1. **Прототип интерфейса системы**

Прототипирование интерфейса программного обеспечения – этап разработки программного обеспечения, процесс создания прототипа программы ­– макета черновой версии программы, обычно – с целью проверки пригодности предлагаемых для применения концепций, архитектурных и технологических решений, а также для представления программы заказчику на ранних стадиях процесса разработки [6]. В ходе разработки был получен прототип интерфейса главного окна, который представлен на рисунке 10.

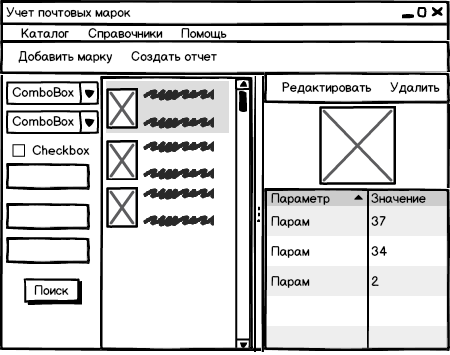


Рисунок 10 – Прототип главного окна

В пункте меню "Каталог" пользователь может выбрать команду создания отчета, создания новой записи или выхода из программы. При сохранении файла открывается соответствующее диалоговое окно.

В пункте меню "Справочники" пользователь может выбрать команду добавления, удаления или редактирования одного из параметров марки.

В пункте меню "Помощь" пользователь может выбрать команду открытия справочной информации в виде интернет-страницы.

Пользователь может задать необходимые критерии (параметры марки) – год выпуска, страна, тема, зубцовка и т.д. – и найти требуемую информацию, нажав кнопку «Поиск». Тогда в центре главного окна появится список марок, удовлетворяющих заданным критериям. Нажав левой кнопкой мыши на интересующую марку, пользователь сможет получить более подробную информацию о марке.

1. **UML–проект**

UML (Unified Modeling Language – унифицированный язык моделирования) – язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но в средствах выполнения UML-моделей как интерпретируемого кода возможна кодогенерация [7].

Использование UML не ограничивается моделированием программного обеспечения. Его также используют для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

UML позволяет также разработчикам программного обеспечения достигнуть соглашения в графических обозначениях для представления общих понятий (таких как класс, компонент, обобщение (generalization), объединение (aggregation) и поведение), и больше сконцентрироваться на проектировании и архитектуре.

UML содержит стандартный набор диаграмм и нотаций для моделирования самых разнообразных видов систем: систем программного обеспечения, технических средств и организационно-экономических систем [8]:

1. диаграммы вариантов использования (use case diagrams);
2. диаграммы классов (class diagrams);
3. диаграммы поведения системы (behavior diagrams):
   * диаграммы состояний (statechart diagrams);
   * диаграммы деятельностей (activity diagrams;
   * диаграммы взаимодействия (interaction diagrams):
   * диаграммы последовательности (sequence diagrams);
   * кооперативные диаграммы (collaboration diagrams).
4. диаграммы реализации (implementation diagrams):
   * диаграммы компонентов (component diagrams);
   * диаграммы размещения (deployment diagrams).

**Диаграмма вариантов использования**

Диаграмма вариантов использования описывает функциональное назначение системы, т.е. то, что система будет делать в процессе своего функционирования. Диаграмма вариантов использования является исходным концептуальным представлением или концептуальной моделью системы в процессе ее проектирования и разработки.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актантов, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актантом или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь вариант использования служит для описания сервисов, которые система предоставляет актанту.

Диаграмма вариантов использования разработанной системы представлена на рисунке 11. На данной диаграмме актант – пользователь, который имеет возможности загружать алгоритм из файла, сохранять алгоритм и трассу в файл. Пользователю предоставляются возможности создать алгоритм вручную или выбрать один из базовых алгоритмов, запустить алгоритм в одном из режимов демонстрации. Также пользователь может задать значения операндов в выбранном формате и посмотреть справочную информацию.

