

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский  
подпись  
«\_\_\_\_\_» \_\_\_\_\_ 2018 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.02 — Информационные системы и технологии  
Разработка модуля анализа локальных неоднородностей  
сельскохозяйственных земель

Руководитель

\_\_\_\_\_  
подпись, дата

доцент, канд. техн. наук Р. В. Брежнев

Выпускник

\_\_\_\_\_  
подпись, дата

М. О. Семенов

Красноярск 2018

Продолжение титульного листа бакалаврской работы по теме «Разработка модуля анализа локальных неоднородностей сельскохозяйственных земель».

Нормоконтроль

\_\_\_\_\_  
подпись, дата

Р. В. Брежнев

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий  
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ Г. М. Цибульский  
подпись

«\_\_\_\_\_» \_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**  
**В форме бакалаврской работы**

Студенту Семенову Матвею Олеговичу

Группа КИ14-11Б, направление 09.03.02 «Информационные системы и технологии», профиль 09.03.02.04 «Информационные системы и технологии в медиаиндустрии».

Тема выпускной квалификационной работы «Разработка модуля устранения шумов в задачах постсегментации спутниковых изображений».

Утверждена приказом по университету № 4533/с от 29.03.2018.

Руководитель ВКР Р. В. Брежнев доцент кафедры систем искусственного интеллекта ИКИТ СФУ.

Исходные данные для ВКР: задание на бакалаврскую работу, полученное в рамках научно-учебной лаборатории «Информационной поддержки космического мониторинга» Института космических и информационных технологий.

Перечень разделов ВКР: введение; обзор предметной области; выводы по главе 1; проектирование и реализация модуля; выводы по главе 2; заключение; список использованных источников; приложение А — В (листинг модуля симметрической разности, листинг модуля преобразования векторных полигонов к выпуклому виду без пересечений, плакаты презентации).

Перечень графического материала: презентация «Разработка модуля анализа локальных неоднородностей сельскохозяйственных земель».

Руководитель ВКР

\_\_\_\_\_

подпись

Р. В. Брежнев

Задание принял к исполнению

\_\_\_\_\_

подпись

М. О. Семенов

«\_\_\_» \_\_\_\_\_ 2018 г.

## График

выполнения выпускной квалификационной работы студентом направления 09.03.02 «Информационные системы и технологии», профиля 09.03.02.04 «Информационные системы в медиаиндустрии» приведен в таблице 1.

Таблица 1 — График выполнения этапов ВКР

Наименование этапа	Срок выполнения этапа	Результат выполнения этапа	Примечание руководителя (отметка о выполнении этапа)
Ознакомление с целью и задачами работы	07.03-13.03	Краткое эссе по теме ВКР	Выполнено
Сбор литературных источников	14.03-20.03	Список источников литературы	Выполнено
Анализ собранных литературных источников	21.03-27.03	Реферат о проблемно-предметной области	Выполнено
Уточнение и обоснование актуальности цели и задач ВКР	28.03-03.04	Окончательная формулировка цели и задач ВКР	Выполнено
Решение первой задачи ВКР	04.04-10.04	Доклад и презентация по первой задаче ВКР	Выполнено
Решение второй задачи ВКР	11.04-17.04	Доклад и презентация по второй задаче ВКР	Выполнено
Решение третьей задачи ВКР	18.04-24.04	Доклад и презентация по третьей задаче ВКР	Выполнено

## Окончание таблицы 1

Наименование этапа	Срок выполнения этапа	Результат выполнения этапа	Примечание руководителя (отметка о выполнении этапа)
Подготовка доклада и презентации по теме ВКР	25.04.08.05	Доклад с презентацией на тему ВКР	Выполнено
Компоновка отчета по результатам решения задач ВКР	09.05-31.05	Отчет по результатам решения задач ВКР	Выполнено
Первичный нормоконтроль (Н/К)	06.06	Пояснительная записка, презентация к ВКР	Выполнено
Предварительная защита результатов ВКР	07.06	Доклад и презентация о проделанной работе	Выполнено
Вторичный нормоконтроль (Н/К)	15.06	Пояснительная записка, презентация к ВКР	Выполнено
Итоговый нормоконтроль (Н/К)	17.06	Пояснительная записка, презентация к ВКР	Выполнено
Защита ВКР	21.06	Доклад и презентация по результатам бакалаврской работы	Выполнено

Студент

Руководитель ВКР  
Доцент кафедры СИИ

\_\_\_\_\_

\_\_\_\_\_

М. О. Семенов

Р. В. Брежнев

## СОДЕРЖАНИЕ

Введение.....	5
Глава 1 Обзор предметной области.....	7
1.1 Точное земледелие .....	7
1.2 Принцип работы системы точного земледелия.....	7
1.3 Методы обработки неоднородных участков растительности в задачах точного земледелия .....	8
1.4 Обзор операций над векторными объектами в QGIS .....	9
1.5 Векторная модель данных .....	11
1.6 Выводы по главе 1 .....	14
Глава 2 Проектирование и реализация модуля.....	15
2.1 Проектирование модуля .....	15
2.1.1 Описание входных данных .....	15
2.1.2 Методология Use Case .....	17
2.1.3 Определение актеров модуля.....	17
2.1.4 Варианты использования.....	18
2.1.5 Описание прецедентов.....	19
2.1.6 Диаграмма деятельности .....	21
2.2 Реализация модуля .....	24
2.2.1 Основной язык разработки.....	24
2.2.2 Программа для работы с геоданными.....	25
2.2.3 Программа для разработки интерфейса.....	27
2.2.4 Интерфейс модуля.....	28
2.2.5 Технические требования к программному окружению .....	29
2.2.6 Выполняемые функции .....	29
2.2.7 Описание кода модуля нахождения симметрической разности векторных полигонов. ....	30
2.2.8 Блок-схема алгоритма нахождения симметрической разности векторных объектов.....	30

2.2.9 Результат работы алгоритмов .....	32
2.2.10 Вывод по главе 2.....	34
Заключение .....	35
Список сокращений .....	36
Список использованных источников .....	37
Приложение А Листинг модуля нахождения симметрической разности полигонов .....	39
Приложение Б Листинг модуля преобразования выбранных полигонов к выпуклой форме без пересечений .....	43
Приложение В Плакаты презентации .....	47
Приложение Г Отчет в системе «Антиплагиат».....	54

## ВВЕДЕНИЕ

**Актуальность.** Разработка модуля анализа локальных неоднородностей сельскохозяйственных земель является частью более глобального алгоритма по обработке снимков сельскохозяйственных контуров для использования в точном земледелии.

В основе точного земледелия лежит представление о неоднородности в рамках одного поля. Для того, чтобы обнаружить эти неоднородности используются последние технологии, в числе которых системы глобального позиционирования (ГЛОНАСС, GPS), снимки со спутников и аэрофотоснимки, специальные датчики, и программы в основе которых лежат геоинформационные системы (ГИС). Информация, которая приходит от вышеизложенных источников используется для того, чтобы проводить ряд сельскохозяйственных мероприятий для улучшения урожайности, путем точного воздействия на участок земли с культурами, которым требуется обработка. Таким образом точное земледелие позволяет существенно уменьшить затраты на обработку сельскохозяйственных культур и как следствие повысить доходы от производства.

Алгоритмы нахождения симметрической разности и преобразование заданных полигонов к выпуклой форме без пересечений является важным этапом в процессе обработки векторизованного снимка неоднородностей. Дело в том, что неоднородности практически никогда не бывают выпуклой формы и часто пересекаются между собой. Получается точная карта неоднородностей, однако такие контуры крайне сложно обработать. Идеальным контуром для обработки трактором является правильная окружность. Трактор начинает обработку по спирали от центра к внешнему радиусу круга. К сожалению, большая часть полей и неоднородностей имеет форму отличную от круга. Однако можно это компенсировать найдя симметрическую разность контуров и приведя их к выпуклой форме. Эта форма контуров будет максимально приближена к форме окружности, что упростит обработку неоднородностей,

выявленных на полях сельскохозяйственных культур, исключит наложение контуров друг на друга, а как следствие и «двойную обработку поля» и снизит стоимость обработки с/х полей.

**Цель выпускной квалификационной работы.** Разработать модуль для анализа локальных неоднородностей сельскохозяйственных земель.

**Решаемые задачи:**

– Провести обзор методов обработки неоднородных участков растительности в задачах точного земледелия.

– Разработка блок-схем алгоритмов для модуля анализа локальных неоднородностей сельскохозяйственных земель.

– Разработка программного модуля, реализующего алгоритма нахождения симметрической разности векторных полигонов и алгоритма преобразования полигонов к выпуклой форме без пересечений.

– Экспериментальная апробация разработанного модуля.

**Практическая значимость.** Представленный в проекте алгоритм реализован программно и может быть использован как модуль в географической информационной системе (ГИС) QGIS и будет работать в автоматизированном режиме. Автоматический режим в разработке, в последующем алгоритм будет встроен в систему агромониторинга на основе действующего программно-аппаратного комплекса центра ДЗЗ Сибирского федерального университета.

**Структура и объем ВКР.** Выпускная квалификационная работа состоит из введения, двух глав, заключения, списка сокращений, списка использованных источников и приложения. Основной текст включает 55 страниц. Изложение иллюстрируется 27 рисунками. Список использованных источников содержит 14 наименований. В приложении приведены программный код и плакаты презентации.

## **Глава 1 Обзор предметной области**

### **1.1 Точное земледелие**

Система точного земледелия содержит в себе множество различных технологий, которые используются как совместно, так и отдельно. Решение о том какие технологии и где применять стоит за руководителями агропредприятий. Некоторые участки требуют одних технологий, а некоторые других, выбираются именно те, что будут максимально эффективны.

Базисом всей системы земледелия являются геопространственные данные. Эти карты содержат все характеристики полей. Естественно для всех полей есть и местные карты с элементами реестра о земельных участках, однако их полезность стоит под сомнением [1].

### **1.2 Принцип работы системы точного земледелия**

Для того, чтобы максимально точно и минимальными временными затратами использовать IT-технологии в направлении точного земледелия необходимо знать не только территориальные границы полей, но и множество других параметров почвы, растительности. В состав этих параметров входят состав почвы на молекулярном уровне, наличие вод протекающий под землей, величина радиационного излучения, горизонтальный уклон, розу ветров, соседние объекты и др. Получая и анализируя все эти факторы, производственный процесс будет гораздо более гибким.

Для того чтобы составить такую подробную точную карту полей требуется не мало сил. Используются лабораторные анализы и системы спутникового наблюдения, а также подробный анализ каждого поля с научной точки зрения. Для того чтобы зафиксировать такой большой объем данных используются информационные системы, которые работают совместно с оборудованием по добычи информации.

Анализируя созданные точные карты полей создаются рекомендации для каждого поля отдельно, для каждой неоднородности. Данные рекомендации содержат количество удобрений, количество воды, которые необходимо вылить на сельскохозяйственную культуру, чтобы рост культуры стал максимально быстрым. Количество семян при засевании тоже необходимо регулировать в соответствии с типом и плодородностью почвы, ведь где-то при большой плотности растения откажутся расти и большинство семян пропадет. Все эти данные передаются на специальные бортовые компьютеры, которые находятся на сельхозтехнике. Система контролирует подачу удобрений, воды или семян, а оператор лишь контролирует корректную работу системы. Данный метод позволяет исключить избыточное или недостаточное количество удобрений, что положительно сказывается на финансовом благополучии агробизнеса [2].

### **1.3 Методы обработки неоднородных участков растительности в задачах точного земледелия**

Перед тем как начать обработку неоднородных участков необходимо их обнаружить, а для этого важно знать какие бывают неоднородности.

