Введение

Почтовая марка – это специальный знак почтовой оплаты, выпускаемый и продаваемый национальными (и иными) почтовыми ведомствами и обладающий определённой номинальной стоимостью (номиналом).

Впервые необходимость в оплате почтовых услуг появилась с появлением самой почты, при этом почта, как и в настоящее время, представляла собой передачу информации в письменном виде от отправителя к получателю. До середины 17 века монополия на почту принадлежала государствам, имеющим необходимость получать как можно раньше важную информацию. В 1653 году в Париже появилась регулярная почтовая служба. Поскольку оплачивать отправку письма мог как отправитель, так и получатель, то письмо обертывалось билетами почтового сбора, и для оплаты на билете проставлялся специальный штемпель. К началу 19 века механизм оплаты отправления писем эволюционировал до оттиска на специальном штемпельном листе. Этот лист мог иметь различную форму (круглую, овальную, восьмиугольную), а разнообразные водяные знаки препятствовали подделкам. В 1840 году идея штемпельного листа трансформировалась в традиционную марку, на которой с одной стороны изображен сложный для подделки рисунок, а с другой имелось покрытие из клеевого слоя. Первая почтовая марка вошла в историю под названием «Черный пенни». С этого момента для оплаты почтовых услуг следовало приобрести марку, наклеить ее на конверт, проставить печать и отправить письмо адресату [1].

В настоящее время роль почты при отправлении писем заметно уменьшилась с развитием Интернета. Пользователь может отправить электронное письмо, содержащее информацию, всего за несколько секунд, при этом не требуется оплачивать отправку письма. Таким образом, роль почтовых марок большей частью сводится к объекту коллекционирования. Стоимость отдельных марок, существующих в мире в единственном экземпляре, может превышать миллион долларов США.

Во время курсового проектирования необходимо разработать автоматизированную систему учета почтовых марок, которая позволит пользователю вести собственный каталог марок и получать информацию в виде отчета.

При проектировании системы будет использоваться методология UML (Unified Modeling Language – унифицированный язык моделирования), которая применяется 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 на основании UML-моделей возможна генерация кода [2].

1. Описание и анализ предметной области

1.1 Основные характеристики почтовой марки

Почтовая марка – это специальный знак почтовой оплаты, выпускаемый и продаваемый национальными (и иными) почтовыми ведомствами и обладающий определённой номинальной стоимостью (номиналом).

Каждая марка имеет следующие обязательные параметры:

* Страна-изготовитель. Параметр содержит название страны, в которой была изготовлена марка (см. рисунки 1, 2, 4);
* Год выпуска. Параметр содержит год, в который марка была выпущена (см. рисунок 1);
* Гашение. Параметр содержит информацию о том, было ли произведено гашение марки, или нет (см. рисунок 1);



Рисунок 1 – Немецкая почтовая марка, выпущенная в 1965 году

* Тема. Параметр содержит информацию о тематике изображения марки (см. рисунок 1);
* Серия. Параметр содержит описание общей темы, события или повода выпуска марок, имеющих похожий стиль оформления (см. рисунок 2);
* Номинал. Параметр содержит цену марки, с помощью которой была произведена оплата почтовых услуг (см. рисунки 1, 2);
* Тираж. Параметр содержит данные о количестве марок, выпущенных одной партией (как правило, на марке нет сведений о тираже);
* Зубцовка. Данный параметр, в свою очередь, разделяется на три независимых признака:

1. технология перфорирования (см. рисунок 3);
2. количество зубцов на каждой стороне марки;
3. направление линий зубцовки (см. рисунок 4);



Рисунок 2 – «Романовская» серия марок, 1913 год

* Бумага. Параметр содержит информацию об основных характеристиках бумаги, на которой печатался тираж марки;
* Размер. Разные марки могут быть разного размера;
* Цвет. Дешевые марки могут быть напечатаны в черно-белом варианте, тогда как более дорогие используют большее число различных цветов.

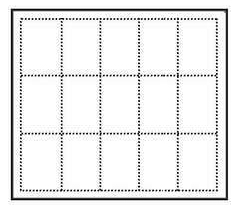


Рисунок 3 – Схема марочного листа с рамочной зубцовкой



Рисунок 4 – Частичная зубцовка на марке США 1954 года

В качестве примера можно рассмотреть образец почтовой марки, выпущенной в 1914 году (см. рисунок 5). На данной марке изображен Георгий Победоносец, убивающий дракона, надпись под изображением гласит: «Въ пользу воиновъ и ихъ семействъ». Поверх изображения сделано гашение с пометкой «Образец». Серия отсутствует. Номинал марки составляет 10 копеек, хотя ее цена на момент выпуска составляла 11 копеек. Изображение является трехцветным, не считая черного цвета гашения. Из-за отсутствия дополнительной информации (размер, бумага, тираж) некоторые параметры не могут быть корректно идентифицированы.



Рисунок 5 – Образец почтовой марки Российской империи 1914 года

1.2 Обзор существующих систем-аналогов

1.2.1 Программный комплекс «Моя филателия»

На сегодняшний день наиболее известной системой подобного рода является программный комплекс «Моя филателия» (см. рисунок 6). С ее помощью пользователь может создавать и редактировать коллекции марок, вести справочники по различным характеристикам марок, таким как базовые данные (страна, год выпуска), внешние характеристики (форма, цвет, размер), а также содержание (оформление, тема). Благодаря данным справочникам пользователь может добавлять в коллекцию исчерпывающие записи о марках, которые содержат не только подробные характеристики марки, но и информацию о возможных продавцах или покупателях данной марки. Кроме того, для удобства пользователя реализована функция поиска нужной марки по запросу, содержащему некоторые данные о марке [3].

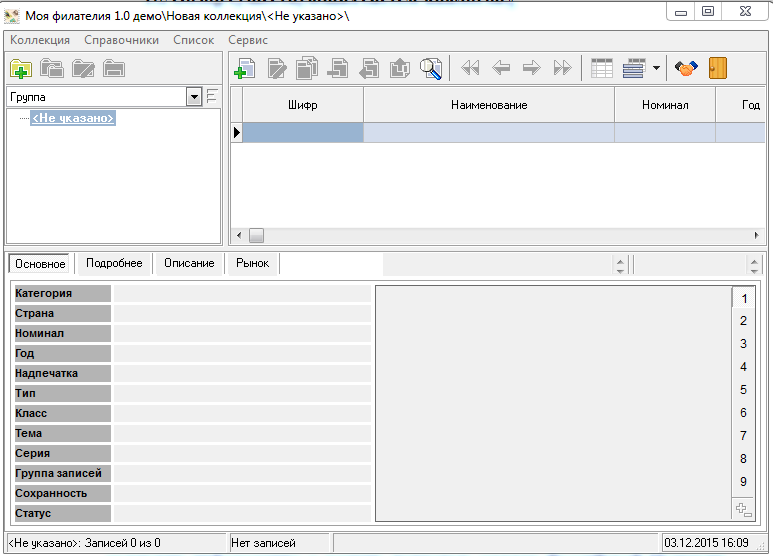


Рисунок 6 – Экранная форма программного комплекса «Моя филателия»

1.2.2 Филателия. Почтовые марки России и СССР

В качестве систем-аналогов также можно рассмотреть сайты, на которых размещена информация о марках, доступных для покупки. Это связано с тем, что люди, коллекционирующие марки, интересуются как характеристиками своих марок, так и возможностями приобрести новые, продать лишние или совершить обмен. Сайты, посвященные торговым маркам, содержат необходимую информацию и обеспечивают надежный доступ к ней. Кроме того, популярность сайта зависит от своевременного обновления новостной ленты, содержащей оригинальный контент.

Сайт, расположенный по адресу [4] (рисунок 7), содержит информацию о редких почтовых марках Российской Империи, РСФСР и СССР. Эти марки сгруппированы как по годам выпуска, так и по общей теме изображения.