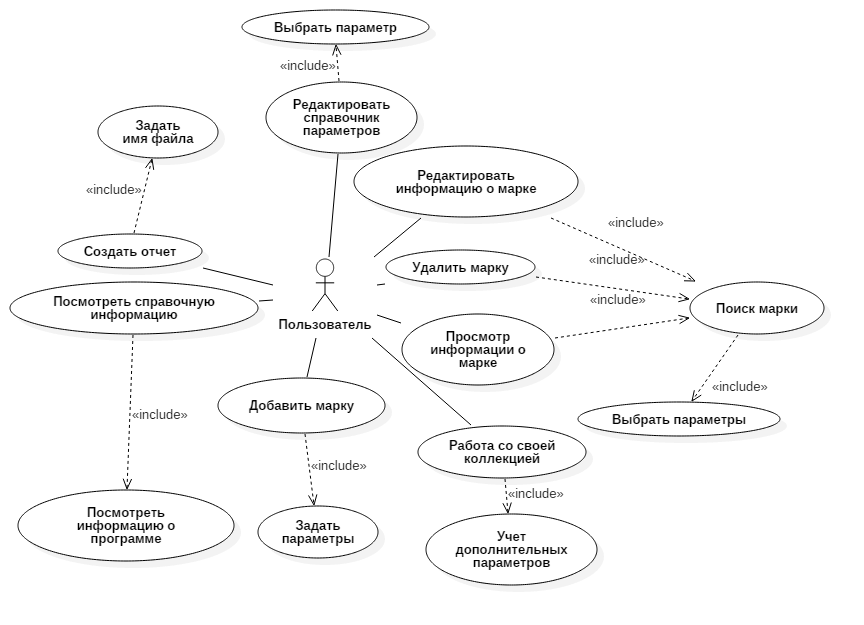


Рисунок 11 – Графическое представление автоматизированной системы учета почтовых марок

Каждый вариант использования должен быть задокументирован как некий сценарий работы с системой. Сценарий - определенная последовательность действий, которая описывает действия актеров и поведение моделируемой системы в форме обычного текста. Ниже приведены примеры оформления сценариев.

***Вариант использования:* Просмотр информации о марке.**

*Краткое описание.* Дает возможность пользователю просмотреть хранящуюся в системе информацию о конкретной интересующей марке. Информация, содержащая названия параметров и их значения, представлена в форме таблицы в главном окне приложения.

*Актант.* Пользователь.

*Предусловия.* Компьютер пользователя включён, на экране – главное окно программы « Автоматизированная система учета почтовых марок».

*Основной поток событий.*

1. На экране появляется основная форма программы. Пользователь вводит критерии поиска марок в каталоге и нажимает кнопку «Поиск».
2. В главном окне приложения отображается список найденных марок.
3. Пользователь кликом мыши выбирает нужную марку в списке.
4. В главном окне приложения отображается информация о выбранной марке.

*Постусловия.* Отображена информация о марке.

***Вариант использования:* Сохранить отчет в файл.**

*Краткое описание.* Дает возможность пользователю сохранить информацию о марках в каталоге. Пользователь открывает окно сохранения выбрав команду «Создать алгоритм» в меню «Каталог» или нажав на аналогичную кнопку в главном окне приложения.

*Актант.* Пользователь.

*Предусловия.* Компьютер пользователя включён, на экране – главное окно программы «Автоматизированная система учета почтовых марок».

*Основной поток событий.*

1. Пользователь выбрал в меню «Каталог» пункт «Создать отчет».
2. На экране появляется стандартное окно для сохранения файла. В нем пользователь выбирает папку для сохранения файла, вводит имя файла вручную и нажимает кнопку «Сохранить». Файл сохраняется. В случае, если файл с таким именем уже существует, выдается сообщение о перезаписи, файл либо сохраняется, либо пользователь возвращается в окно сохранения файла. Вариант использования завершается успешно.

А1: Нажата кнопка «Отмена»

*Альтернативы.*

А1: Нажата кнопка «Отмена»

А1.1: Система закрывает окно сохранения и возвращается в главную форму приложения.

*Постусловия.* Отчет сохранен в файл.

***Вариант использования:* Создать новую марку в каталоге.**

*Краткое описание.* Дает возможность пользователю создать новую запись в о марке в системе. Пользователь вводит параметры марки в диалоговом окне. После создания пользователь может работать с новой записью.

*Актант.* Пользователь.

*Предусловия.* Компьютер пользователя включён, на экране – главное окно программы «Автоматизированная система учета почтовых марок.

*Основной поток событий.*

1. Пользователь нажимает на кнопку «Добавить марку».
2. На экране появляется диалоговое окно с полями ввода для параметров добавляемой марки.
3. Пользователь вводит значения, соответствующие характеристикам марки, в поля ввода. После ввода информации пользователь нажимает на кнопку «Сохранить»

А1: Заполнены не все обязательные поля.

