

СОГЛАСОВАНО

УТВЕРЖДАЮ

Сторона ЗАКАЗЧИКА

Сторона ИСПОЛНИТЕЛЯ

Инженер-программист НИО 9740
филиала РФЯЦ-ВНИИЭФ
«НИИИС им. Ю.Е. Седякина»

Профессор кафедры
ИАНИ ННГУ, д.т.н.

Ю.А. Живчикова
« ____ » _____ 2021 г.

Н.В. Старостин
« ____ » _____ 2021 г.

Отчет

«API и формат входных и выходных данных»

Этап 1. Разработка API. Разработка формата входных и выходных данных.

**НИР «Разработка и реализация программного обеспечения для
выполнения логических операций над множествами ортогональных
многоугольников»**

(Шифр ПО «SoR»)

Ответственный исполнитель

В.А. Куликов

« ____ » _____ 2021 г.

2021

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
2. НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	3
3. ВХОДНЫЕ ДАННЫЕ.....	3
4. ВЫХОДНЫЕ ДАННЫЕ	4
4.1. Решение	4
4.2. Протокол работы системы.	4
5. API.....	5

1. ВВЕДЕНИЕ

1.1. Наименование НИР

Разработка и реализация программного обеспечения для выполнения логических операций над множествами ортогональных многоугольников (ПО «SoR»).

1.2. Краткая характеристика области применения

Объектом автоматизации является работа над множествами многоугольников. ПО «SoR» должно обеспечивать решение задач связанных с обработкой и осуществлением булевых операций над множествами многоугольников.

2. НАЗНАЧЕНИЕ РАЗРАБОТКИ

ПО «SoR» предназначено для осуществления булевых операций над множествами ортогональных многоугольников и визуализации полученных результатов.

3. ВХОДНЫЕ ДАННЫЕ

Входные данные содержатся в файле формата .txt. В файле указываются координаты точек многоугольника в порядке обхода по часовой стрелке. Координаты каждого нового многоугольника записываются с новой строки. Координаты многоугольника отделяются символом « , ». Первая строка файла входных данных содержит количество многоугольников.

Пример файла входных данных приведен на рис. 1.

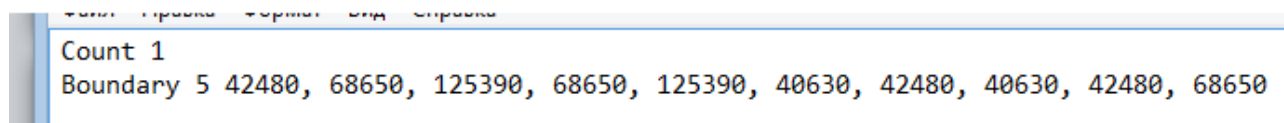


Рисунок 1 – Файл входных данных

4. ВЫХОДНЫЕ ДАННЫЕ

К выходным данным ПО «SoR» относятся:

- решение;
- протокол работы системы.

4.1. Решение

Решением является множество многоугольников, полученных в результате выполнения логических операций.

ПО «SoR» осуществляет выгрузку полученного решения в файл формата .txt. Формат файла решения совпадает с форматом файла входных данных и описан в разделе 3.

4.2. Протокол работы системы.

Вместе с решением в качестве входных данных система генерирует лог-файл (документ формата .txt), содержащий протокол работы системы.

Структура сообщений в лог-файле:

`logger->WriteLog(LOGGER::LogLevel::Data, Time, MessageType, Message)`

Аргумент `MessageType` – тип сообщения. Аргумент может принимать следующие значения:

– `Error` – сообщение об ошибке.

Аргумент `Message` – непосредственно сообщение.

Пример сообщений:

- `Data\Time\Read first file – successfully (unsuccessfully);`
- `Data\Time\Read second file - successfully (unsuccessfully);`
- `Data\Time\Merge – successfully (unsuccessfully);`
- `Data\Time\Read file with answer - successfully (unsuccessfully);`
- `Data\Time\Intersect – successfully (unsuccessfully);`
- `Data\Time\Subtrac – successfully (unsuccessfully);`
- `Data\Time\Accepted – successfully (unsuccessfully);`
- `Data\Time\Write answer- successfully (unsuccessfully);`

- Data\Time\Error Not enough input data;
- Data\Time\Error Incorrect path to first file;
- Data\Time\Error Incorrect path to second file;
- Data\Time\Error Incorrect operation;
- Data\Time\Error Incorrect path to file with answer;
- Data\Time\Error Wrong Answer.