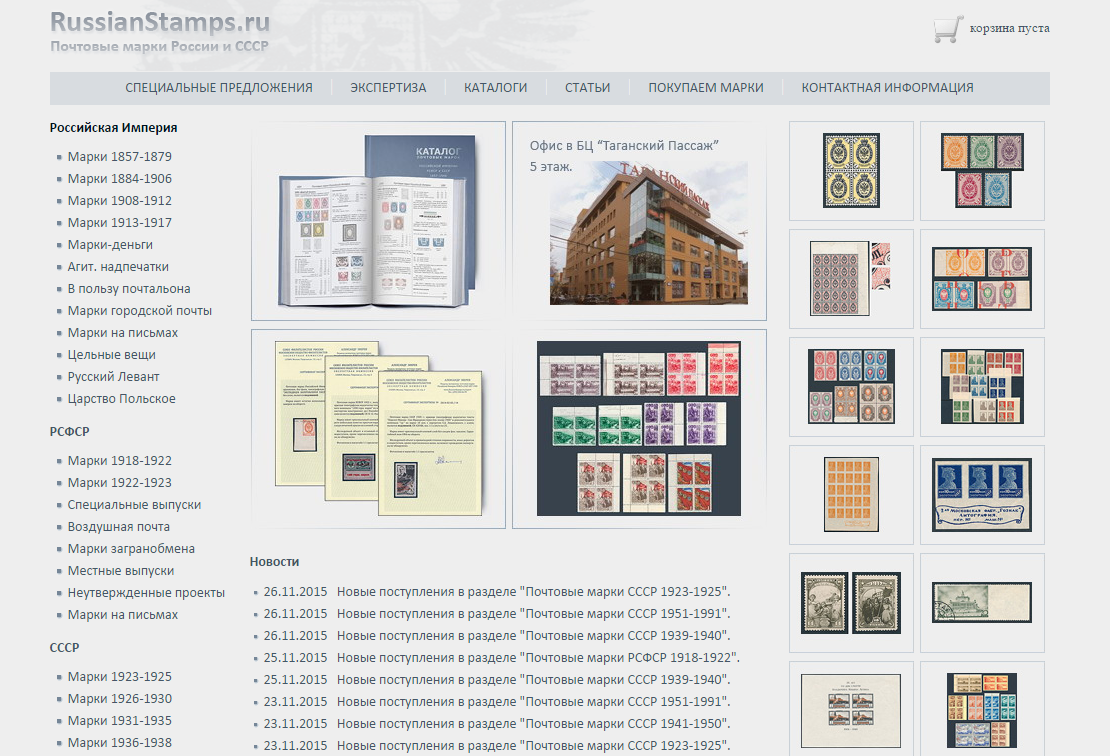


Рисунок 7 – Главная форма сайта «Филателия.  
Почтовые марки России и СССР»

В разделе «Специальные предложения» можно приобрести марки комплектами по более низкой цене. Раздел «Экспертиза» определяет понятие экспертизы марки, ее условия, сроки и стоимость проведения. Раздел «Каталоги» предлагает посетителю сайта приобрести различные каталоги марок. Раздел «Покупаем марки» позволяет посетителю продать марки магазину или отдать на реализацию на комиссионной основе. Посетитель также имеет возможность найти нужную марку по ее году выпуска и купить у лиц, указанных в разделе «Контактная информация». Благодаря своевременным обновлениям сайта посетители имеют доступ к информации о новых марках, появившихся в продаже.

1.2 Диаграмма объектов предметной области

Диаграмма объектов представляет статическую составляющую взаимодействующих между собой объектов, она должна включать в себя только те объекты предметной области, которые потом преобразуются в диаграмму классов.

Связи между объектами показывают отношения между ними, при необходимости в диаграмме можно привести и атрибуты (свойства) объектов. На рисунке 8 приведена диаграмма объектов для рассматриваемой системы.

Пользователь составляет коллекцию, состоящую из некоторого числа всех возможных марок. Эта коллекция хранится в базе данных. Имея коллекцию, пользователь может создать отчет по ней, который будет храниться в файле заданного формата.

1.4 Постановка задачи

В рамках курсового проектирования перед авторами поставлена задача – разработать автоматизированную систему учета почтовых марок, которая позволит пользователю вести свою коллекцию почтовых марок,

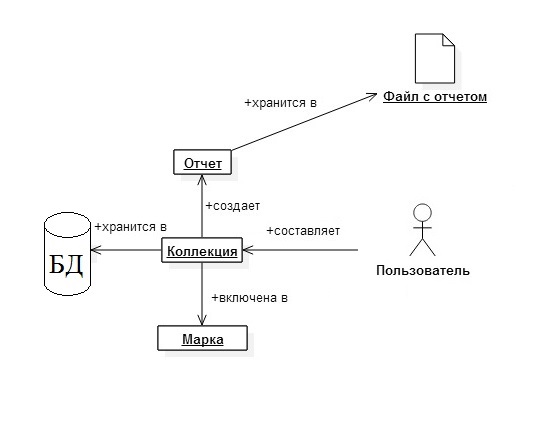


Рисунок 8 – Диаграмма объектов предметной области

редактировать ее и производить поиск записей коллекции по заданному критерию.

В системе должны быть предусмотрены возможности управления каталогом, содержащим марки. Эти марки будут различаться по названию страны-изготовителя, номиналу, году выпуска, типу бумаги и иным параметрам, указанным в п.2 приложения к заданию на курсовой проект. Кроме того, к марке можно будет прикрепить фотографию. Некоторые параметры, такие как цвет марки или ее валюта, можно будет задать, используя информацию из справочников. При этом записи справочников также можно редактировать, удалять или добавлять новые. Корректно задав параметры марки, пользователь сможет создать новую марку в каталоге. Марки каталога могут быть добавлены в коллекцию пользователя или быть удалены из нее. Как справочники, так и каталог должны удовлетворять принципам ведения баз данных. Пользователь также сможет найти требуемые марки, настроив критерии поиска, в качестве которых выступают параметры марки. При необходимости пользователь сможет сохранить информацию о марках, соответствующих выбранным критериям, в виде отчета в файл, настроив предварительно формат файла.

В системе также должна быть обеспечена возможность получения справочной информации как о самой системе, так и о предоставляемых ею возможностях.

Таким образом, система должна выполнять следующие функции:

1. работа со справочниками БД:

* добавление записи;
* удаление записи;
* редактирование записи;

1. ввод информации о почтовой марке;

* добавление описания;
* задание параметров марки;
* прикрепление фотографии;

1. редактирование информации о марках;
2. удаление информации о марках;
3. формирование отчета:
   1. настройка параметров отчета;
   2. импортирование отчета в файл заданного формата;
4. отбор информации по заданному критерию (параметру марки);
5. выбор критериев отбора;
6. выдача справочной информации о системе.
7. ПРОЕКТИРОВАНИЕ СИСТЕМЫ
8. Структурная схема системы

Система (греч. – «составленное из частей», «соединение» от «соединяю») – множество элементов, находящихся в отношениях и связях друг с другом, которое образует определённую целостность, единство.

Как следует из определения, отличительным (главным свойством) системы является ее целостность: комплекс объектов, рассматриваемых в качестве системы, должен обладать общими свойствами и поведением. Очевидно, необходимо рассматривать и связи системы с внешней средой. В самом общем случае понятие «система» характеризуется:

* наличием множества элементов;
* наличием связей между ними;
* целостным характером данного устройства или процесса.

Система должна представлять собой совокупность элементов (объектов, субъектов), находящихся между собой в определенной зависимости и составляющих некоторое единство (целостность), направленное на достижение определенной цели. Система может являться элементом другой системы более высокого порядка (надсистема) и включать в себя системы более низкого порядка (подсистемы). То есть систему можно рассматривать как набор подсистем, организованных для достижения определенной цели и описанных с помощью набора моделей (возможно, с различных точек зрения), а подсистему – как группу элементов, часть которых составляет спецификацию поведения, представленного другими ее составляющими.

На рисунке 9 представлена структурная схема системы. В состав системы входят следующие подсистемы:

1. подсистема работы со справочниками, которая позволяет добавлять, удалять и редактировать записи;
2. подсистема работы с марками, в состав которой входят:

Подсистема работы с алгоритмами

Подсистема моделирования  
работы МТ




Рисунок 9 – Структурная схема системы

1. подсистема работы с коллекцией, которая позволяет добавлять марки из каталога в коллекцию, а также задавать дополнительные параметры.
2. подсистема поиска, позволяющая пользователю найти нужную информацию по введенным критериям (параметрам марки);
3. подсистема формирования отчета, которая позволяет создать отчет, содержащий информацию о марках, соответствующих введенным критериям поиска;
4. файловая подсистема, которая отвечает за сохранение отчета и изображений марок в файлах заранее определенного типа;
5. справочная подсистема, которая содержит сведения о системе (руководство пользователя) и о её разработчиках.
6. Разработка прототипа 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.

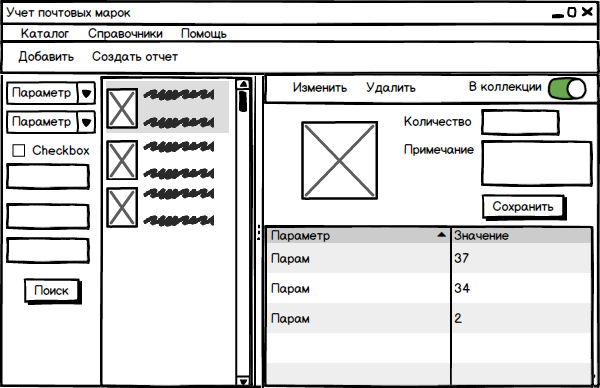


Рисунок 10 – Прототип главного окна

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 "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.

После активации переключателя «В коллекции» пользователь может изменить параметры выбранной марки, доступные только для коллекции. После нажатии на кнопку «Сохранить» введенные данные сохраняются.