А2: Нажата кнопка «Отмена»

1. Система закрывает окно создания и возвращается в главную форму приложения.

*Альтернативы.*

А1: Заполнены не все обязательные поля.

А1.1: Система выводит сообщение «Некорректные данные», содержащее информацию, какие конкретно обязательные поля не были заполнены. При закрытии сообщения при пользователь возвращается в диалоговое окно. Вариант использования завершается успешно.

А2: Введены значения, не подходящие по ограничениям.

А2.1: Система выводит сообщение «Некорректные данные», содержащее информацию, какие поля были заполнены некорректно, а так же ограничения для них. При закрытии сообщения при пользователь возвращается в диалоговое окно. Вариант использования завершается успешно.

*Постусловия.* Запись о марке добавлена в систему.

### Диаграмма классов

Диаграммы классов – это наиболее часто используемый тип диаграмм, которые создаются при моделировании объектно-ориентированных систем, они показывают набор классов, интерфейсов и коопераций, а также их связи. На практике диаграммы классов применяют для моделирования статического представления системы, они служат основой для целой группы взаимосвязанных диаграмм – диаграмм компонентов и диаграмм размещения.

На рисунке 12 представлены диаграммы классов модулей приложения.

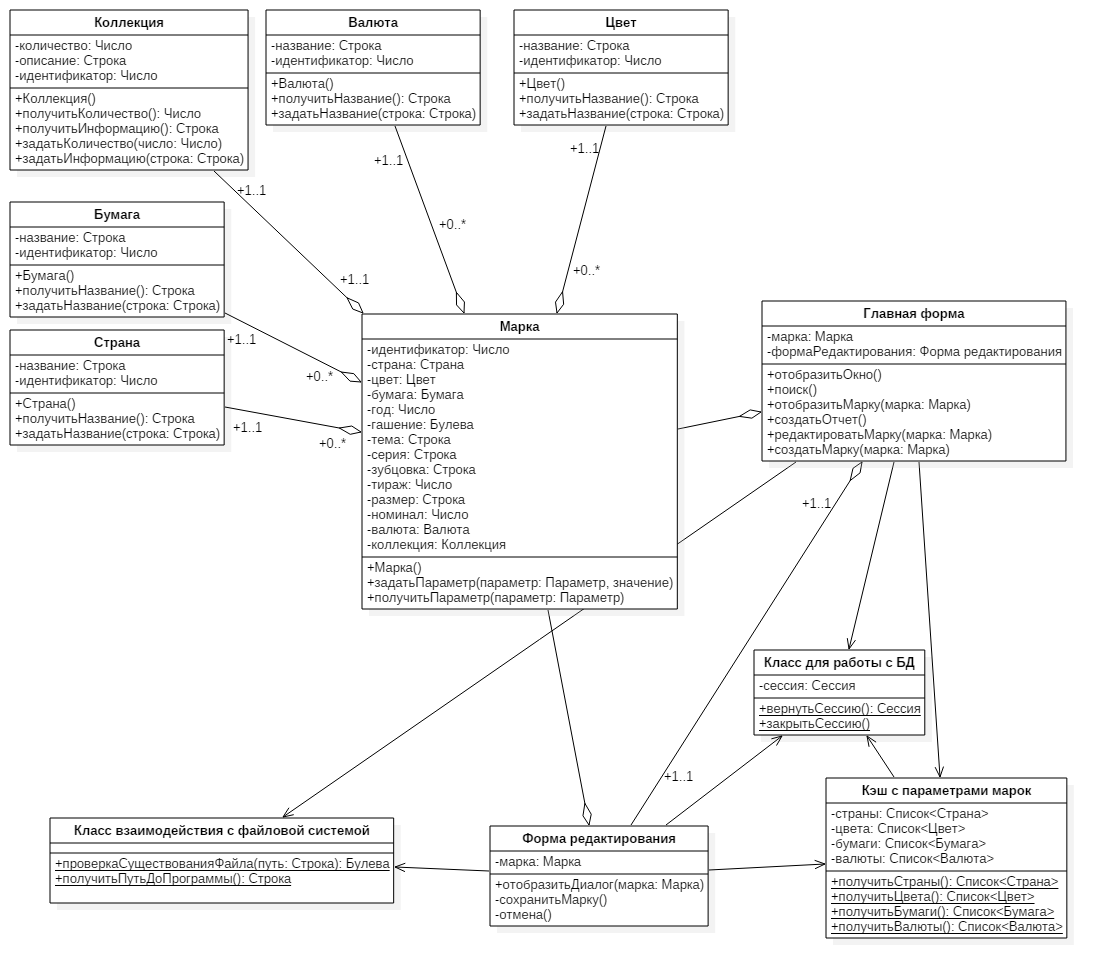


Рисунок 12 – Диаграмма классов приложения

### Диаграмма состояний

Главное назначение диаграммы состояний – описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение моделируемой системы в течение всего ее жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий. Системы, которые реагируют на внешние действия от других систем или от пользователей, иногда называют реактивными. Если такие действия инициируются в произвольные случайные моменты времени, то говорят об асинхронном поведении модели. Диаграмма состояний системы представлена на рисунке 13.

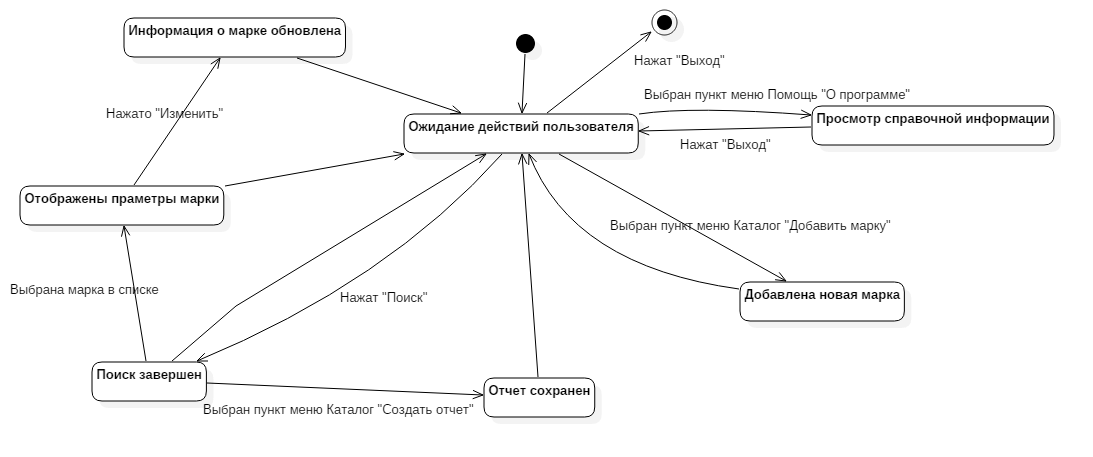


Рисунок 13 – Диаграмма состояний системы