Неоднородности могут находиться внутри других неоднородностей, это тоже важно понимать, к тому же тип неоднородности постоянно меняется и зависит от нескольких факторов, таких как дождь, заморозки, туман, долгое отсутствие дождя, стихийные бедствия, налет насекомых и другие. Также неоднородности зависят и от характеристик почвы (ее химический состав, который во много определяет плодородна она или нет), а также от того каким способ данный участок земли обрабатывали. По ряду точечных характеристик возможно определить от чего пострадала культура и какие меры предпринять, чтобы избежать этого в последующих засевах или устранить в ближайшее время.

Есть два метода, которые позволяют учесть неоднородности полей, оба метода используют карты агро-физико-химических показателей почвы.

Первый способ основывается на проведение анализа полей, а конкретно показатели почвы, история прошлых лет засева и прочее. Способ подразумевает прогноз предстоящих затрат на выращивание культуры.

Второй способ подразумевает фазовый контроль роста культуры. То есть данные обновляются каждый раз, когда культура перешла в новую фазу роста [3]. Фазовый контроль роста делиться на подвиды:

- взятия образцов и измерение их характеристик;
- определение параметров, таких как температура, влажность, роза ветров, диаметр стеблей дистанционно;
- сенсорное наземное детектирование;
- дистанционное зондирование Земли.

#### 1.4 Обзор операций над векторными объектами в QGIS

Пересечение (рисунок 1.1) — совмещает слои таким образом, что в выходном слое содержатся только участки, в которых оба слоя пересекаются.



Рисунок 1.1 — Пример операции пересечения

Объединение (рисунок 1.2) – совмещает слои таким образом, что в выходном слое содержатся как участки пересечения, так и участки, принадлежащие только одному из слоев.

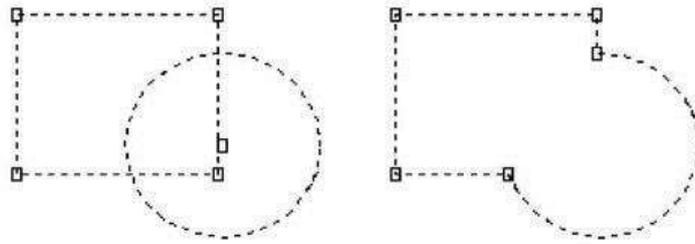


Рисунок 1.2 — Пример операции объединение

Симметрическая разность (рисунок 1.3) — совмещает слои таким образом, что в выходном слое содержатся только те участки, в которых исходные слои не пересекаются.

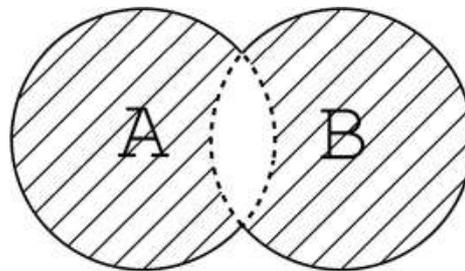


Рисунок 1.3 — Пример операции симметрической разности

Отсечение (рисунок 1.4) — совмещает слои таким образом, что в выходном слое содержатся только те участки, которые пересекаются со слоем отсечения.

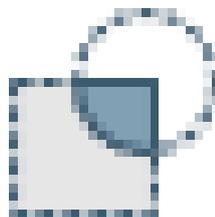


Рисунок 1.4 — Пример операции отсечение

Разность (рисунок 1.5) — совмещает слои таким образом, что в выходном слое содержатся только те участки, которые не пересекаются со слоем отсечения.

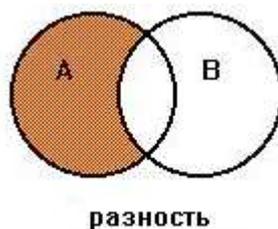


Рисунок 1.5 — Пример операции разность

Разбить составные объекты (рисунок 1.6) — преобразует составные объекты (мульти-полигоны или мульти-полилинии) в несколько простых объектов (полигонов или полилиний).



Рисунок 1.6 — Пример операции «разбить на составные объекты»

Объединение по признаку (рисунок 1.7) — объединяет объекты на основе значения указанного поля. Все объекты с одинаковым значением поля будут объединены в один объект [8].

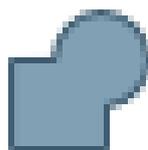


Рисунок 1.7 – Объединение по признаку

## 1.5 Векторная модель данных

Векторная модель данных или цифровое представление точечных, линейных и полигональных пространственных объектов в виде набора координатных пар, с описанием только геометрии объектов, что соответствует нетопологической модели. Векторно-нетопологическое представление данных

в ГИС называют модель «спагетти». Векторным моделям соответствует векторный формат пространственных данных.

Векторные модели строятся с использованием векторов, в которых каждая точка на карте определяется через ее удаленность от опорной точки и величину угла между направлением на точку из опорной точки и направлением на Север (по часовой стрелке). Векторные изображения занимают значительно меньше памяти ЭВМ при хранении, чем растровые, требуют меньше затрат времени на обработку.

Базовым примитивом векторных моделей ГИС является точка. Через понятие «точка» определяются все остальные объекты векторной модели.

Безразмерные типы объектов:

- Точка определяет геометрическое местоположение объекта.
- Узел — топологический переход или конечная точка, также может определять местоположение объекта.

Одномерные типы объектов:

- Линия — одномерный объект.
- Линейный сегмент — прямая линия между двумя точками.
- Дуга — геометрическое место точек, которые формируют кривую, определенную математической функцией.
- Связь — соединение между двумя узлами.
- Направленная связь – связь с одним определенным направлением.
- Кольцо — последовательность непересекающихся цепочек, строк, связей или замкнутых дуг.

Двумерные типы объектов:

- Область — ограниченный непрерывный объект, который может включать или не включать в себя собственную границу.
- Внутренняя область — область, которая не включает собственную границу.

– Полигон (контур) — двумерный (площадной) объект, внутренняя область которого образована замкнутой последовательностью сегментов в модели «спагетти».

Векторное изображение можно получить различными способами. Наиболее часто используют векторизацию сканированного (растрового) изображения. Векторизация заключается в распознавании на растровом изображении объектов, выделение их, представление каждого объекта в векторном формате. Для автоматической векторизации необходимо иметь изображения высокого качества.

К особенностям векторных моделей можно отнести следующие:

– В векторной модели легко осуществляются некоторые операции с объектами, например, разбивка объекта (речной сети) на участки, замена условных обозначений.

– Легко проводятся изменение масштаба, повороты, растягивание и другие операции.

– Векторные модели имеют преимущество перед растровыми моделями в точности представления точечных объектов.

Векторная модель оперирует такими основными объектами, как точка, линия, полилиния, полигон, мультиполигон.

Множество точечных объектов, образующее набор однородных данных, можно представить в виде векторного слоя с таблицей упорядоченных или неупорядоченных последовательностей записей, каждая из которых должна содержать по меньшей мере три атрибута: уникальный идентификационный номер объекта, значение координаты X и значение координаты Y в некоторой системе координат [7]. Векторная модель точечного объекта выглядит следующим образом:

$$m = \{x_i, y_i\}, \quad (1.1)$$

где  $i = 0 \dots n$ ,

$n$  – количество точек контура.

## **1.6 Выводы по главе 1**

В данной главе проведено ознакомление с понятием точного земледелия и рассмотрен принцип работы системы точного земледелия. Также произведено ознакомление с методами обработки неоднородных участков растительности в задачах точного земледелия. Произведен обзор операций над векторными объектами в ГИС QGIS, а также рассмотрена векторная модель данных.

На основании изученного материала, который представлен в первой главе можно сделать вывод, что создание модуля для решения задачи реализации алгоритма объединения пересекающихся векторных объектов геометрической формы целесообразно, чтобы приводить векторизованные данные со спутника к удобному виду для дальнейшей обработки в системе агромониторинга, которая создана на основе действующего программно-аппаратного комплекса центра ДЗЗ Сибирского федерального университета.

## **Глава 2 Проектирование и реализация модуля**

### **2.1 Проектирование модуля**

При выполнении выпускной квалификационной работы описаны прецеденты и созданы диаграммы вариантов использования и диаграмма деятельности, которые дают представление о будущем модуле и требованиях к его функционалу и интерфейсу.

#### **2.1.1 Описание входных данных**

В процессе обработки сельскохозяйственных неоднородностей на этапе сегментирования внутри исследуемого объекта возникают пространственные неоднородности внутри его контура в результате природно-климатических и техногенных воздействий. Это определяем неоднородность структуры объекта, характер неоднородности изменяется во времени.

Для того чтобы принять решение о точечном применении техники, необходимо определить, в каком состоянии находиться та или иная неоднородная область.

Также на этом этапе идет преобразование сегментов в векторную форму, в результате которого получаются области, принадлежащие разным неоднородностям, описанным ниже. На следующем этапе идет приведение контуров объекта к выпуклой форме, на котором получаются пересечения неоднородностей.

Далее идет выбор объектов, не соответствующих нормативному состоянию, которые определяются на этапе сегментации. На данном этапе есть возможность сортировки объектов по толщине и площади.

Объект характеризуется следующими группами характеристик, существенных для организации системы мониторинга:

– Пространственная локализация (точные географические координаты объекта с учетом топологических особенностей).

– Наименование и сорт сельскохозяйственной культуры текущего полевого сезона.

– График сезонного хода вегетации, проявляющийся в смене фенологических фаз культуры.

– Возможные аномалии (в привязке к фенофазам), такие как невсхожесть, переувлажнение почвы, засуха, засоренность и т. д.

– Техногенное воздействие, выраженное в планируемых и осуществляемых агротехнических мероприятиях (АТМ) (таблица 1.1).

– Характеристики сельскохозяйственной культуры (такие как интегральное состояние, фенологические фазы)

После выбора нормативного состояния анализируем взаиморасположение объектов, используя симметрическую разность.

Входные данные были получены со спутника World-View-2 с пространственным разрешением 0,5 метров, дата получения: 25 июня 2016 года. Данные представлены в формате shape-файла и изображены на рисунке 2.1.

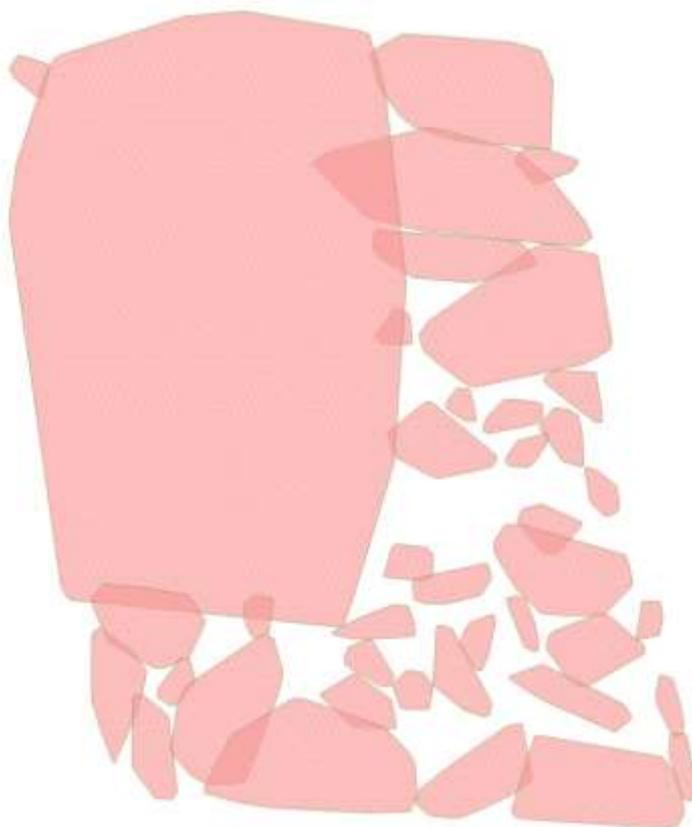


Рисунок 2.1 — Входные данные (спутник World-View-2, дата получения 25.06.2016)

### **2.1.2 Методология Use Case**

Use Case — сценарная техника описания взаимодействия. Эта техника помогает описать требования от пользователя, требования к взаимодействию систем. Также Use Case может описать взаимодействие людей на предприятии в рамках бизнес процессов, даже взаимодействие реальных компаний.