При нажатии на кнопки «Добавить» и «Изменить» открывается окно, прототип которого представлен на рисунке 11. В данном окне пользователь может вводить значения параметров марки. При нажатии на кнопку «Сохранить» в случае корректных данных создается запись о марке. При нажатии на кнопку «Отмена» диалоговое окно закрывается, а введенные данные не сохраняются.

В пункте меню "Справочники" пользователь может выбрать команду открытия окна управления справочником одного из параметров марки. Прототип диалогового окна изображен на рисунке 12. Нажимая на соответствующие кнопки, пользователь может создавать, редактировать и удалять записи в выбранном справочнике. В случаях добавления и изменения записей, открывается дополнительное диалоговое окно, где пользователь может ввести значение записи.

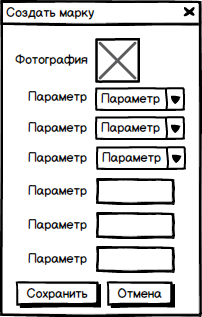


Рисунок 11 – Прототип окна создания и редактирования марки

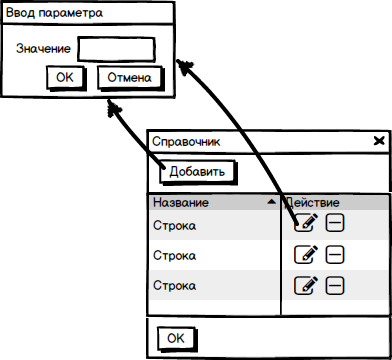


Рисунок 12 – Прототип окна управления справочником

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.

Общая диаграмма вариантов использования разработанной системы представлена на рисунке 13. На данной диаграмме актант – пользователь, который может управлять каталогом марок и собственной коллекцией марок, добавляя и убирая марки из нее, храня дополнительную информацию. Также пользователь может управлять справочниками характеристик марок, вести поиск и создавать отчеты с учетом фильтров. Диаграмма вариантов использования при работе с марками изображена на рисунке 14. Пользователь может создавать, редактировать и просматривать записи о марках, а также удалять их из каталога.

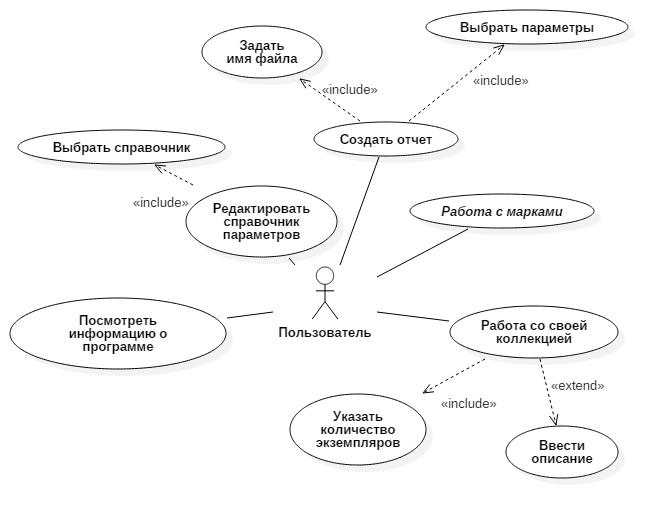


Рисунок 13 – Общая диаграмма вариантов использования системы

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.

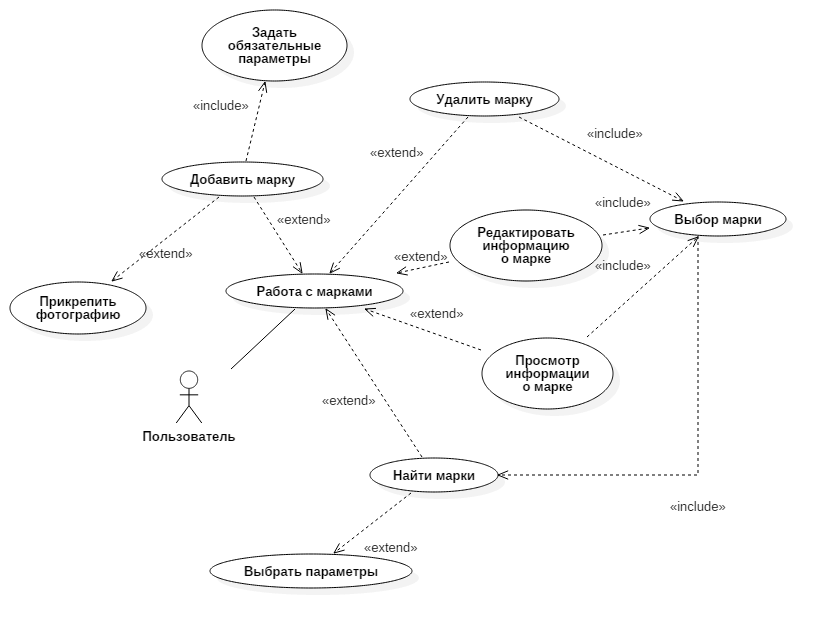


Рисунок 14 –Диаграмма вариантов использования системы   
(работа с марками)

***Вариант использования:* Добавить марку в каталог.**

*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. Главная форма приложения содержит меню программы, панель поиска, панель вывода результатов поиска и панель, в которой отображается подробная информация о выделенной марке.
2. Пользователь нажимает на кнопку «Добавить марку» или выбрать пункт «Добавить» в меню «Каталог».
3. На экране появляется диалоговое окно «Создать марку» с полями ввода для параметров добавляемой марки (см. рисунок 11).
4. Пользователь вводит значения, соответствующие характеристикам марки, в поля ввода.
5. Пользователь нажимает на кнопку «Сохранить».

А1: Заполнены не все обязательные поля.

А2: Нажата кнопка «Отмена» или кнопка «Х».

1. Система закрывает окно создания новой марки и возвращается в главную форму приложения. Вариант использования завершается успешно.

*Альтернативы.*

А1: Заполнены не все обязательные поля.

А1.1: Система выводит в отдельном окне сообщение «Некорректные данные», содержащее информацию, какие конкретно обязательные поля не были заполнены.

A1.2: Пользователь нажимает на кнопку «OK». Окно закрывается, переход к пункту 4 основного потока событий.

А2: Введены значения, не подходящие по ограничениям.

А2.1: Система выводит в отдельном сообщение «Некорректные данные», содержащее информацию, какие поля были заполнены некорректно, а так же ограничения для них.

A2.2: Пользователь нажимает на кнопку «OK». Окно закрывается, переход к пункту 4 основного потока событий.

*Постусловия.* Запись о марке добавлена в базу данных.

***Вариант использования:* Добавить марку в коллекцию.**

*Краткое описание.* Дает возможность пользователю добавить марку из каталога в коллекцию.

*Актант.* Пользователь.

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

*Основной поток событий.*

1. Главная форма приложения содержит меню программы, панель поиска, в центральной панели выделена одна марка из числа соответствующих заданным критериям поиска, в информационной панели марки, расположенной в правой части главного окна, отображается информация о выделенной марке.
2. Пользователь нажимает на кнопку «В коллекции».
3. В информационной панели отображаются поля «Количество» и «Примечание», где пользователь может задать соответствующие значения данных полей.
4. Пользователь вводит значения, соответствующие характеристикам марки, в поля ввода.
5. Пользователь нажимает на кнопку «Сохранить».

А1: Не заполнено поле «Количество».

1. Система сохраняет информацию о введенных пользователем данных. Вариант использования завершается успешно.

*Альтернативы.*

А1: Не заполнено поле «Количество».

А1.1: Система выводит в отдельном окне сообщение «Некорректные данные», содержащее информацию о том, что поле «Количество» было заполнено неверно.

A1.2: Пользователь нажимает на кнопку «OK». Окно закрывается, переход к пункту 4 основного потока событий.

*Постусловия.* Выделенная марка добавляется в коллекцию пользователя.

### 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 15 представлена диаграмма сущностных классов приложения.

Класс «Марка» содержит информацию о марке – ее значения параметров, такие как зубцовка, тираж или размер.

Классы «Страна», «Валюта», «Цвет», «Бумага» содержат информацию о всех возможных значениях соответствующих параметров марки.

Класс «Коллекция» содержит информацию о количестве экземпляров марки и описание марки. Одной марке соответствует один объект класса «Коллекция».