### Диаграмма последовательностей

Для моделирования взаимодействия объектов в языке UML используются соответствующие диаграммы взаимодействия. Так, взаимодействия объектов можно рассматривать во времени, и тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

На диаграмме последовательности изображаются объекты, которые непосредственно участвуют во взаимодействии. Для диаграммы последовательности ключевым моментом является динамика взаимодействия объектов во времени. Взаимодействия объектов реализуются посредством сообщений, которые посылаются одним объектом другим и также образуют порядок по времени своего возникновения.

На рисунке 14 представлена диаграмма последовательности варианта использования «Создать запись о марке». На главной форме пользователь наживает кнопку «Добавить марку». Система отображает новое диалоговое окно, содержащее поля ввода. Пользователь задает значения параметров и нажимает кнопку «Сохранить». Происходит проверка корректности введенных значений. Если данные некорректны, то формируется форма ошибки и выводится сообщение об ошибке. Если значения параметров заданы корректно, то происходит создание записи о марке.

На рисунке 15 представлена диаграмма последовательности варианта использования «Добавить марку в коллекцию». На главной форме пользователь наживает кнопку «В коллекции». Система отображает поля ввода. Пользователь задает значения параметров и нажимает кнопку «Сохранить». Происходит проверка корректности введенных значений. Если данные некорректны, то формируется форма ошибки и выводится сообщение об ошибке. Если значения параметров заданы корректно, то происходит создание записи о марке.

На рисунке 16 представлена диаграмма последовательности варианта использования «Создать отчёт». На главной форме пользователь нажимает кнопку «Создать отчет», после чего отрывается стандартная форма сохранения файла. Пользователь набирает имя файла вручную. В случае, если файл с таким именем уже существует, выдается сообщение о перезаписи, файл либо сохраняется, либо пользователь возвращается в окно сохранения файла. Иначе, файл сохраняется.