Так же логируются начало вызова программы из консоли и завершение работы программы.

Пример лог-файла приведен на рис. 2.

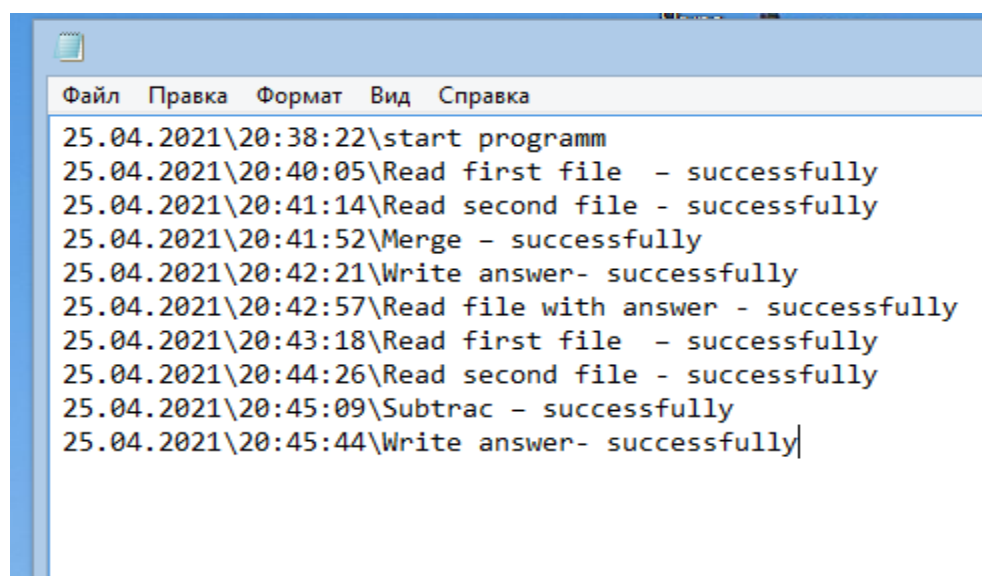


Рисунок 2 – Лог-файл

5. API

API ПО «SoR» предоставляет следующие функции:

1) Парсинг - read_input

Сигнатура функции:

```
void read_input (string &path, Polygons &res)
```

Данная функция позволяет преобразовать данные из файла в класс.

Входные параметры:

- string path - путь к файлу с входными данными.

Возвращает: класс, представляющий входные данные.

2) Функция объединения множеств - merge

Сигнатура функции:

```
void merge(Polygons &data1, Polygons &data2, Polygons &res)
```

Осуществляет объединение множеств.

В случае возникновения ошибки, записывает в файл информацию, что программа делала в этот момент.

Входные параметры:

- Polygons data1 - первое множество;
- Polygons data2 - второе множество.

Возвращает: результирующий класс в переменную res.

3) Функция пересечения множеств - intersect

Сигнатура функции:

```
void intersect(Polygons &data1, Polygons &data2, Polygons &res)
```

Осуществляет пересечение множеств.

В случае возникновения ошибки, записывает в файл информацию, что программа делала в этот момент.

Входные параметры:

- Polygons data1 - первое множество;
- Polygons data2 - второе множество.

Возвращает: результирующий класс в переменную res.

4) Функция вычитания множеств - subtract

Сигнатура функции:

```
void subtract(Polygons &data1, Polygons &data2, Polygons &res)
```

Осуществляет вычитание множеств.

В случае возникновения ошибки, записывает в файл информацию, что программа делала в этот момент.

Входные параметры:

- Polygons data1 - первое множество;

- Polygons data2 - второе множество.

Возвращает: результирующий класс в переменную res.

5) Метод записи выходных данных - write_output

Сигнатура функции:

```
void write_output(string &path, Polygons &data)
```

Данная функция позволяет преобразовать результирующий класс в выходные данные и дальнейшую их запись в файл.

Входные параметры:

- Polygons data – результирующий класс.

Возвращает: количество многоугольников, количество точек в многоугольниках, координаты многоугольников в порядке их обхода по часовой стрелке.