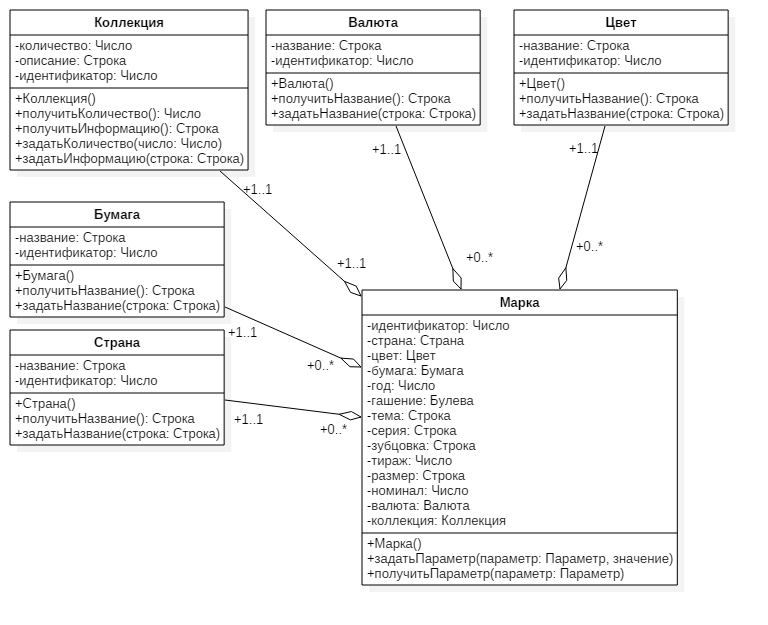


Рисунок 15 – Диаграмма сущностных классов приложения

### 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.

Общая диаграмма состояний системы представлена на рисунке 16.

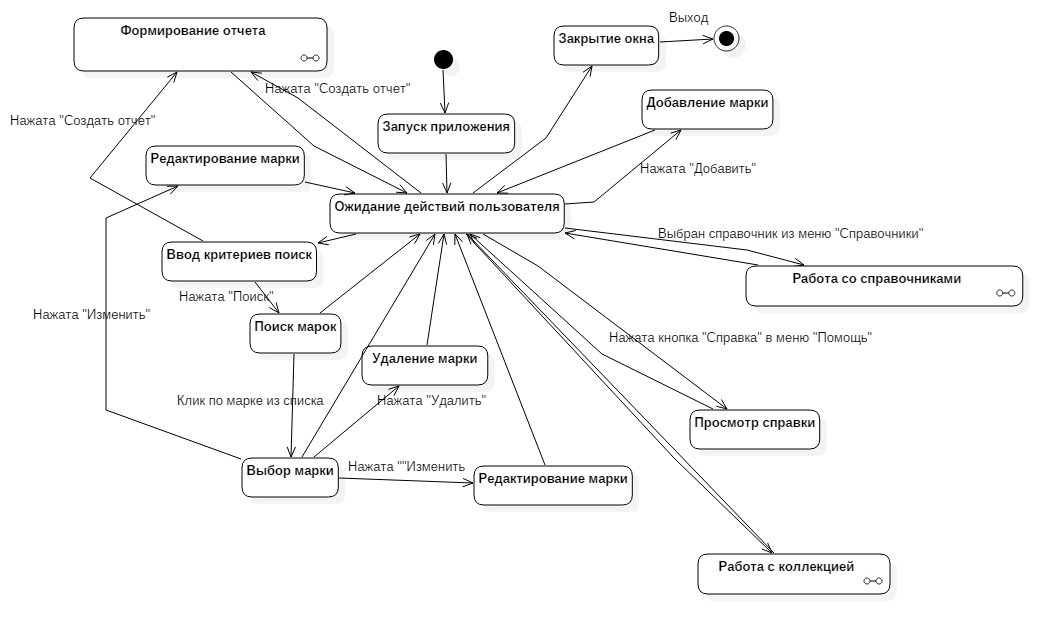


Рисунок 16 – Общая диаграмма состояний системы

После запуска приложения отображается главное окно программы. При нажатии на кнопку «Добавить» открывается окно создания марки, при нажатии на кнопку «Справка» отображается справочная информация о системе, при нажатии на кнопку «Поиск» осуществляется поиск марок по введенным критериям поиска, при нажатии на марку из списка отображается список её параметров. При нажатии на кнопку «Изменить» открывается окно редактирования параметров марки, при нажатии на кнопку «Удалить» происходит удаление марки. По нажатию на «Х» окно закрывается.

На рисунке 17 изображена диаграмма состояний «Работа со справочниками». При нажатии на кнопку «Добавить» открывается окно ввода нового значения. При нажатии на кнопку «Изменить» открывается окно ввода нового значения. При попытке сохранения значения происходит его проверка. Если данное значение уже присутствует в справочнике, второе такое же добавить нельзя. При нажатии на кнопку «Удалить» значение, если оно не используется, удаляется. При нажатии на «Х» форма управления справочником закрывается.

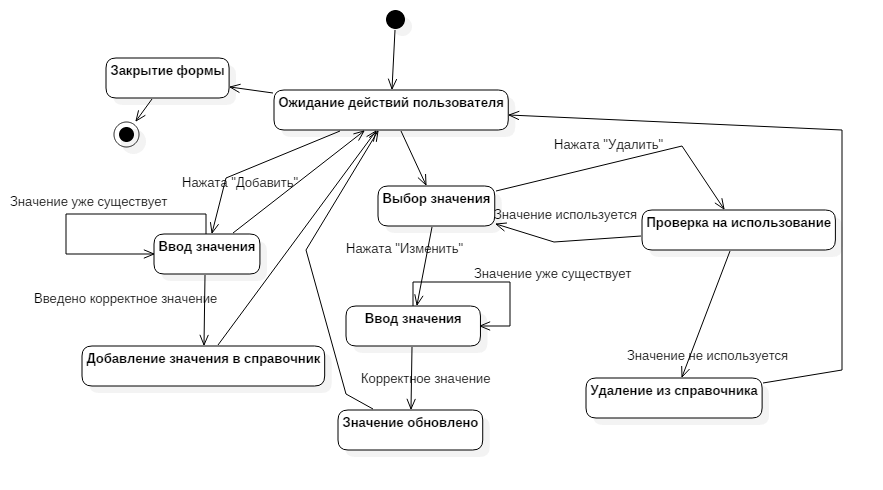


Рисунок 17 – Диаграмма состояний «Работа со справочниками»

### 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 18 представлена диаграмма последовательности варианта использования «Добавить марку в каталог». На главной форме пользователь наживает кнопку «Добавить». Система отображает новое диалоговое окно, содержащее поля ввода. Пользователь задает обязательные значения параметров и, возможно, прикрепляет фотографию, после чего 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 19 представлена диаграмма последовательности варианта использования «Добавить марку в коллекцию». На главной форме пользователь нажимает кнопку «В коллекции». Система отображает поля ввода количества экземпляров и примечания. 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 добавление марки в коллекцию.

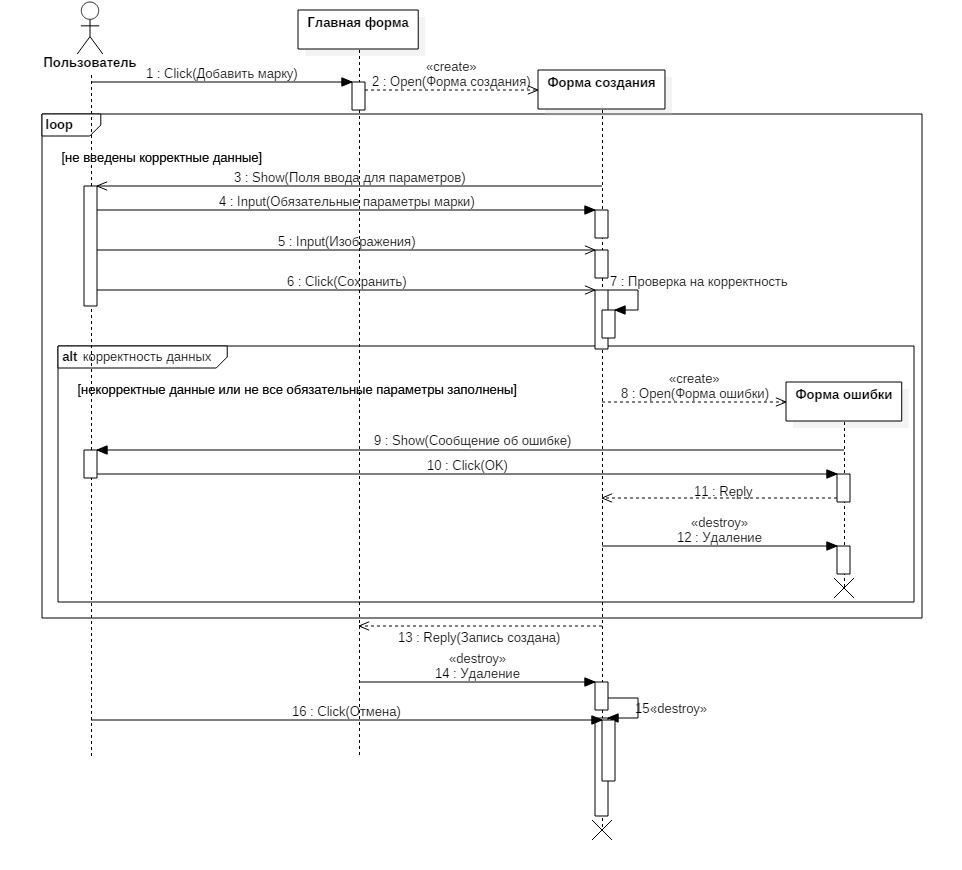


Рисунок 18 – Диаграмма последовательности варианта использования   
«Добавить марку в каталог»