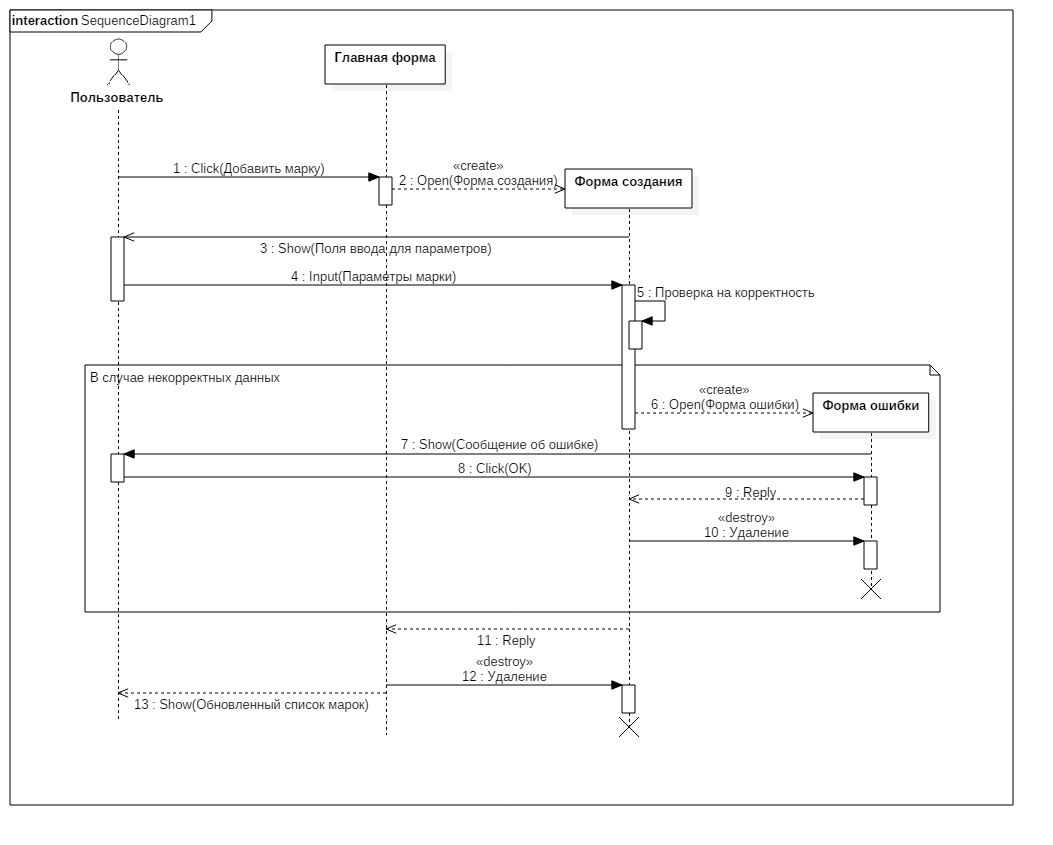


Рисунок 14 – Диаграмма последовательности варианта использования   
«Создать запись о марке»