С помощью диаграмм Use Case можно описывать как взаимодействуют два или более участников, которые имеют какую-то задачу:

- доставка товара (Покупатель – Курьер);
- покупка товара (Покупатель – Продавец);
- просмотр ролика в интернете (Пользователь – Браузер – Web-сервер).

Данную технику сценария активно используют при разработке программного обеспечения, чтобы описать взаимодействие пользователя и системы.

Для того, чтобы разработать много различных требований используют набор Use Case диаграмм, каждая из которых отображает требования с каждой стороны разрабатываемой информационной системы. Именно это обеспечивает полноту требований конечного пользователя к системе.

Чтобы описать сущность этой диаграммы надо понимать, что система состоит из множества сущностей, иначе их называют актеры. Каждый актер взаимодействует с системой при помощи так называемых вариантов использования. Актер называется любой, кто взаимодействует с системой. Это может быть, как пользователь, так и другие системы.

### **2.1.3 Определение актеров модуля**

Актер — любая внешняя, относительно разрабатываемой системы, сущность, которая взаимодействует с системой и использует ее функциональные возможности для решения частных задач или достижения конечной цели. Для

представления актера на диаграмме использует фигурка «человека» с подписью имени актера.

Примером актеров могут служить любые сущности, которые взаимодействуют с системой и пользуются ее функционалом. Например, продавец, покупатель, заказчик, курьер, программист, скульптор, работник цеха.

В проектируемом модуле актерами могут быть 2 субъекта: «Пользователь» и «Система агромониторинга».

#### **2.1.4 Варианты использования**

Диаграмма вариантов использования описывает порядок действий, которые должны происходить при взаимодействии того или иного актера с проектируемым модулем. Имеется возможность дополнить диаграмму вариантов подписями к действиям, которые более подробно раскроют суть того или иного действия. Эти подписи называются примечанием. Конкретный вариант использования иллюстрируется в диаграмме эллипсом, название содержит глагол, описывающий действие.

На рисунке 2.2 изображена диаграмма вариантов использования взаимодействия «Оператор» и «Сервис агромониторинга».

На диаграмме можно увидеть, что вариантами использования являются прецеденты «Выбрать объект», «Применить алгоритм симметрической разности», «Применить алгоритм преобразования заданных полигонов к выпуклой форме без пересечения».

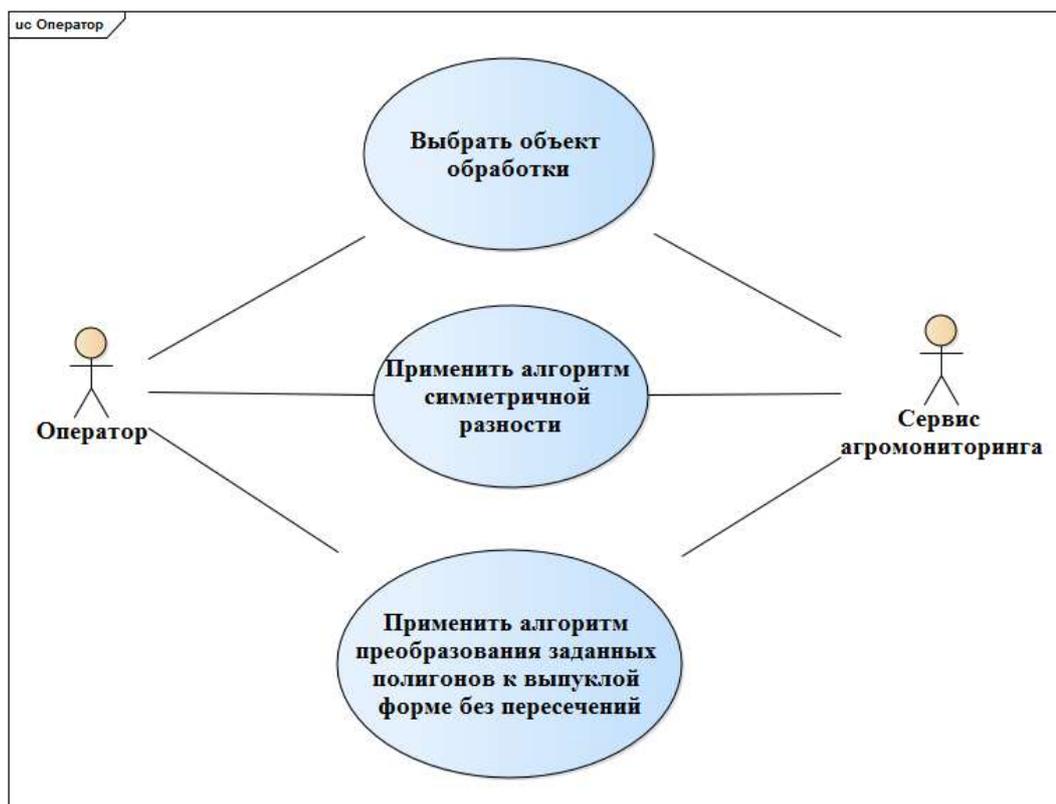


Рисунок 2.2 — Варианты использования актеров «Пользователь», «Система агромониторинга»

## 2.1.5 Описание прецедентов

Прецедент D1: выбрать объект обработки

Перед тем как начинается этот прецедент, «Пользователь» должен добавить share-файл в проект.

Краткое описание прецедента:

- Пользователь загружает share-файл в проект
- Пользователь выделяет загруженный объект в ГИС QGIS

Поток событий начинается с загрузки share-файла в проект

В базовом потоке происходит запись выбранного слоя share-файла в переменную.

- Пользователь загружает share-файл в проект.
- Пользователь выделяет слой share-файл.
- Система записывает выбранный слой share-файла в переменную.

Прецедент содержит следующие постусловия. При успешном окончании прецедента система запишет выбранный слой shape-файла в переменную. При не успешном окончании прецедента система выдаст ошибку в логе скрипта.

Прецедент D2: применить алгоритм симметрической разности полигонов выбранного объекта.

Перед тем как начинается этот прецедент, «Пользователь» должен выделить необходимый объект.

Краткое описание прецедента:

- пользователь нажимает кнопку выполнить скрипт.
- действующие лица этого прецедента – «пользователь», «система агромониторинга».

Поток событий начинается с нажатия пользователем на кнопку «выполнить» в интерфейсе скрипта.

В базовом потоке происходит преобразование выбранного объекта в симметрическую разность данного объекта.

- Пользователь выбирает слой проекта.
- Пользователь нажимает кнопку «Выполнить».
- Система находит индекс данного слоя.
- Система включает редактирование слоя.
- Создает шаблон запроса, в котором считается объект пересечения и затем из исходного объекта вычитается объект пересечения.
- Выполняется запрос.
- Создается виртуальный слой на основе запроса и добавляется в проект.

Прецедент содержит следующие постусловия. При успешном окончании прецедента система автоматически откроет результирующий файл. При не успешном окончании прецедента система выдаст ошибку в логе скрипта.

Прецедент D3: применить алгоритм преобразования выбранных полигонов к выпуклой форме без пересечений.

Перед тем как начинается этот прецедент, «Пользователь» должен выделить необходимый объект.

Краткое описание прецедента:

- пользователь нажимает кнопку выполнить скрипт.
- действующие лица этого прецедента – «пользователь», «система агромониторинга».

Поток событий начинается с нажатия пользователем на кнопку «выполнить» в интерфейсе скрипта.

В базовом потоке происходит преобразование выбранных полигонов к выпуклой форме.

- Пользователь выбирает слой проекта.
- Пользователь нажимает кнопку «Выполнить».
- Система считывает геометрию выбранного слоя
- Система создает выходной файл
- Система приводит выбранный слой к выпуклому виду
- Система записывает результат в выходной файл

Прецедент содержит следующие постусловия. При успешном окончании прецедента система автоматически откроет результирующий файл. При не успешном окончании прецедента система выдаст ошибку в логе скрипта.

### **2.1.6 Диаграмма деятельности**

Диаграмма деятельности — UML-диаграмма, которая позволяет проследить цепочку действий, происходящих во время взаимодействия пользователя с информационной системой. Все действия в диаграмме соединяют потоки,двигающиеся от выходов одного узла к входам другого.

Использование диаграмм деятельности полезно при моделировании различных процессов.

Начало процесса иллюстрируется черным кругом, конец процесса — черным кругом с обводкой. Действия отождествляют прямоугольники с

закругленными углами, а решения – ромбы. Широкая полоса обозначает начало и окончание ветвления действий. Альтернативой диаграммы деятельности является блок-схема. Потоками управления или потоками данных являются стрелки на диаграмме деятельности.

Диаграмма деятельности модуля получения симметрической разности изображена на рисунке 2.3. Оператор запускает программный модуль в ответ система запрашивает входные значения. Оператор выбирает файл для обработки и вводит в систему путь к файлу, затем вводит путь к файлу выходных данных. После чего система применяет алгоритм нахождения симметрической разности и сохраняет результат работы алгоритма в выходной файл и завершает работу.