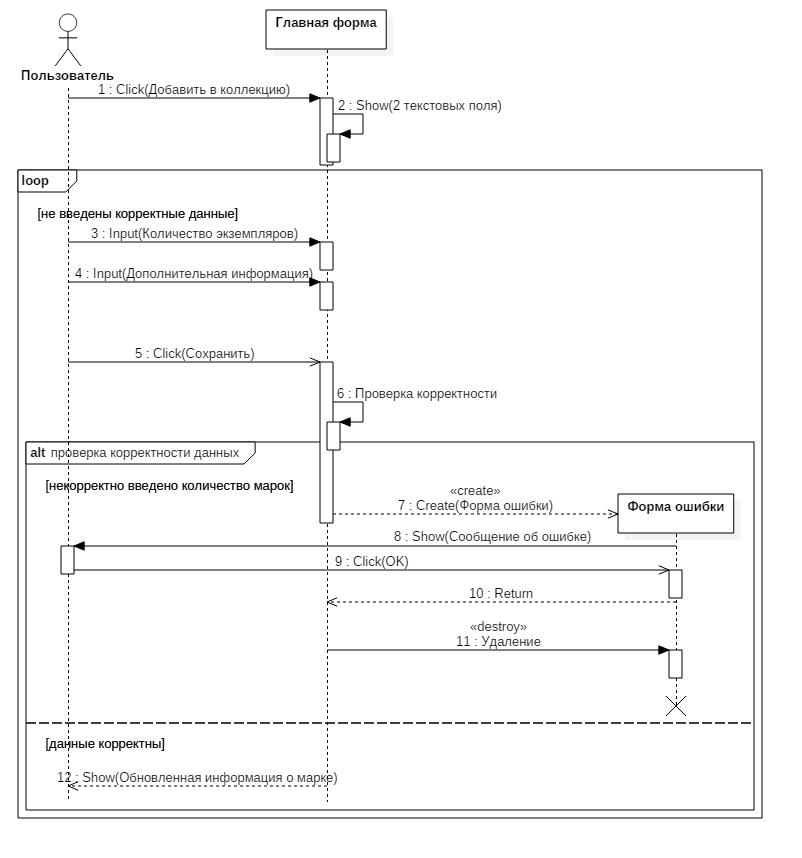


Рисунок 19 – Диаграмма последовательности варианта использования   
«Добавить марку в коллекцию»

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 20 представлена диаграмма деятельности «Работа с марками».

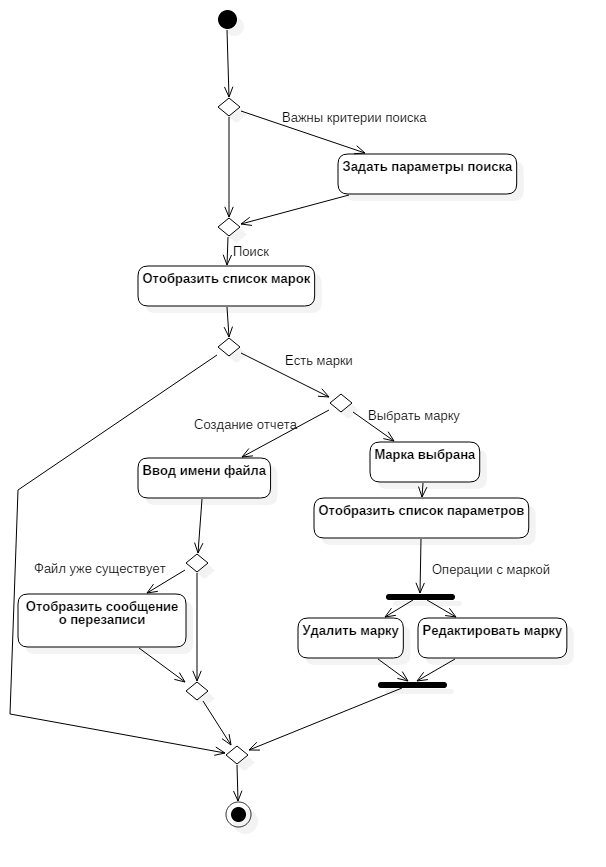


Рисунок 20 – Диаграмма деятельности «Работа с марками»

* 1. Логическая модель базы данных

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью.

Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но без привязки к конкретной СУБД. Более того, логическая модель данных необязательно должна быть выражена средствами именно реляционной модели данных. Основным средством разработки логической модели данных в настоящий момент являются различные варианты ER-диаграмм (Entity-Relationship, диаграммы сущность-связь).

На рисунке 21 изображена логическая модель базы данных системы.

Основная сущность – «Марка». Марка имеет набор различных характеристик, каждая из которых, кроме коллекции, имеет ровно одно значение. Если марки нет в коллекции пользователя, то её атрибут «коллекция» не содержит значения.

Сущность «Коллекция» хранит количество экземпляров конкретной марки в коллекции и примечание к ним. Каждая такая сущность может быть связана только с одной маркой.

Сущности «Бумага», «Валюта», «Страна», «Цвет» имеют только названия.

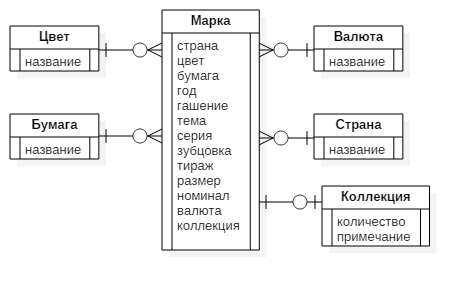


Рисунок 21 – Логическая модель базы данных системы

* 1. Выбор и обоснование комплекса программных средств
     1. Выбор языка программирования и среды разработки

Java – [объектно-ориентированный язык программирования](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), разработанный компанией [Sun Microsystems](https://ru.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) (в последующем приобретённой компанией [Oracle](https://ru.wikipedia.org/wiki/Oracle" \o "Oracle)). Дата официального выпуска – 23 мая 1995 года.

Программы на Java [транслируются](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80) в [байт-код](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4), выполняемый [виртуальной машиной Java](https://ru.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) – программой, обрабатывающей байтовый код и передающей инструкции оборудованию как [интерпретатор](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80). Достоинством подобного способа выполнения программ является полная независимость байт-кода от [операционной системы](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и [оборудования](https://ru.wikipedia.org/wiki/%D0%90%D0%BF%D0%BF%D0%B0%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0), что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Основные возможности Java:

* [автоматическое управление памятью](https://ru.wikipedia.org/wiki/%D0%A1%D0%B1%D0%BE%D1%80%D0%BA%D0%B0_%D0%BC%D1%83%D1%81%D0%BE%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5));
* расширенные возможности обработки исключительных ситуаций;
* богатый набор средств фильтрации ввода-вывода;
* набор стандартных коллекций: [массив](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2), [список](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)), [стек](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BA) и т. п.;
* наличие простых средств создания сетевых приложений (в том числе с использованием [протокола](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB) [RMI](https://ru.wikipedia.org/wiki/RMI));
* наличие классов, позволяющих выполнять [HTTP](https://ru.wikipedia.org/wiki/HTTP)-запросы и обрабатывать ответы;
* встроенные в язык средства создания многопоточных приложений, которые потом были портированы на многие языки (например, python);
* унифицированный доступ к [базам данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85):
  + на уровне отдельных [SQL](https://ru.wikipedia.org/wiki/SQL)-запросов — на основе [JDBC](https://ru.wikipedia.org/wiki/JDBC), [SQLJ](https://ru.wikipedia.org/wiki/SQLJ);
  + на уровне концепции объектов, обладающих способностью к хранению в базе данных — на основе [Java Data Objects](https://ru.wikipedia.org/w/index.php?title=Java_Data_Objects&action=edit&redlink=1) и [Java Persistence API](https://ru.wikipedia.org/wiki/Java_Persistence_API);
* поддержка обобщений;
* поддержка лямбд, замыканий, встроенные возможности функционального программирования;
* параллельное выполнение подпрограмм.

IntelliJ IDEA — [интегрированная среда разработки программного обеспечения](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8) на многих языках программирования, в частности [Java](https://ru.wikipedia.org/wiki/Java), [JavaScript](https://ru.wikipedia.org/wiki/JavaScript), [Python](https://ru.wikipedia.org/wiki/Python" \o "Python), разработанная компанией [JetBrains](https://ru.wikipedia.org/wiki/JetBrains" \o "JetBrains).