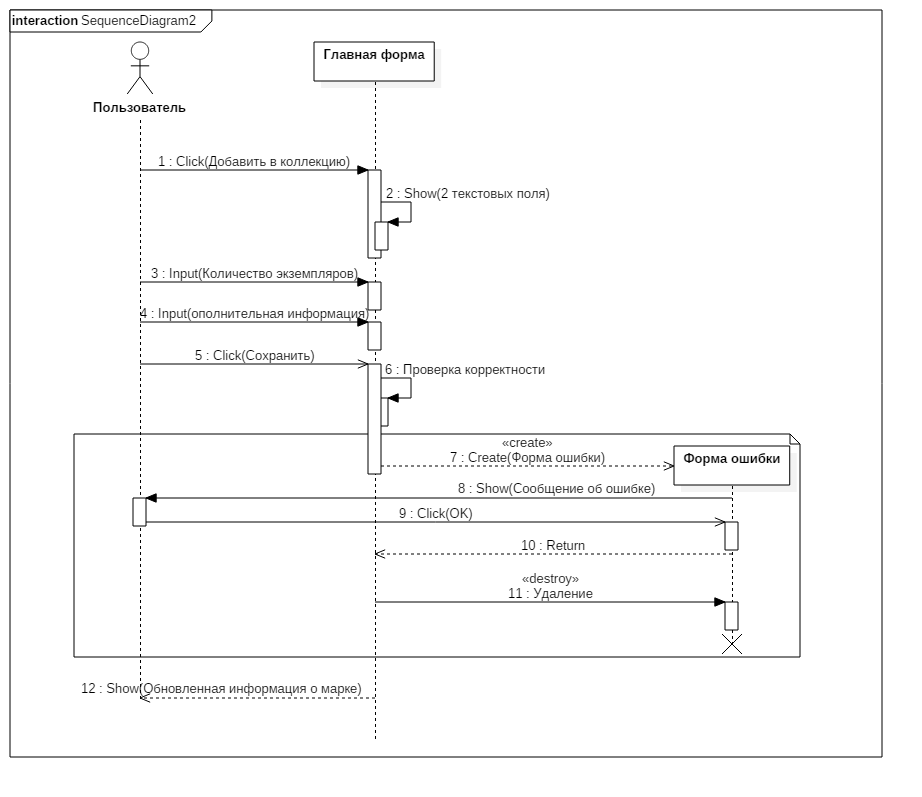


Рисунок 15 – Диаграмма последовательности варианта использования   
«Добавить марку в коллекцию»

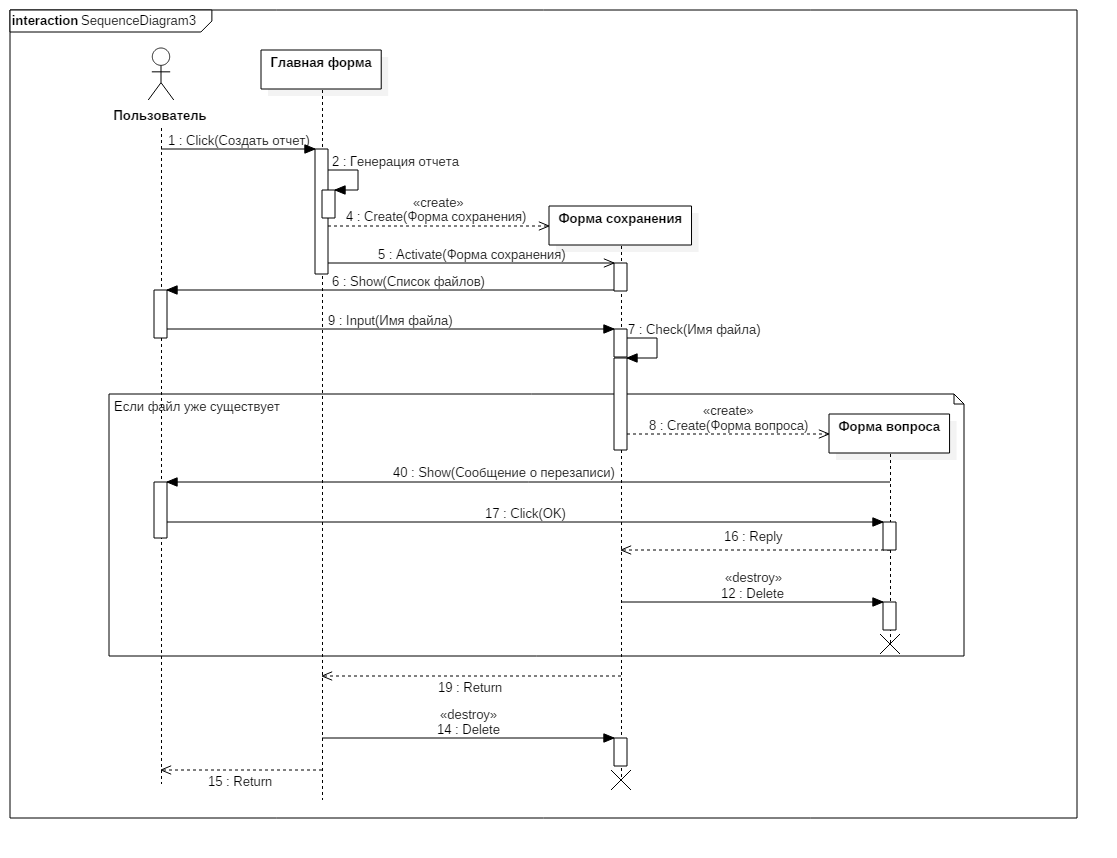


Рисунок 16 – Диаграмма последовательности варианта использования   
«Создать отчёт»

Диаграмма деятельности

Диаграммы деятельности используются для моделирования процесса выполнения операций. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, а переход в следующее состояние срабатывает только при завершении этой операции в предыдущем состоянии. На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фокусируется на результате деятельности. Сам же результат может привести к изменению состояния системы или возвращению некоторого значения. На рисунке 16 представлена диаграмма деятельности системы.