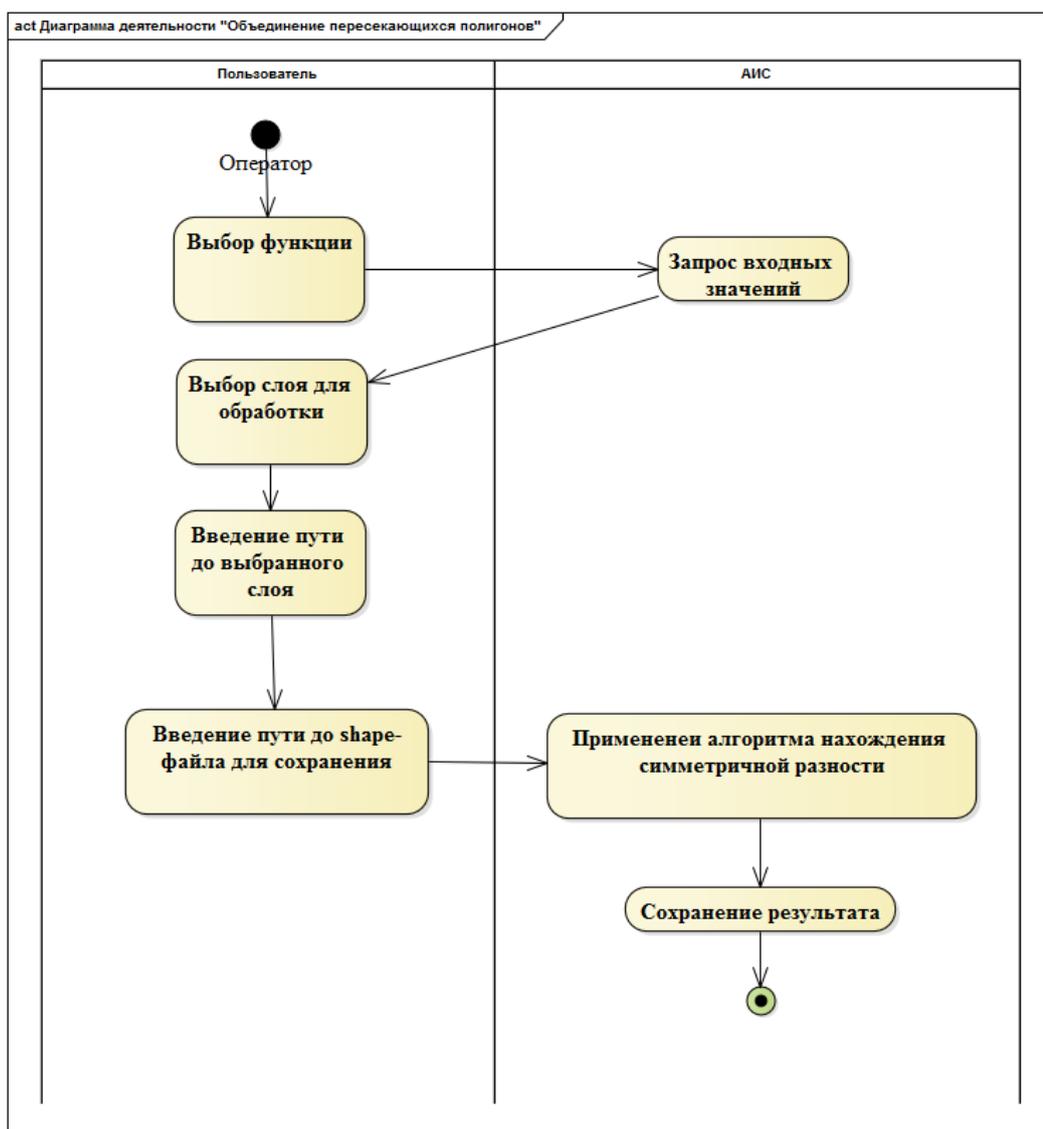


Рисунок 2.3 — Диаграмма деятельности модуля получения симметрической разности

Диаграмма деятельности модуля преобразования заданных полигонов к выпуклой форме без пересечений изображена на рисунке 2.4. Оператор запускает программный модуль в ответ система запрашивает входные значения. Оператор выбирает файл для обработки и вводит в систему путь к файлу, затем вводит путь к файлу выходных данных. После чего система применяет алгоритм преобразования заданных полигонов к выпуклой форме без пересечений и сохраняет результат работы алгоритма в выходной файл и завершает работу.

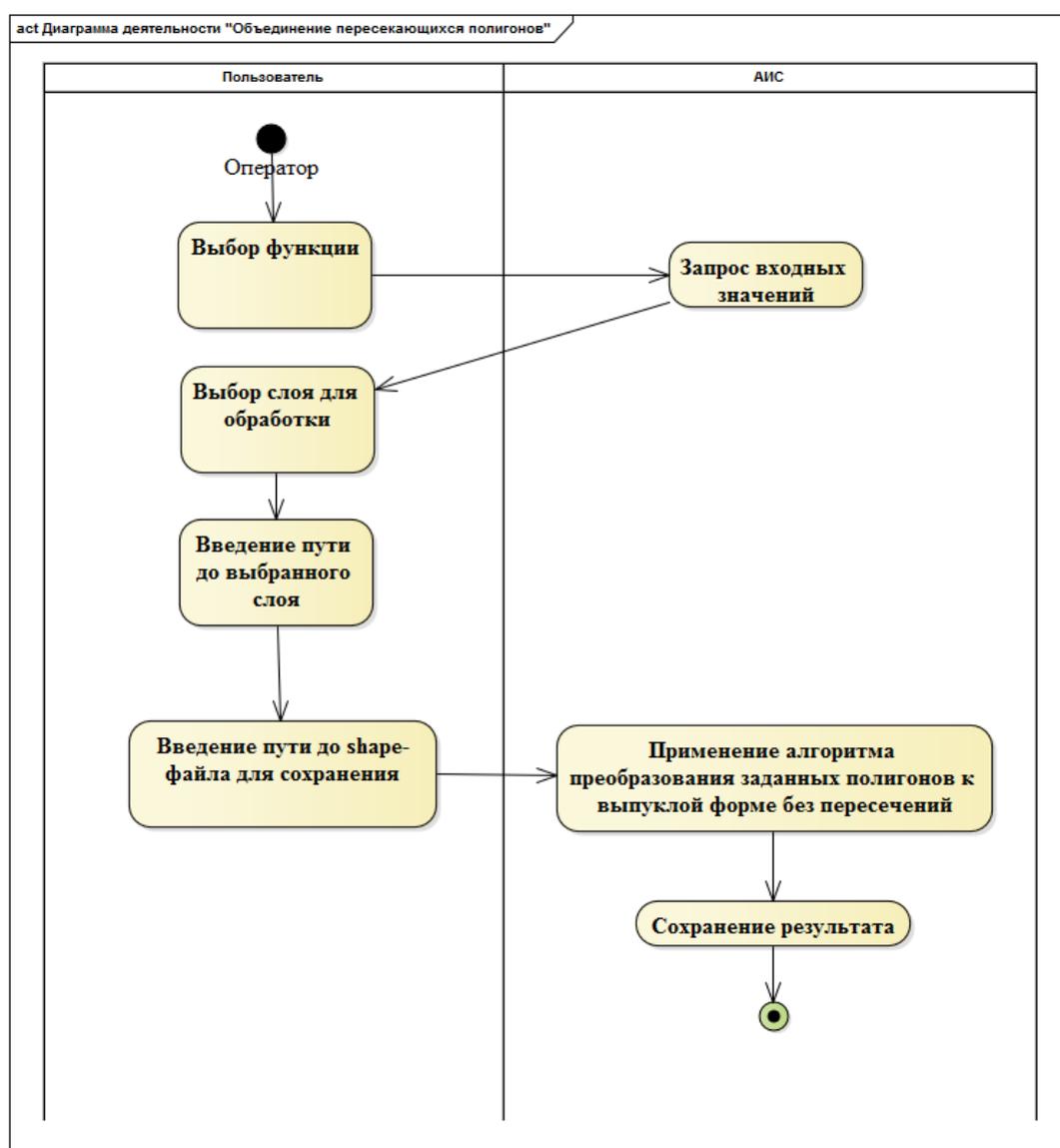


Рисунок 2.4 — Диаграмма деятельности модуля преобразования заданных полигонов к выпуклой форме без пересечений

## 2.2 Реализация модуля

Для создания программного модуля использовался высокоуровневый язык программирования Python. Для работы с географическими данными использована географическая информационная система Quantum GIS (QGIS). А также реализованы некоторые функции, которые необходимы для решения поставленной задачи.

### 2.2.1 Основной язык разработки

Python — язык общего назначения, производительный и мощный. В стандартной библиотеке содержится много полезных методов. Синтаксис очень минималистичен, что позволяет в короткие сроки начать писать на этом языке.

Python позволяет разбивать программы на модули, которые затем могут быть использованы в других программах. Python поставляется с большой библиотекой стандартных модулей, которые можно использовать как основу для программ или в качестве примеров при изучении языка. Стандартные модули предоставляют средства для работы с файлами, системных вызовов, сетевые соединения и даже интерфейсы к различным графическим библиотекам.

Дело в том, что экосистема языка Python по-настоящему большая и существует бесчисленное множество библиотек, которые упрощают решение поставленных задач в разных предметных областях. Это не только ускоряет процесс написания программы, но и уменьшает объем кода, что не маловажно в крупных проектах. Рассмотрим наиболее популярные библиотеки для языка программирования Python.

На крупнейшем веб-сервисе для хостинга IT-проектов GitHub можно найти библиотеки в открытом доступе. Ниже представлены самые популярные библиотеки:

– Requests – используется для уменьшения ручной работы с HTTP-запросами.

- Simplejson – библиотека которая поможет кодировать и декодировать json-файлы.
- SQL Alchemy – библиотека для работы с базами данных.
- TensorFlow – библиотека для произведения расчетов с использованием графов потоков данных.
- Scrapy – парсер веб-сайтов.
- Matplotlib – библиотека 2D числовых построений.
- Scikit-Learn – набор методов для работы в проектах с машинным обучением.
- Pygame – библиотека для создания приложений работающих с мультимедиа.
- Arrow – используется для замены стандартных методов для работы с датой и временем.
- wxPython – графическая библиотека для Python [4].

### **2.2.2 Программа для работы с геоданными**

Для работы с геоданными требуется программа, которая может производить векторные операции и открывать файлы векторного формата. Географическая информационная система Quantum GIS (ГИС QGIS), скриншот которой изображен на рисунке 2.5, удовлетворила всем ниже приведенным требованиям:

- Наличие опыта работы с ГИС.
- QGIS — полноценная географическая система с открытым исходным кодом.
- QGIS — кроссплатформенная информационная система, которая может работать на всех популярных операционных системах Windows, Linux, MacOS X, BSD, Android.

– QGIS имеет набор стандартных методов для работы с векторными данными.

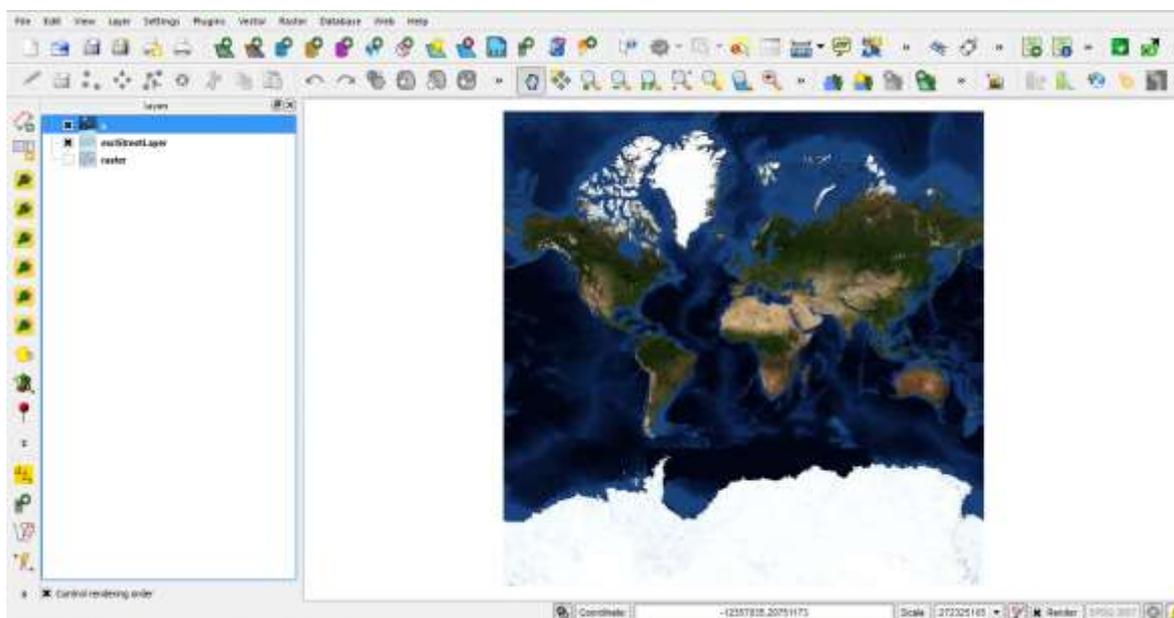


Рисунок 2.5 — ГИС Quantum GIS

Программа может работать с векторными и растровыми данными, создавать новые проекты, редактировать их, конвертировать в другие форматы.

Некоторые возможности геоинформационной системы представлены в списке ниже:

- Работа с share-файлами;
- Возможность геокодирования;
- OpenStreetMap – новые возможности в области визуализации и редактирования файлов;
- PluginBilder – плагин обеспечивающий создание новых модулей программы;
- PluginReloader – плагин для перезагрузки модулей (если внесены поправки в код модуля, его необходимо перезагрузить для отображения изменений);
- Управление свойствами объектов векторного формата;

– Совмещение изображений векторного и растрового формата и наложение их друг на друга.

Интерфейс программы минималистичен, интуитивен и имеет возможность настройки рабочей области под конкретный тип задачи.

Интерфейс QGIS предоставляет быстрый доступ к следующим функциям:

- добавление нового слоя;
- возможность добавлять к рабочим объектам подписи;
- возможность выделения необходимых объектов по средствам инструментов выделения;
- компоновка карт;
- возможность добавить и изменить атрибуты объектов;
- функция масштабирования;
- возможность установить стороны света, сделать масштабную линейку и легенду;
- быстрая загрузка данных и сохранение [5].