Некоторые преимущества IntelliJ IDEA:

* умное автодополнение, инструменты для анализа качества кода, удобная навигация, расширенные рефакторинги и форматирование для Java, Groovy, Scala, HTML, CSS, JavaScript, CoffeeScript, ActionScript, LESS, XML и многих других языков;
* поддержка всех популярных фреймворков и платформ, включая Java EE, Spring Framework, Grails, Play Framework, GWT, Struts, Node.js, AngularJS, Android, Flex, AIR Mobile и многих других;
* интеграция с серверами приложений, включая Tomcat, TomEE, GlassFish, JBoss, WebLogic, WebSphere, Geronimo, Resin, Jetty и Virgo;
* инструменты для работы с базами данных и SQL файлами, включая удобный клиент и редактор для схемы базы данных;
* интеграция с коммерческими системами управления версиями Perforce, Team Foundation Server, ClearCase, Visual SourceSafe;
* инструменты для запуска тестов и анализа покрытия кода, включая поддержку всех популярных фреймворков для тестирования.
  + 1. Выбор операционной системы

Были рассмотрены две наиболее популярные среди пользователей операционные системы: Windows и Mac OS.

Mac OS – это операционная система, разработанная компанией Apple, и в настоящее время является второй наиболее часто используемой ОС после Windows. У нее менее 20% доли рынка. Mac OS в отличие от Windows основана на Unix. Таким образом, Mac OS считается частью семейства Unix ОС, как и Linux. К основным плюсам можно отнести скорость, надежность и стабильность работы, отсутствие вирусов, удобность интерфейса, предварительный (быстрый) просмотр и большое количество полезных программ.

Windows – это одна из самых известных операционных систем, разработанных Microsoft. ОС Windows первоначально была основана на MS-DOS. Эти серии ОС Windows известны как 9-ые серии. Все последующие ОС Windows основаны на Windows NT. К основным плюсам относится большое количество дополнительных программ, множество игр и привычность, простота работы.

1. Реализация системы
2. Разработка и описание интерфейса пользователя

Для предоставления пользователю удобного доступа ко всем функциям приложения был разработан графический интерфейс, максимально подходящий для работы с системой. Главная форма приложения состоит из основного меню, левая часть формы содержит окно поиска, в центральной части отображаются марки, удовлетворяющие заданным критериям поиска, а в правой части отображается окно, содержащее подробные сведения о выделенной марке (см. рисунок 21).

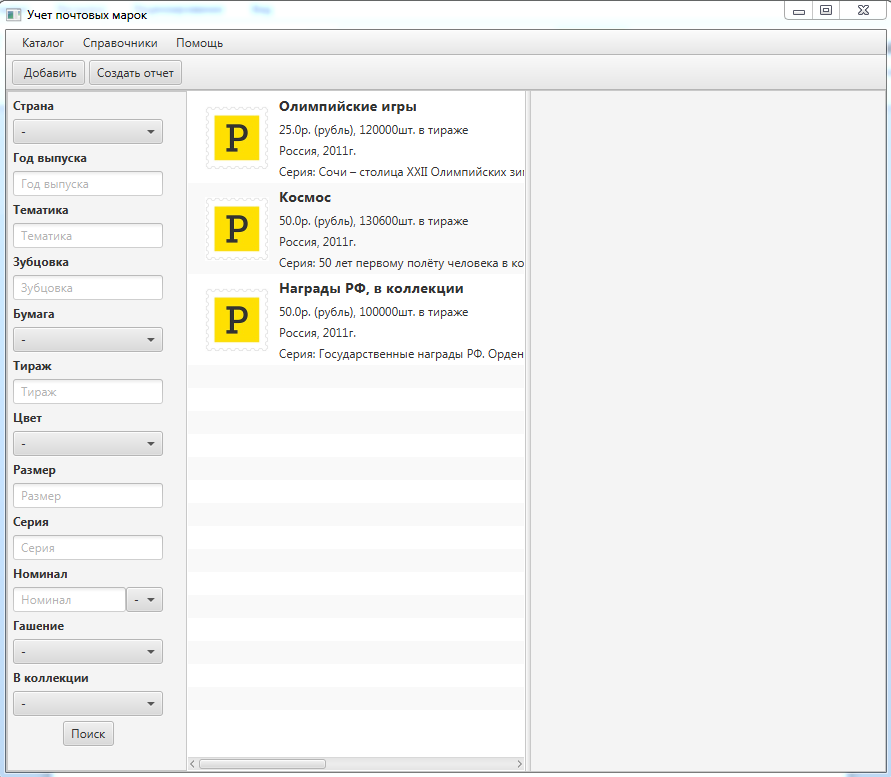


Рисунок 21 – Главная форма приложения

В основном меню содержатся необходимые функции для работы с марками. В пункте меню «Каталог» содержатся кнопки добавления марки и создания отчетов. Для удобства пользователя эти кнопки дублированы непосредственно на главной форме.

В пункте меню «Справочники», содержится информация о бумаге, цвете, типе валюты и стране-изготовителе марки. Пользователь может ознакомиться с содержимым справочников, редактировать, добавлять и удалять записи.

В пункте меню «Помощь» содержится справочная информация о разработчиках и о системе.

### Описание контрольного примера

В качестве контрольного примера рассмотрим создание новой марки, добавление ее в пользовательскую коллекцию, поиск марки по критериям поиска и печать выбранных по критериям поиска марок в файл.

Для создания новой марки нужно нажать кнопку «Добавить», после чего появится диалоговое окно, содержащее параметры марки (см. рисунок 22).

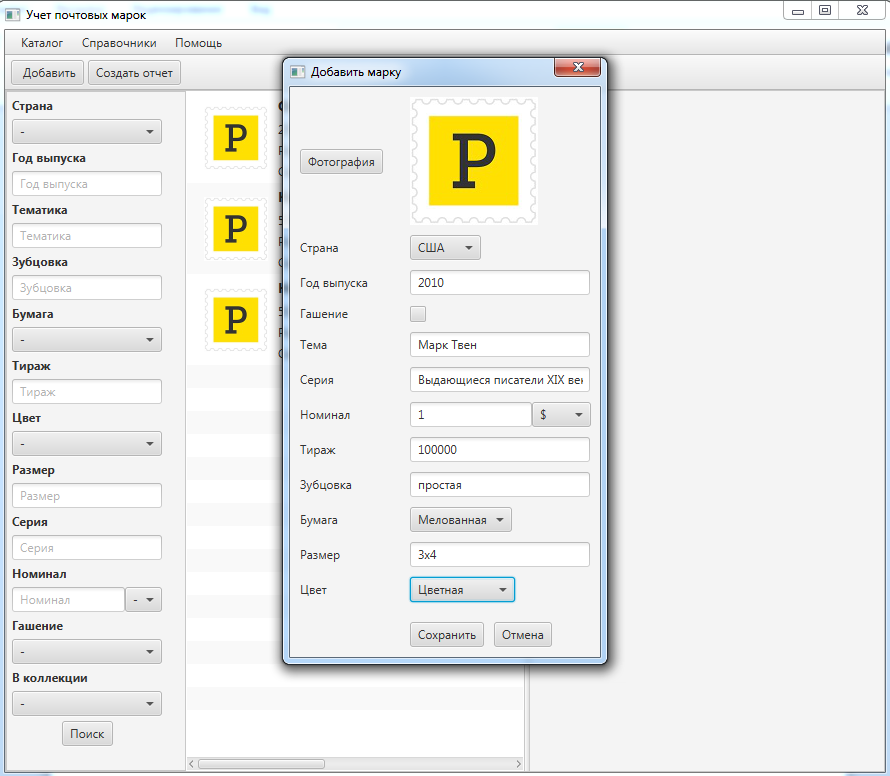


Рисунок 22 – Диалоговое окно создания новой марки

В этом окне пользователь должен задать значения параметров марки, а при желании он может прикрепить фотографию. Нажатием кнопки «Сохранить» пользователь создаст новую марку каталога.

По умолчанию критерии поиска не выставлены, поэтому отображены все марки. Нажав левой кнопкой мыши в центральной части главной формы на интересующую марку, пользователь может увидеть полную информацию о ней в правой части формы (см. рисунок 23).