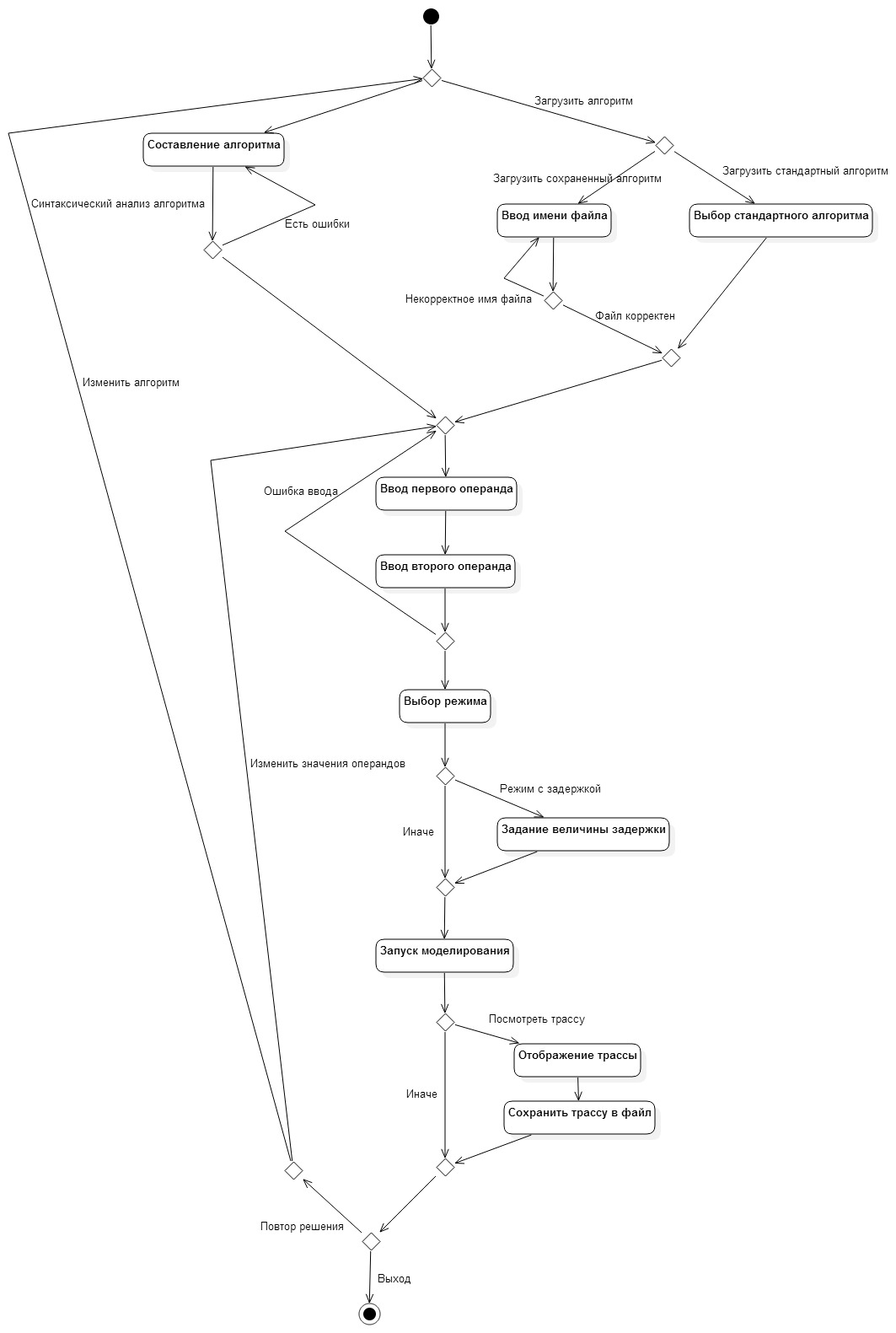


Рисунок 16 – Диаграмма деятельности системы