### **2.2.3 Программа для разработки интерфейса**

Для разработки интерфейса модуля использовалась программа Qt Creator, которая имеет набор необходимых функций для решения этой задачи. Интерфейс программы вы можете наблюдать на рисунке 2.6. Есть возможность создавать интерфейс модуля на языке XML или создавать интерфейс путем «перетаскивания элементов управления», при котором XML код генерируется автоматически. В обоих случаях полученный XML код сохраняется в файл с расширением \*.ui. Qt это фреймворк для разработки программного обеспечения, который используется для разработки приложений, работающих на Windows, Mac, Linux, а также на различных мобильных операционных системах. Сам QGIS написан с использованием Qt [6].

## 2.2.4 Интерфейс модуля

В автоматизированном режиме интерфейс модуля содержит единственную кнопку – «Выполнить» по нажатию на которую запускается алгоритм. Интерфейс модуля показан на рисунке 2.6. В автоматическом режиме модуль запускается из интерпретатора Python – OSGeo4W, где пользователю необходимо ввести входные данные, а именно входной и выходной слой.

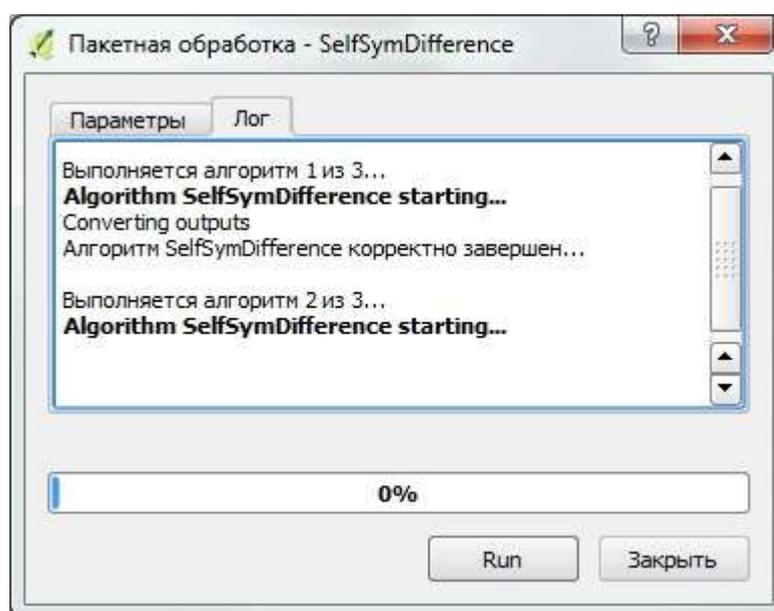


Рисунок 2.6 — Интерфейс модуля

Модуль открывается из интерфейса программы QGIS из списка пользовательских скриптов. На рисунке 2.7 показано, как модуль выглядит в программе QGIS.

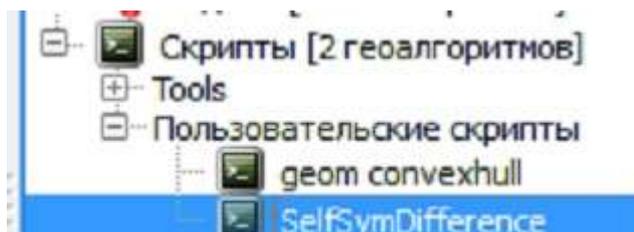


Рисунок 2.7 – Модуль нахождения симметрической разности полигонов в QGIS

### **2.2.5 Технические требования к программному окружению**

Язык Python — объекто-ориентированный язык программирования, но не компилируемый, как большинство его конкурентов, а интерпретируемый. Это принципиально отличает его от других языков, ведь для того, чтобы код программы на языке Python исполнился необходимо его интерпретировать, а затем выполнить командами. Из этого вытекает, что для использования языка Python необходимо установить интерпретатор языка Python. Интерпретатор — это программа, которая выполняет написанный код на конкретно языке программирования.

Также для того, чтобы программировать на Python необходима операционная система. Так как Python является кроссплатформенным, то на нем можно писать программы под любую операционную систему.

### **2.2.6 Выполняемые функции**

Функция определения активного слоя необходимая для определения индекса слоя, чтобы в дальнейшем к нему обращаться.

Функция нахождения симметрической разности выбранных полигонов.  
Функция преобразования выбранных полигонов к выпуклой форме без пересечений.

Входные данные программы организованы в виде файла формата \*.shp. Данные файлы должны находиться в папке с открытым проектом на локальном или внешнем носителе, который отформатирован согласно требованиям операционной системы.

Выходные данные программы создаются автоматически в процессе ее выполнения и сохраняются на локальный или внешний носитель в папке с проектом в формате \*.shp.

### **2.2.7 Описание кода модуля нахождения симметрической разности векторных полигонов.**

В приложении А представлен программный код модуля симметрической разности. Происходит подключение необходимых библиотек, а конкретно библиотеки OGR. Далее происходит создание ряда переменных:

- Переменная для записи активного слоя;
- Переменная для входного слоя;
- Переменная для выходного слоя.

Считываются входные данные, считывается входной слой, создается shp-файл для выходного слоя. Считывается геометрий векторных объектов из слоя входного файла. Начинается цикл в котором происходит обход всех объектов, затем во вложенном цикле происходит присвоение текущему объекту симметрической разности полученной в результате обхода всех остальных полигонов. Далее сохраняем полученную геометрию в слой и завершаем циклы. Записываем выходные данные в выходной shp-файл.

### **2.2.8 Блок-схема алгоритма нахождения симметрической разности векторных объектов**

Симметрическая разность двух множеств — теоретико-множественная операция, результатом которой является новое множество, включающее все элементы исходных множеств, не принадлежащие одновременно обоим исходным множествам. Другими словами, если есть два множества А и В, их симметрическая разность есть объединение элементов А, не входящих в В, с элементами В, не входящими в А.

Ниже на рисунке 2.8 представлена блок-схема алгоритма нахождения симметрической разности векторных объектов.

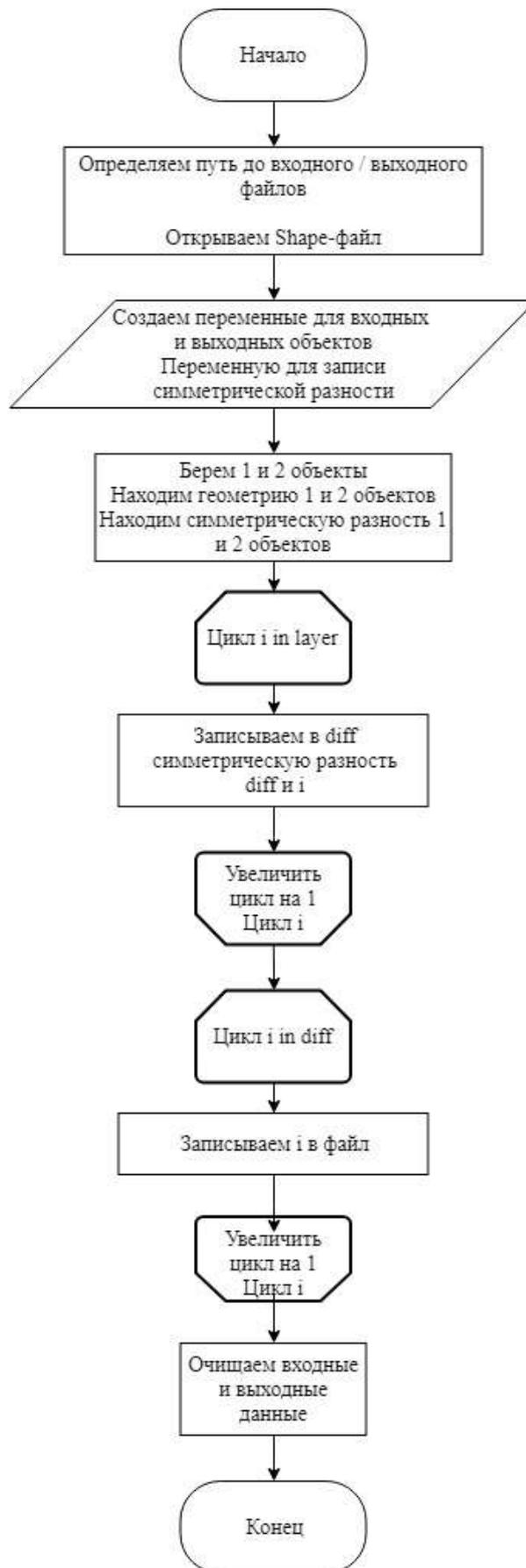


Рисунок 2.8 — Блок-схема алгоритма нахождения симметрической разности

## 2.2.9 Результат работы алгоритмов

По шаговый результат работы алгоритмов можно наблюдать на рисунках ниже. Как видно на рисунке 2.9 изображен результат работы алгоритма нахождения симметрической разности. Симметрическая разность найдена верно.

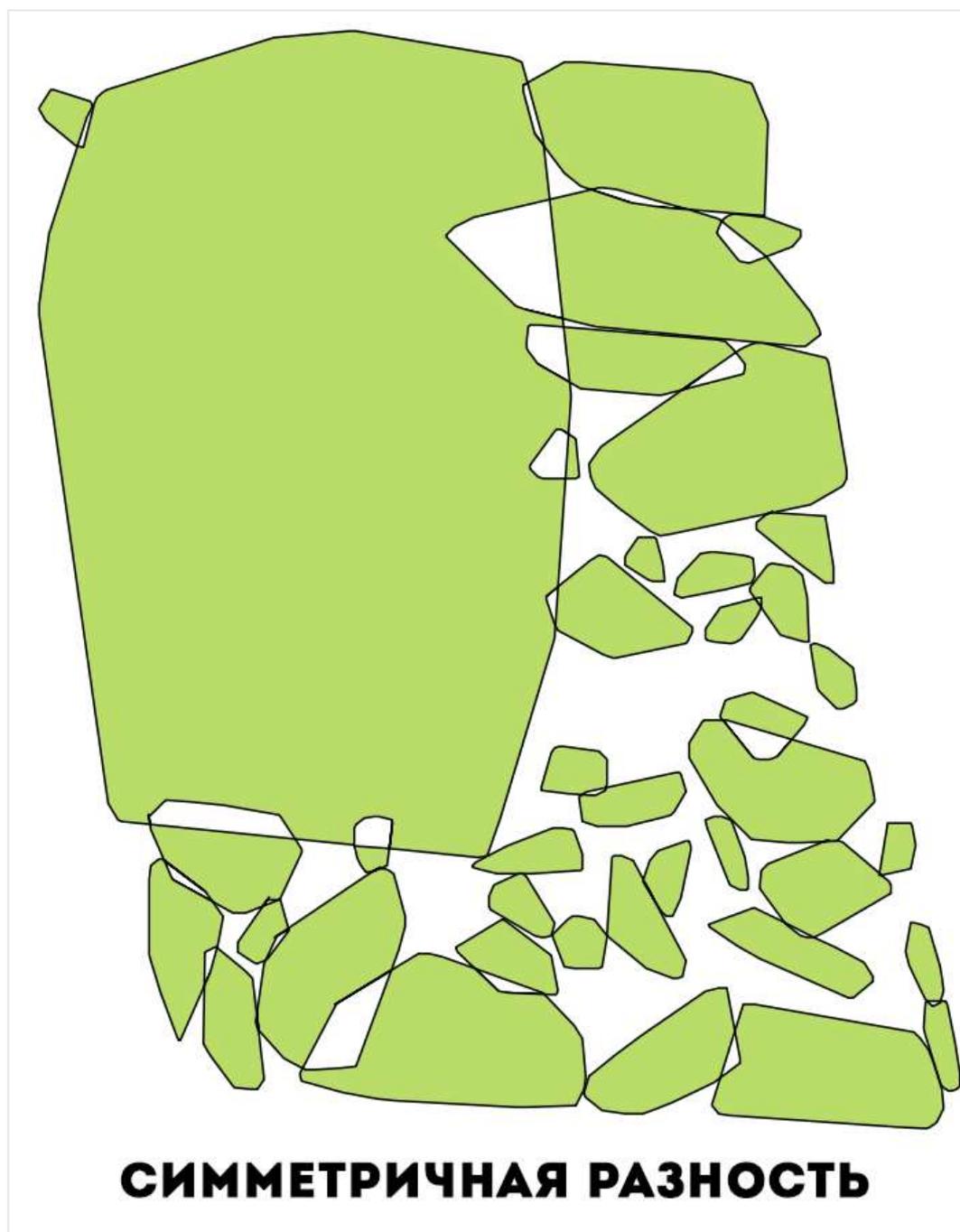


Рисунок 2.9 — Результат работы алгоритма нахождения симметрической разности

После применения алгоритма симметрической разности применяем алгоритм преобразования полигонов к выпуклой форме без пересечений. Результат работы данного алгоритма можем наблюдать на рисунке 2.10.