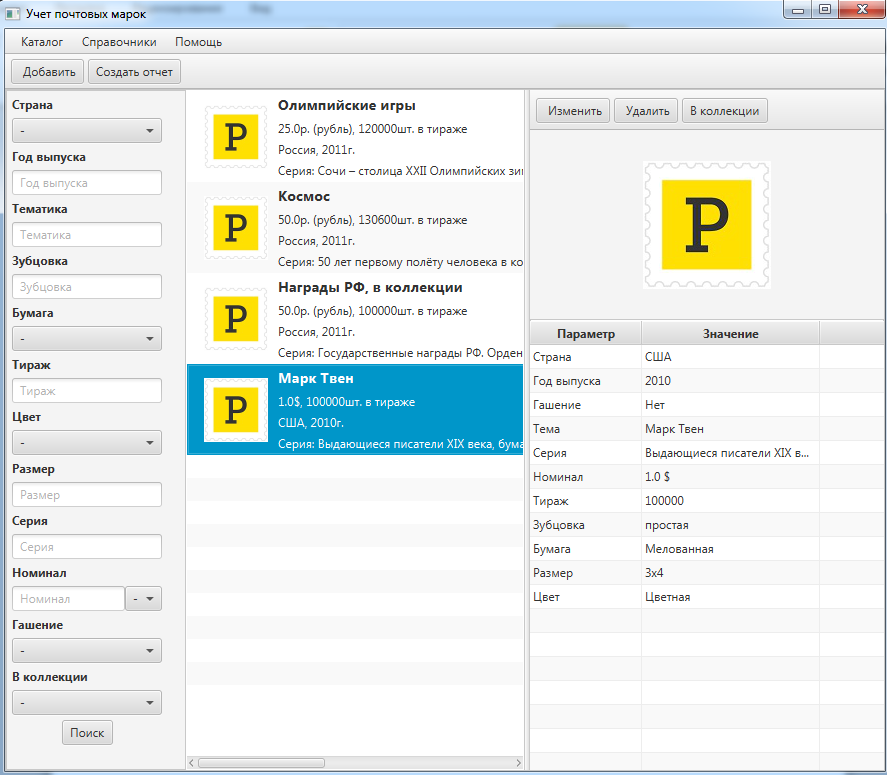


Рисунок 23 – Отображение подробной информации о выделенной марке в правой части главного окна приложения

Чтобы добавить марку в коллекцию, пользователю следует нажать на кнопку «В коллекцию», после чего, задав количество имеющихся у него марок и добавив их описание, нажать на кнопку «Сохранить» и зафиксировать внесенные изменения (см. рисунок 24).

Пользователь также может провести поиск по интересующим его критериям. На рисунке 25 показано отображение данных обо всех марках, имеющихся в пользовательской коллекции.

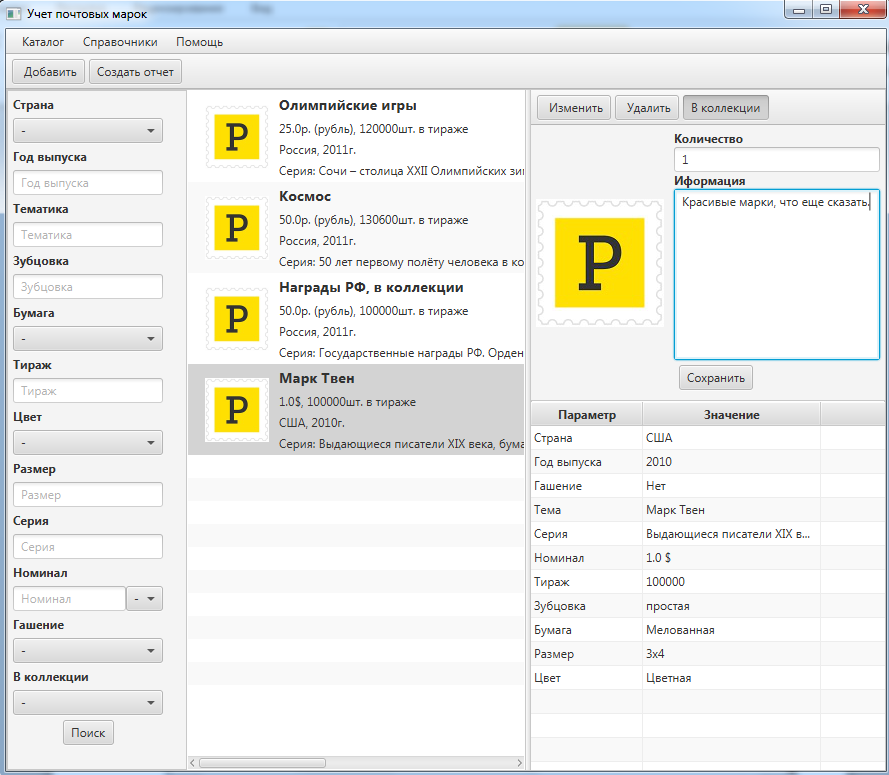


Рисунок 24 – Добавление марки из каталога в коллекцию пользователя

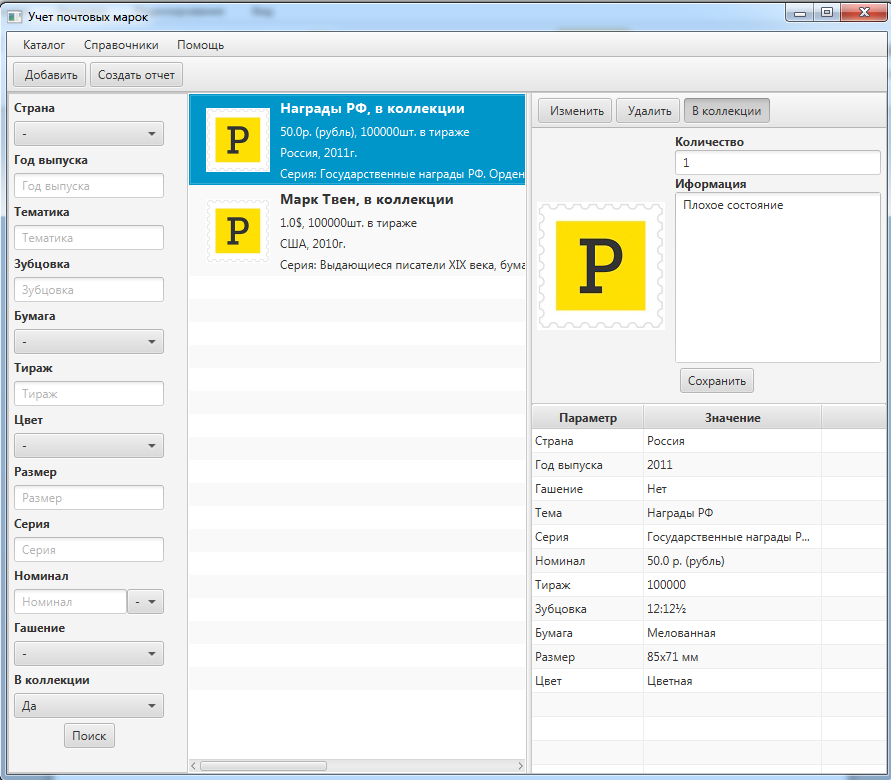


Рисунок 25 – Отображение всех марок, имеющихся в коллекции пользователя

Наконец, пользователь может распечатать отчет о выбранных марках в файл заданного формата – \*.rtf или \*.xls. Для этого ему нужно нажать на кнопку «Создать отчет» и выбрать требуемый формат файла в поле «Тип файла»(см. рисунок 26).

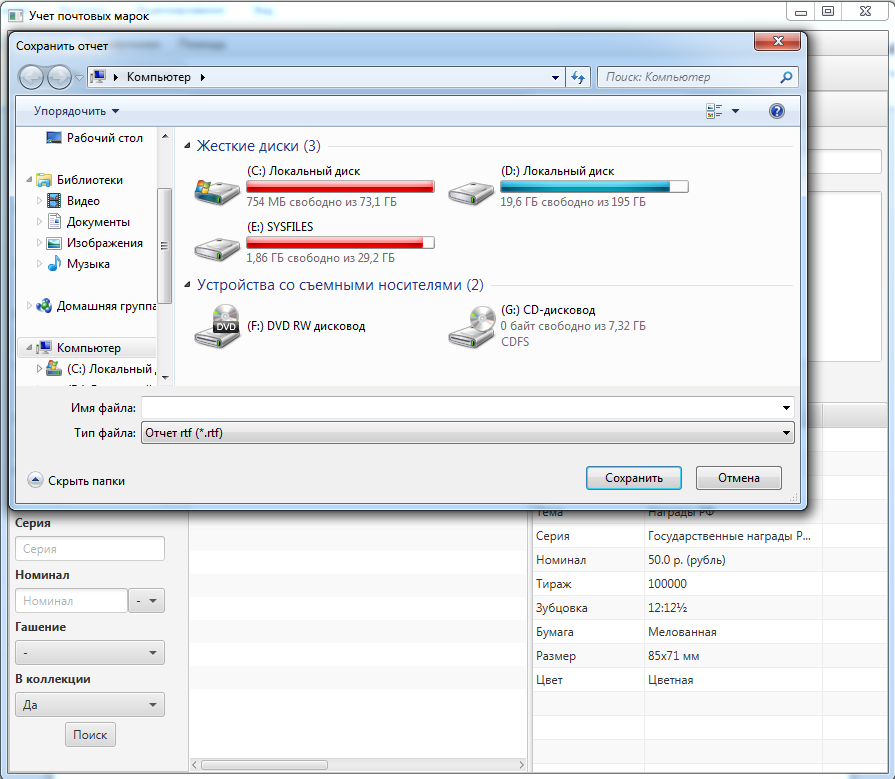


Рисунок 26 – Печать отчета о марках в текстовый файл формата \*.rtf

**ЗАКЛЮЧЕНИЕ**

В соответствии с техническим заданием был произведен анализ предметной области, реализовано программное обеспечение, составлены тестовые примеры и произведены тестирование и отладка автоматизированной системы учета почтовых марок.

