя целое число. Пользователю доступны типы меток от 1 до 255. Метка 0 зарезервирована.

Набор меток для контейнера задаётся функцией установки меток. **(необходимо уточнение)**

Набор значений меток задаётся функцией задания набора меток.

Цена установки метки O(1)

**(уточнить)**

**5.3.1. Требования к цепи**

Цепь — последовательность 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 установки типов меток
3. 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 установки типов меток
3. 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. **Может быть получен только преобразованием. Только метки.**

Слой отрезков должен содержать функции:

1. добавление нового отрезка **(описать по каким правилам происходит добавление)**

**5.3.6. Требования к слою контуров**

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

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. добавление нового контура

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: указатель на точку или null **(пока не утверждено). Отрезок?**

Функция осуществляет поиск точки пересечения двух отрезков. Результатом является указатель на точку, содержащую координаты пересечения или null, если пересечения нет.

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, точка

**Вход: отрезок, отрезок, отрезок (как множество точек пересечения)**

Выход: булево значение

Функция корректирует процесс поиска точек пересечений. Получив два пересекающихся отрезка и их общую точку, 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 точек пересечения набора 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

Функции преобразования должны удовлетворять следующим требованиям:

* Алгоритмы 1, 2 должны иметь сложность O(n\*log(n))
* Алгоритмы 3, 4 должны иметь сложность O(n)
* Алгоритм 5 должен иметь сложность O(n)

**6. График релизов**

Таблица 1. График релизов

|  |  |  |  |
| --- | --- | --- | --- |
| № релиза | Содержание релизов | Пункты ТЗ | Дата релиза |
| 1 | * Геометрические объекты | 5.2 | 04.11.2022 |
| * 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 |
| * Функции построения логических операций со слоями отрезков и слоями контуров | 5.5.4 |
| Пример логической операции «и» для слоя отрезков | 7.5 |
| 3 | * Пайплайн * Два демо-примера | 8.1 | 19.12.2022 |
| * Пример логической операции «и» для слоя контуров | 8.2 |

**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 | **Согласование форматов входных и выходных данных** |  |  | **Исполнитель, Заказчик** | **Пояснительная записка** |
| 3 | Разработка архитектуры библиотеки |  |  | Исполнитель | **описание архитектуры** |
| 4 | Разработка кода библиотеки |  |  | Исполнитель | Исходные коды библиотеки |
| 5 | Разработка демонстрационных примеров |  |  | Исполнитель | Исходные коды примеров |
| 6 | Разработка программной документации |  |  | Исполнитель | Руководство программиста |
| 7 | Формирование верификационного базиса |  |  | Заказчик, Исполнитель | Unit-тесты |
| 8 | Разработка научно-технического отчёта |  | 26 декабря | Исполнитель | Научно-технический отчёт |
| 9 | Разработка методики испытаний |  | 26 декабря | Исполнитель | Программная методика испытаний |
| 10 | Испытания системы |  | 15 января | Заказчик, Исполнитель | Протокол |
| 11 | Приёмка НИОКР |  | 15 января | Заказчик | Акт приёмки |