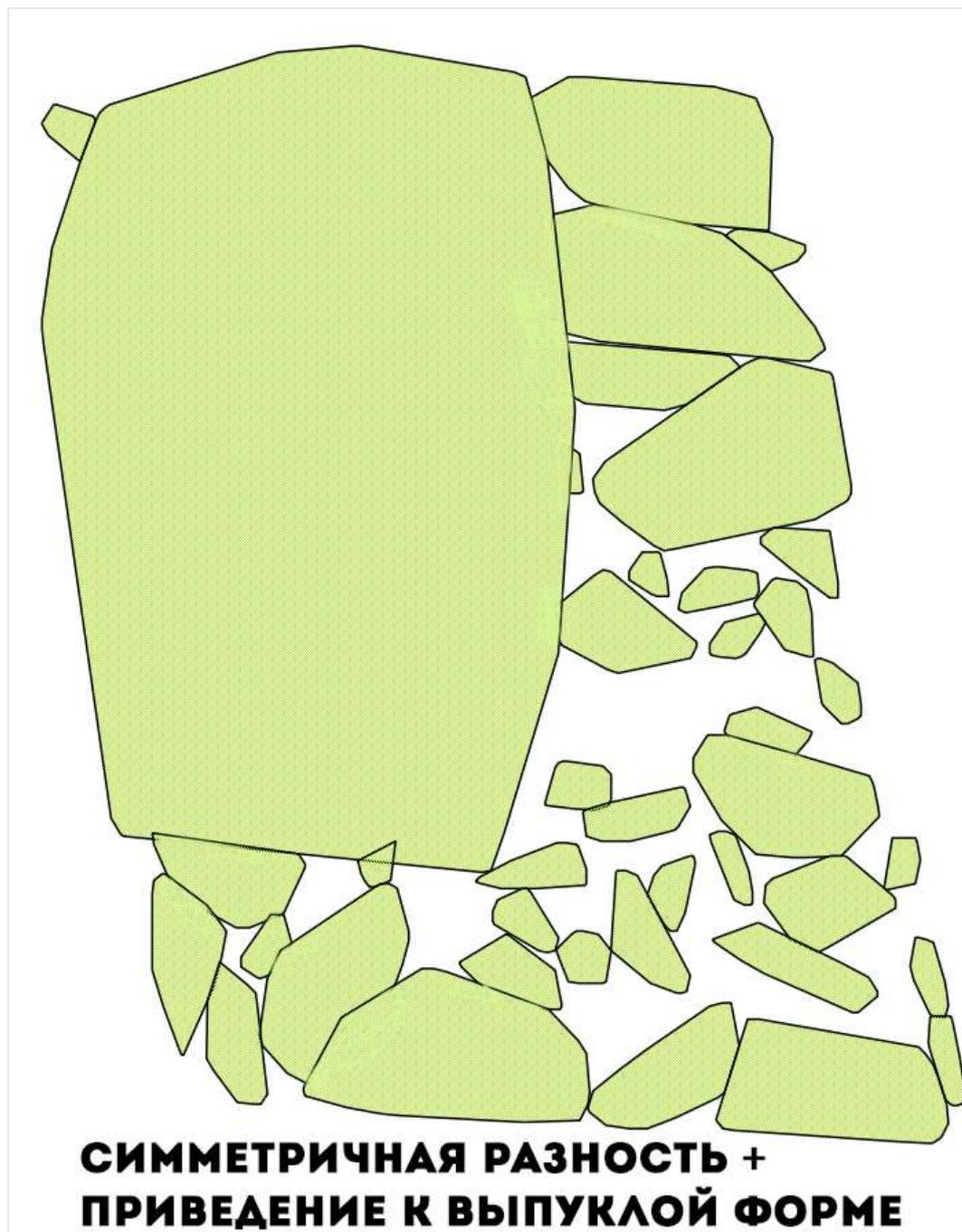


Рисунок 2.10 — Результат работы алгоритма нахождения симметрической разности

## 2.2.10 Вывод по главе 2

В данной главе проведена работа по проектированию и реализации разрабатываемого модуля для геоинформационной системы QGIS. Разработаны диаграммы, описывающие процессы системы.

Диаграммы вариантов использования и деятельности являлись основой для реализации программных модулей. В результате проделанной работы разработаны программные модули:

- Нахождение симметрической разности векторных полигонов
- Преобразование векторных полигонов к выпуклой форме без пересечений

## ЗАКЛЮЧЕНИЕ

В процессе выполнения бакалаврской работы изучена предметная область алгоритмов получения симметрической разности и алгоритма преобразования заданных полигонов к выпуклой форме без пересечений в которой планируется применять созданный программный модуль, сформированы требования к системе, функциональные возможности, а также изучены средства для разработки.

Спроектированы модели взаимодействия оператора и системы в виде:

- диаграмма вариантов использования;
- диаграммы деятельности.

Данные диаграммы описывают функционал программного модуля в полной мере.

Разработаны блок-схемы алгоритмов модуля анализа локальных неоднородностей сельскохозяйственных земель.

Реализованы алгоритмы:

- Нахождения симметрической разности выбранных полигонов;
- Преобразования выбранных полигонов к выпуклой форме без пересечений.

Проведена экспериментальная апробация разработанного модуля.

Поставленные цель и задачи выполнены, разработанный модуль успешно встроен в ГИС QGIS, где работает в автоматизированном режиме. А также модуль встроен в систему агромониторинга на основе действующего программно-аппаратного комплекса центра ДЗЗ Сибирского федерального университета.

## СПИСОК СОКРАЩЕНИЙ

В данной работе применены следующие сокращения:

АИС – автоматизированная информационная система;

ДЗЗ – дистанционное зондирование Земли;

ГИС – географическая информационная система;

ИС – информационная система;

ПО – программное обеспечение;

ИТ – информационные технологии;

SQL – язык структурированных запросов;

UML – унифицированный язык моделирования;

WEB – всемирная паутина;

XML – расширяемый язык разметки.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Технологии точного земледелия, их перспективы и возможности. Агрохимия и почва. [Электронный ресурс] — URL: <http://selo-delo.ru> (дата обращения: 12.03.2018)
2. Точное земледелие: принцип работы и перспективы. Сельхозпортал. [Электронный ресурс] — URL: <http://сельхозпортал.рф/> (дата обращения: 12.04.2018)
3. Афанасьев, Р. А. Проблемы координатного земледелия и пути их решения / Р.А. Афанасьев // Доклады ТСХА. — 2006. — № 278. — С. 187–190.
4. Балабанов, В. И. Навигационные системы в сельском хозяйстве. Координатное земледелие : учебное пособие / В. И. Балабанов. — Москва : Изд-во РГАУ-МСХА им. К. А. Тимирязева, 2013. — 143с.
5. Десятка лучших библиотек для программирования на Python. Techrocks.ru | программирование, стартапы, новости IT. [Электронный ресурс] — URL: <https://techrocks.ru/> (дата обращения: 01.05.2018)
6. Установка модулей расширения в QGIS. GIS-Lab: Геоинформационные системы и Дистанционное зондирование Земли. [Электронный ресурс] — URL: <http://gis-lab.info/> (дата обращения 03.05.2018)
7. Якушев, В. П. Точное земледелие: состояние исследований и задачи агрофизики / В. П. Якушев, Р. А. Полуэктов [и др.] // В кн.: «Агрофизические и экологические проблемы сельского хозяйства в 21 веке». — СПб., 2002. — Т. 3. — С. 26–73.
8. Якушев, В. П. Информационное обеспечение точного земледелия / В. П. Якушев, В. В. Якушев. — СПб.: Изд-во ПИЯФ РАН, 2007. — 384 с.
9. Qt 4.3: Qt Creator Документация. Все о кроссплатформенном программировании | Документация. [Электронный ресурс] — URL: <http://www.doc.crossplatform.ru/> (дата обращения: 03.05.2018)

10. Batte, T. M. Precision farming adoption and use in Ohio: Case studies of six leading-edge adopters. / T. M. Battle, W. M. Arnholt // *Computers and Electronics in Agriculture* — 2003. — N38. — P. 125-139.

11. Blackmore, B. S. The role of Precision Farming in Sustainable Agriculture; A European Perspective. / B. S. Blackmore, P. N. Wheeler, J. Morris, P. C. Robert, R. H. Rust, W. E. Larson // Presented at 2nt Int. Conf. on Site-Specific Management for Agricultural Systems, — USA, Minneapolis, March 1995.

12. Gould, J. D. How to design usable systems. / J. D. Gould // — New York, 1995. — P. 93-121.

13. Lutz M. Learning Python, 5th Edition: tutorial / Mark Lutz // — Ph. O'Reilly Media, 2013. — 1594 pgs.

14. Langran, G. Time in Geographical Information Systems. / G. Langran // ed. Taylor & Francis, London. — 1992. — 193 pgs.

## ПРИЛОЖЕНИЕ А

Листинг модуля нахождения симметрической разности полигонов

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
# подключаем библиотеки
from osgeo import ogr
import sys
import os
import math
import itertools
# находим текущий путь
currentPath = os.path.dirname(sys.argv[0])
#объявляем функцию Симметрической разности
def symmetricDiff(inputShapefile, outShapefile =
currentPath):
    # Открываем SHP-файл
    inputData = ogr.Open(inputShapefile, False)
    # Проверяем, что он не пустой
    if inputData is None:
        print("ERROR: open failed")
        sys.exit(1)
    # Записываем 0-ой слой в Layer
    layer = inputData.GetLayer(0)
    # Проверяем, что слой не пустой
    if layer is None:
        print("ERROR: reading file")
        sys.exit(1)
    # Получаем доступ к атрибутам
```

```

inFeatDef = layer.GetLayerDefn()
# Создаем новый shp-файл
drv = ogr.GetDriverByName("ESRI Shapefile")
# Удаляем shp-файл, если он уже создан
if os.path.exists(outShapefile):
    drv.DeleteDataSource(outShapefile)
# Создаем DataSet
outputData = drv.CreateDataSource(outShapefile)
# Создаем outputLayer - выходной слой
outputLayer = outputData.CreateLayer(outShapefile, None,
ogr.wkbPolygon)
# Добавляем новые атрибуты в выходной слой
idField = ogr.FieldDefn("id", ogr.OFTInteger)
areaField = ogr.FieldDefn("area", ogr.OFTReal)
areaField.SetWidth(10)
areaField.SetPrecision(6)
perimeterField = ogr.FieldDefn("perimeter", ogr.OFTReal)
perimeterField.SetWidth(10)
perimeterField.SetPrecision(4)
thicknessField = ogr.FieldDefn("thickness", ogr.OFTReal)
thicknessField.SetWidth(10)
thicknessField.SetPrecision(4)
outputLayer.CreateField(idField)
outputLayer.CreateField(areaField)
outputLayer.CreateField(perimeterField)
outputLayer.CreateField(thicknessField)
# Создаем схему (таблица атрибутов) слоя
featureDefn = outputLayer.GetLayerDefn()
feature = ogr.Feature(featureDefn)

```