В первом разделе были проведены описание и анализ предметной области, выполнен обзор существующих систем аналогов. В результате была представлена постановка задачи – детально перечислены функции, которые должны быть реализованы в разрабатываемой автоматизированной системе.

При проектировании системы был построен прототип интерфейса пользователя системы и соответствующие проекту диаграммы.

При реализации системы была произведена разработка и описание пользовательского интерфейса и описание контрольного примера. На данной стадии была продемонстрирована работа автоматизированной системы учета почтовых марок.

Созданная система имеет эргономичный пользовательский интерфейс, легка и надежна в использовании и не требует серьезных аппаратных затрат. При выполнении работы использовалась методология объектно-ориентированного анализа и проектирования, кроме того применялась технология быстрой разработки приложений RAD и методология UML.

Приложение А  
Руководство пользователя

А.1 Назначение системы

Данное приложение представляет собой программу, реализующую автоматизированную систему учета почтовых марок.

Приложение предназначено для добавления, удаления и редактирования в каталоге записей о марках, просмотре информации и печати выбранных марок в файл. При добавлении новой записи пользователь может задать параметры марки, используя сведения о марках и справочники, записи которых также можно редактировать. При поиске марок по требуемым критериям отображается информация о марках, которую можно распечатать в файл.

А.2 Условия работы системы

Для корректной работы системы необходимо наличие соответствующих программных и аппаратных средств.

Требования:

* персональный компьютер с ОС Windows XP SP2 или выше;
* не менее 460 Мб оперативной памяти;
* 8,033 Гб на жестком диске;
* дисплей с разрешением не менее 800 x 600 пикселей;
* клавиатура или иное устройство ввода;
* мышь или иное манипулирующее устройство.

А.3 Установка системы

Система поставляется в виде msi файла. Установка системы производится путем запуска файла, при этом программа устанавливается в директорию C:\Program Files\Philately-1.0. Запускаемым файлом системы является файл Philately-1.0.exe.

А.4 Работа с системой

Главная форма открывается непосредственно при запуске программы и предоставляет основные функции по управлению системой. Ее вид представлен на рисунке 28.

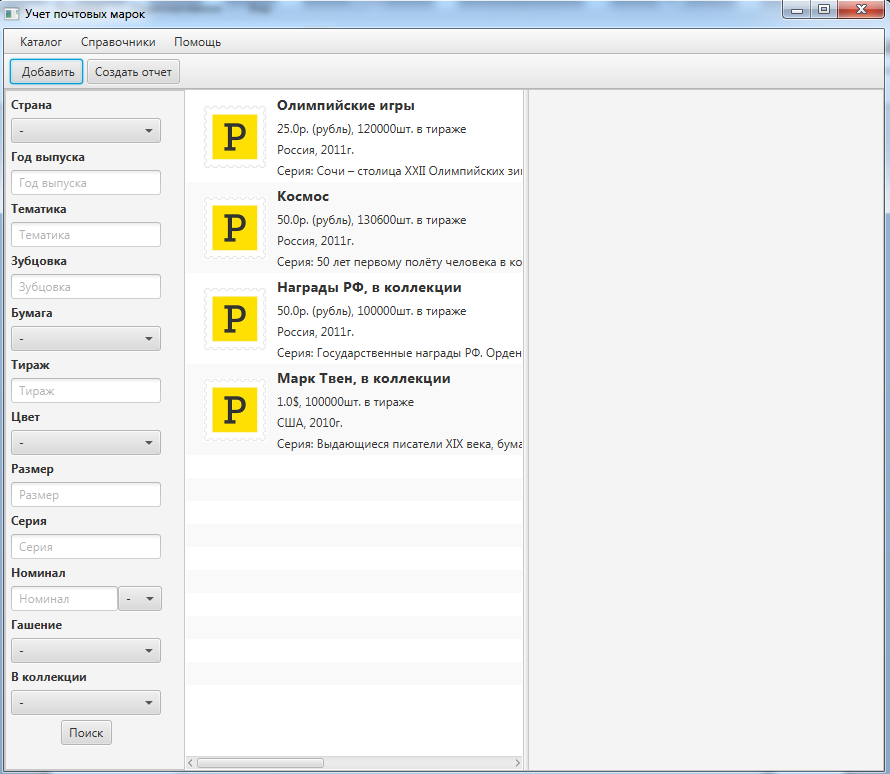


Рисунок 28 – Главная форма приложения

### А.4.1 Работа с каталогом

Пункт "Каталог" содержит пункты "Добавить", "Создать отчет", "Выход". Кроме того, под основным меню расположены кнопки, дублирующие функции добавления марки и создания отчета. Они позволяют пользователю добавить новую запись о марке, создать отчет, содержащий сведения о выбранных марках, а также покинуть программу. Подменю пункта "Каталог" представлено на рисунке 29.

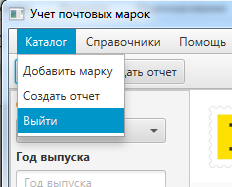


Рисунок 29 – Панель «Каталог»

В п.3.1 было рассмотрено добавление, удаление и редактирование параметров марки, поиск марок по заданным критериям, а также печать отчета в файл. Рассмотрим также справочники системы (см. рисунок 30).

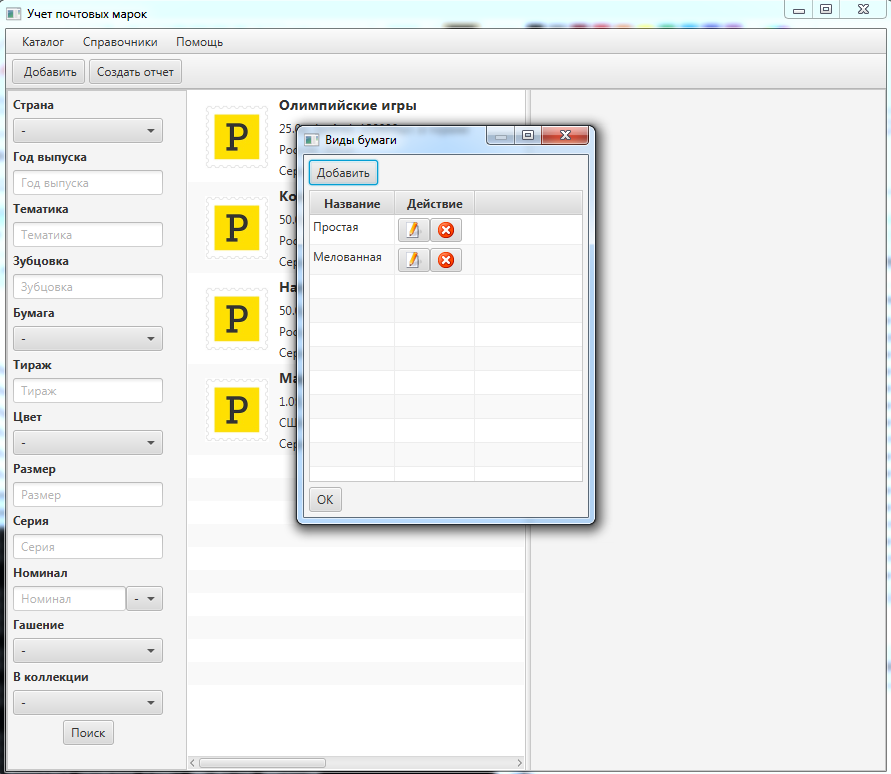


Рисунок 30 – Справочник видов бумаги

При задании таких параметров марки, как виды бумаги, тип валюты, название страны-изготовителя и цвет марки, пользователь должен выбрать наиболее подходящий вариант из записей соответствующего справочника. При необходимости пользователь может добавить новую запись в справочник, удалить ее или отредактировать.

А.4.2 Справка по системе

Для получения справочной информации на панели инструментов предусмотрена кнопка «Помощь». Нажав эту кнопку, пользователь сможет получить доступ либо к справочной информации программы, либо к информации о программе (см. рисунок 31).

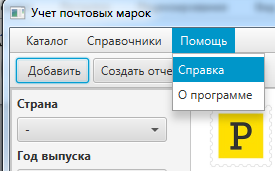


Рисунок 31 – Выпадающее меню помощи пользователю