```

first = next(layer) # берем 1 объект слоя
second = next(layer) # берем 2 объект слоя
first_geom = first.GetGeometryRef() # находим геометрию
1 объекта
second_geom = second.GetGeometryRef() # находим
геометрию 2 объекта
diff = first_geom.SymDifference(second_geom) # считаем
Симметрическую разность
print "\nFirst step of processing...",
for geom in layer:
    diff = diff.SymDifference(geom.GetGeometryRef())
print("completed")
print("\nSecond step of processing..."),
id = 0 # переменная для записи id объекта
# разделяем полученный мультиполигон на части
try:
    for part in diff:
        areaImKvadMeters = part.GetArea()
        perimeterInMeters = part.Boundary().Length()
        try:
            area = areaImKvadMeters / 10000
            perimeter = perimeterInMeters / 1000
            thickness = (4 * 3.14) * (areaImKvadMeters /
(perimeterInMeters**2))
        except ZeroDivisionError:
            area = 0
            perimeter = 0
            thickness = 0
        feature.SetField("id", id)

```

```

        feature.SetField("area", area)
        feature.SetField("perimeter", perimeter)
        feature.SetField("thickness", thickness)
        # записываем результат в файл
        feature.SetGeometry(part)
        outputLayer.CreateFeature(feature)
        id+=1
    except Exception as e:
        print(e),
    else:
        print("completed")
    print("\nProcessed " + str(id) + " objects")
    #отчищаем входные/выходные данные
    feature = None
    inputData.Destroy()
    outputData.Destroy()
def main():
    if len(sys.argv) != 3:
        print ("\n\nERROR PARAMETERS. EXAMPLE:
\nsymmetric_diff.py input.shp output.shp\n")
        sys.exit(1)
    else:
        inputSHPPath = sys.argv[1]
        outputSHPPath = sys.argv[2]
        symmetricDiff(inputSHPPath, outputSHPPath)
        print("\nSuccessfully. Symmetric difference is
found.\nThe result in: " + sys.argv[2])
if __name__ == "__main__":
    main()

```

## ПРИЛОЖЕНИЕ Б

Листинг модуля преобразования заданных полигонов к выпуклой форме без пересечений

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-
from osgeo import ogr
import sys
import os
import math

currentPath = os.path.dirname(sys.argv[0])
def convexHull(inputShapefile, outShapefile = currentPath +
"/temp/convexhull.shp", inThickness='0', inArea='0'):
    # Open base SHP File
    inputData = ogr.Open(inputShapefile, False)
    if inputData is None:
        print("ERROR: open failed")
        sys.exit(1)
    # Get once Layer from SHP File
    inputLayer = inputData.GetLayer(0)
    if inputLayer is None:
        print("ERROR: reading file")
        sys.exit(1)
    filter = ""
    if inThickness != '0':
        filter += "thickness <= " + inThickness
    if inArea != '0':
        filter += " and area <= " + inArea
    # "thickness <= 0.2 and area <= 50"
    if filter != "":
```

```

    print(filter)
    inputLayer.SetAttributeFilter(filter)
# Dostup k atributam
inFeatDef = inputLayer.GetLayerDefn()
# Create new SHP file
drv = ogr.GetDriverByName("ESRI Shapefile")
# Create exit dataset
# outShapefile = currentPath + "/temp/states_convexhull.shp"
# Remove output shapefile if it already exists
if os.path.exists(outShapefile):
    drv.DeleteDataSource(outShapefile)
outputData = drv.CreateDataSource(outShapefile)
# Create exit Layer
outputLayer = outputData.CreateLayer(outShapefile, None,
ogr.wkbPolygon)
# Add new attributes to output Layer
idField = ogr.FieldDefn("id", ogr.OFTInteger)
areaField = ogr.FieldDefn("area", ogr.OFTReal)
areaField.SetWidth(10)
areaField.SetPrecision(6)
perimeterField = ogr.FieldDefn("perimeter", ogr.OFTReal)
perimeterField.SetWidth(10)
perimeterField.SetPrecision(4)
thicknessField = ogr.FieldDefn("thickness", ogr.OFTReal)
thicknessField.SetWidth(10)
thicknessField.SetPrecision(4)
outputLayer.CreateField(idField)
outputLayer.CreateField(areaField)
outputLayer.CreateField(perimeterField)

```

```

outputLayer.CreateField(thicknessField)
# Create the feature and set values
featureDefn = outputLayer.GetLayerDefn()
feature = ogr.Feature(featureDefn)
# ID value
k = 0
# chitaem postrochno file shp
for inFeature in inputLayer:
    k = k + 1
    geom = inFeature.GetGeometryRef()
    # print(geom)
    # Out geometry type
    # print geom.GetGeometryName()
    # Calculate Convex Hull
    convexhull = geom.ConvexHull()
    # print(convexhull)
    # Area in meters kvadr
    areaImKvadMeters = convexhull.GetArea()
    # Area in kilometers kvadr
    # area = area / 1000000
    # Area in Gektar
    area = areaImKvadMeters / 10000
    # Perimeter in meters
    perimeterInMeters = convexhull.Boundary().Length()
    # Perimeter in kilometers
    perimeter = perimeterInMeters / 1000
    # Thickness in range 0..1
    thickness = (4 * 3.14) * (areaImKvadMeters /
(perimeterInMeters**2))

```

```

feature.SetField("id", k)
# feature.SetField("CLASS", cl)
feature.SetField("area", area)
feature.SetField("perimeter", perimeter)
feature.SetField("thickness", thickness)
feature.SetGeometry(convexhull)
outputLayer.CreateFeature(feature)

feature = None
# Save and close DataSource
inputData.Destroy()
outputData.Destroy()

# =====
if __name__ == "__main__":
    if len(sys.argv) < 5:
        print ("\n\nERROR PARAMETERS. EXAMPLE: \ngeom_convexhull.py
C:/NDVI/ndvi/input.shp C:/NDVI/ndvi/output.shp [thickness_value]
[area_value]\n")
        sys.exit(1)
    else:
        inputSHPPath = sys.argv[1]
        outputSHPPath = sys.argv[2]
        if sys.argv[3] != None or sys.argv[4] != None:
            inThickness = sys.argv[3]
            inArea = sys.argv[4]
            convexHull(inputSHPPath, outputSHPPath, inThickness, inArea)
        else:
            convexHull(inputSHPPath, outputSHPPath)
        # print("P1: {}, P2: {}, P3: {}, P4: {}".format(sys.argv[1],
sys.argv[2], sys.argv[3], sys.argv[4]))

```



## ЗАДАЧИ:

- Обзор методов обработки неоднородных участков растительности в задачах точного земледелия.
- Разработка блок-схем алгоритмов для модуля анализа локальных неоднородностей сельскохозяйственных земель.
- Разработка программного модуля, реализующего алгоритма нахождения симметрической разности векторных полигонов и алгоритма преобразования полигонов к выпуклой форме без пересечений.
- Экспериментальная апробация разработанного модуля.



Рисунок В.3 — Плакат презентации № 3

## АКТУАЛЬНОСТЬ

- Зачем нужен этот алгоритм?
- Приведение векторизованных с/х полей к более удобной форме для их последующей обработки с/х техникой
- Исключение наложения контуров и приведение к выпуклой форме
- Что позволит экономить время работы с/х техники, сократить затраты на обработку полей



Рисунок В.4 — Плакат презентации № 4

## ВХОДНЫЕ ДАННЫЕ

Входные данные были получены со спутника World-View-2 с пространственным разрешением 0,5 метров, дата получения: 25 июня 2016 года

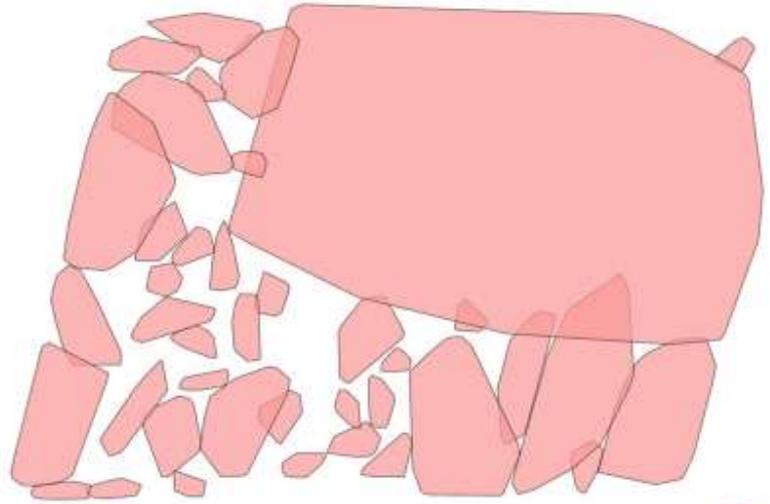


Рисунок 1 — Входные данные

Рисунок В.5 — Плакат презентации № 5

## USE CASE. ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ



Рисунок 2 — Диаграмма вариантов использования

Рисунок В.6 — Плакат презентации № 6

## USE CASE. ДИАГРАММЫ ДЕЯТЕЛЬНОСТИ

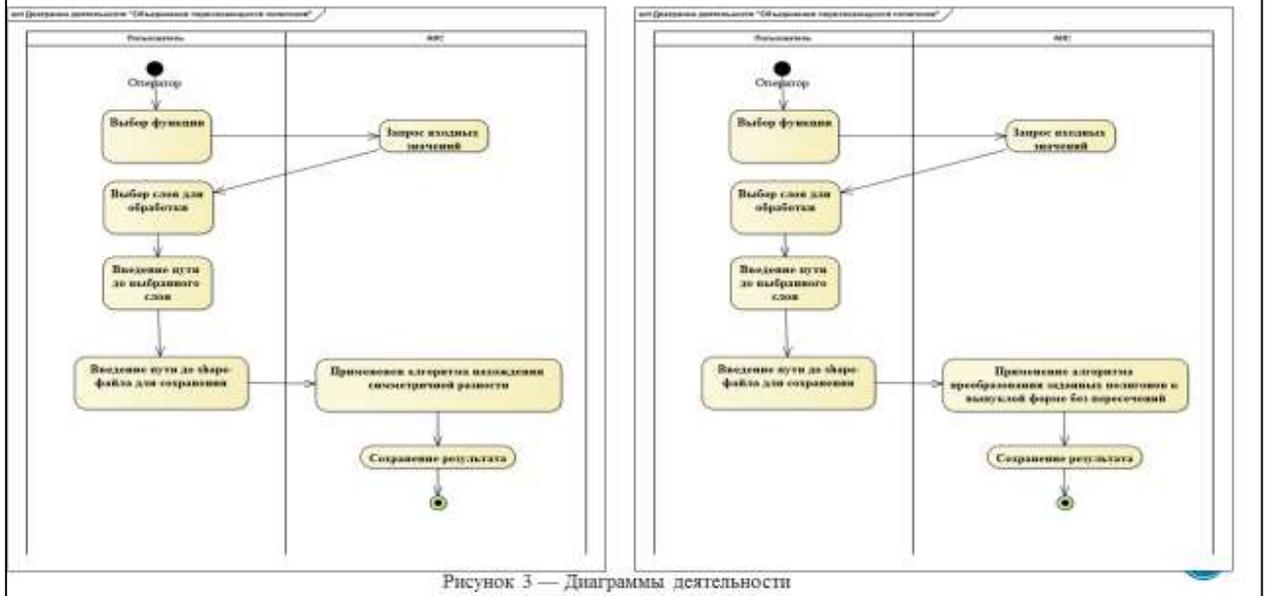


Рисунок В.7 — Плакат презентации № 7

## ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОКРУЖЕНИЮ

- Интерпретатор python OSGeo4W
- Среда выполнения кроссплатформенная
- Библиотека GDAL/OGR



Рисунок В.8 — Плакат презентации № 8



Рисунок В.9 — Плакат презентации № 9

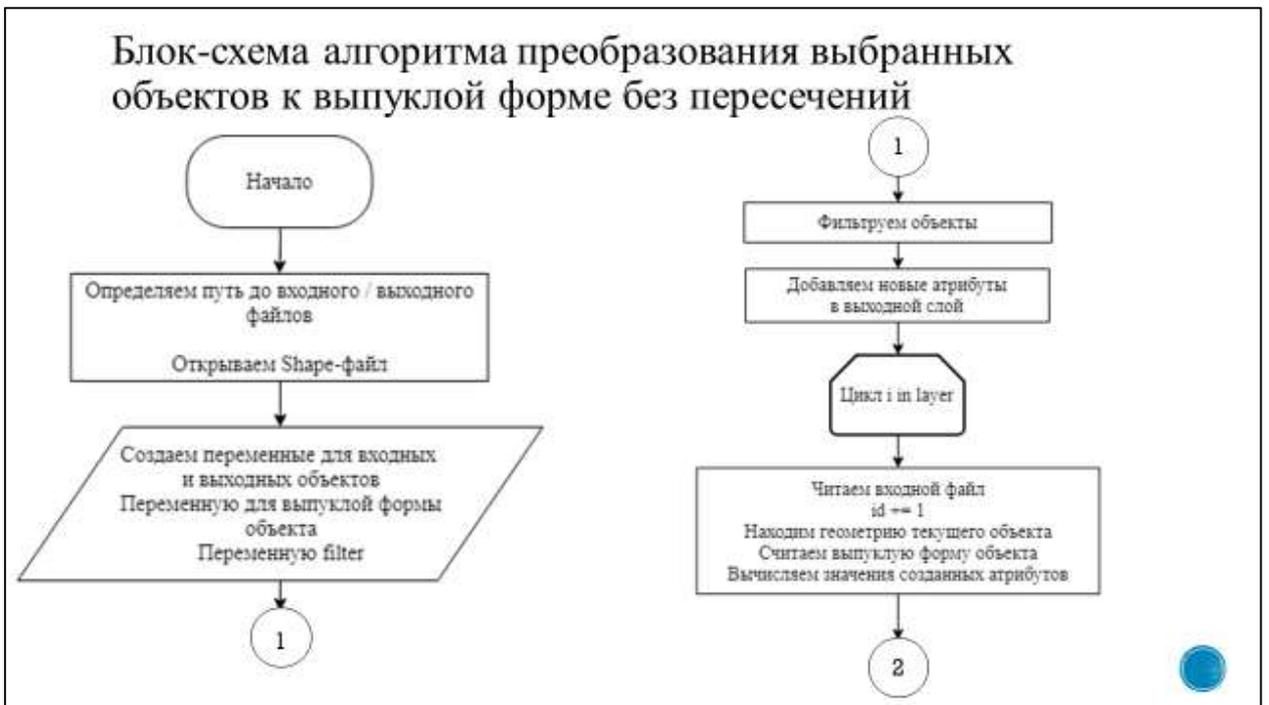


Рисунок В.10 — Плакат презентации № 10

## Блок-схема алгоритма преобразования выбранных объектов к выпуклой форме без пересечений

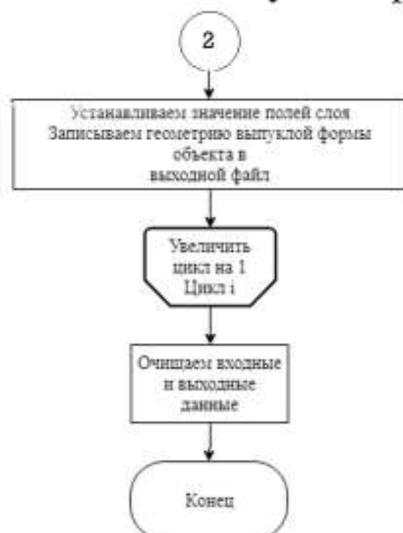


Рисунок В.11 — Плакат презентации № 11

## РЕЗУЛЬТАТ НАХОЖДЕНИЯ СИММЕТРИЧЕСКОЙ РАЗНОСТИ ОБЪЕКТОВ

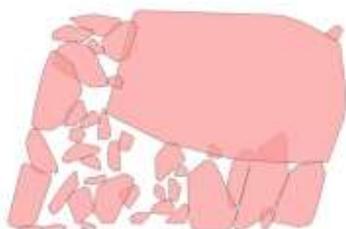


Рисунок 4 — Входные данные

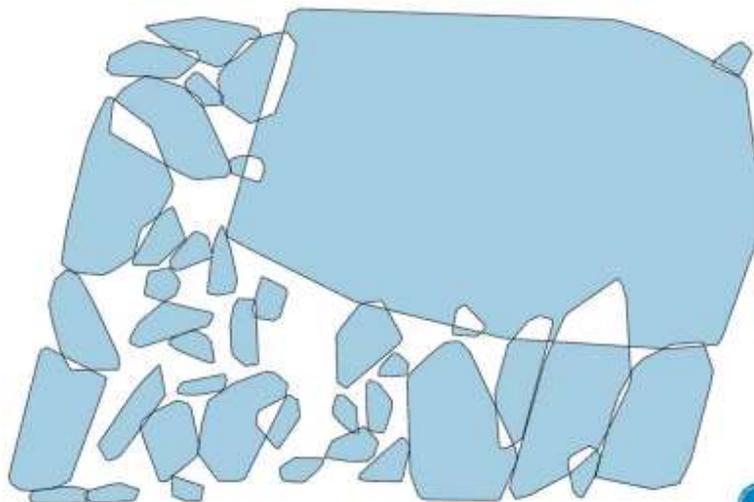


Рисунок 5 — Симметрическая разность

Рисунок В.12 — Плакат презентации № 12

## РЕЗУЛЬТАТ ПРЕОБРАЗОВАНИЯ ОБЪЕКТОВ К ВЫПУКЛОЙ ФОРМЕ БЕЗ ПЕРЕСЕЧЕНИЙ

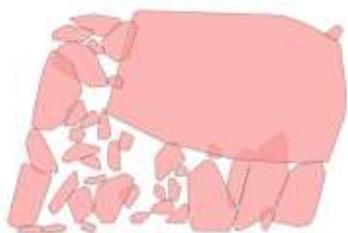


Рисунок 6 — Входные данные

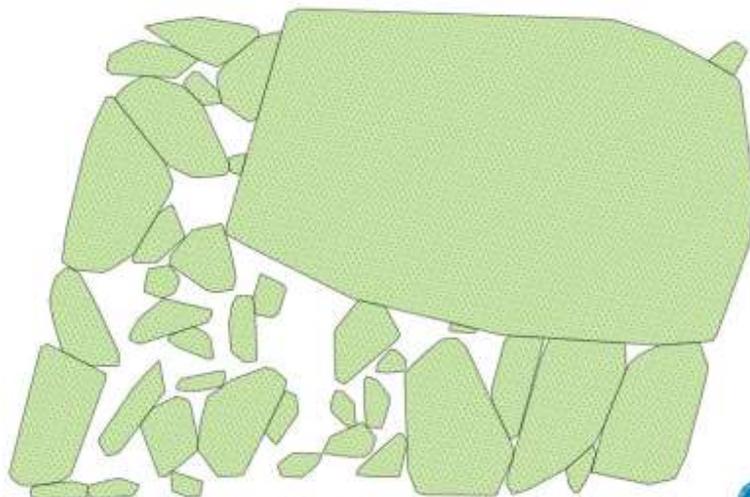


Рисунок 7 — Выпуклая форма объектов

Рисунок В.13 — Плакат презентации № 13

# ПРИЛОЖЕНИЕ Г

## Отчет в системе «Антиплагиат»

Министерство образования и науки Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования «Сибирский федеральный университет»				
<b>НАУЧНАЯ БИБЛИОТЕКА</b>				
660049, Красноярск, пр. Свободный ,79/10, тел.(3912) 2-912-820, факс (3912) 2-912-773 E-mail: bik@sfu-kras.ru				
ОТЧЕТ о результатах проверки в системе «АНТИПЛАГИАТ»				
Автор: Семенов Матвей Олегович				
Заглавие: Разработка модуля анализа локальных неоднородностей сельскохозяйственных земель				
Вид документа: Выпускная квалификационная работа бакалавра				
По результатам проверки оригинальный текст составляет 84,53%				
Источник	Коллекция / модуль поиска	Ссылка на источник	Доля в отчете	Доля в тексте
Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Учебное пособие	bibliorossica	<a href="http://www.bibliorossica.com/book.html?&amp;currBookId=12103">http://www.bibliorossica.com/book.html?&amp;currBookId=12103</a>	0,17	0,17
Теоретические и прикладные аспекты математики, информатики и образования. Материалы Международной научной конференции. Архангельск, 16–21 ноября 2014 г.	bibliorossica	<a href="http://www.bibliorossica.com/book.html?&amp;currBookId=19573">http://www.bibliorossica.com/book.html?&amp;currBookId=19573</a>	0,14	0,14
140619	directmedia	<a href="http://biblioclub.ru/index.php?page=book_red&amp;id=140619">http://biblioclub.ru/index.php?page=book_red&amp;id=140619</a>	0	0,41
Брежнев, Руслан Владимирович МОДЕЛИ И АЛГОРИТМЫ ИНФОРМАЦИОННОЙ ПОДДЕРЖКИ РЕШЕНИЯ ЗАДАЧ МОНИТОРИНГА ОБЪЕКТОВ НЕОДНОРОДНОЙ ПРОСТРАНСТВЕННОЙ СТРУКТУРЫ ПО ДАННЫМ ДИСТАНЦИОННОГО ЗОНДИРОВАНИЯ ЗЕМЛИ : диссертация ... кандидата технических наук : 05.13.17 Красноярск...	disser.rsl	<a href="http://dlib.rsl.ru/rsl01005000000/rsl01005103000/rsl01005103322/rsl01005103322.pdf">http://dlib.rsl.ru/rsl01005000000/rsl01005103000/rsl01005103322/rsl01005103322.pdf</a>	2,68	2,73
Матвеевко, Дмитрий Александрович диссертация ... кандидата сельскохозяйственных наук : 06.01.03 Санкт-Петербург 2012	disser.rsl	<a href="http://dlib.rsl.ru/rsl01005000000/rsl01005090000/rsl01005090450/rsl01005090450.pdf">http://dlib.rsl.ru/rsl01005000000/rsl01005090000/rsl01005090450/rsl01005090450.pdf</a>	0	0,77
Гостева, Анна Александровна диссертация ... кандидата технических наук : 05.11.13 Красноярск 2006	disser.rsl	<a href="http://dlib.rsl.ru/rsl01002000000/rsl01002977000/rsl01002977218/rsl01002977218.pdf">http://dlib.rsl.ru/rsl01002000000/rsl01002977000/rsl01002977218/rsl01002977218.pdf</a>	0,01	0,58
Якушев, Вячеслав Викторович диссертация ... доктора сельскохозяйственных наук : 06.01.03 Санкт-Петербург 2013	disser.rsl	<a href="http://dlib.rsl.ru/rsl01007000000/rsl01007953000/rsl01007953855/rsl01007953855.pdf">http://dlib.rsl.ru/rsl01007000000/rsl01007953000/rsl01007953855/rsl01007953855.pdf</a>	0	0,51

Рисунок Г.1 — Отчет в системе «Антиплагиат»