

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра комплексной информационной безопасности электронно-вычислительных систем
(КИБЭВС)

УТВЕРЖДАЮ

заведующий каф. КИБЭВС

_____ А.А. Шелупанов

« _____ » _____ 2015г.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ДЛЯ ПРОВЕДЕНИЯ СОРЕВНОВАНИЙ В
ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Отчет по групповому проектному обучению

Группа КИБЭВС-1502

Ответственный исполнитель

студент гр. 723

_____ Д.Е. Муковкин

« _____ » _____ 2015г.

Научный руководитель

аспирант каф. КИБЭВС

_____ А.И. Гуляев

« _____ » _____ 2015г.

РЕФЕРАТ

Курсовая работа содержит 62 страниц, 53 рисунка, 7 таблицы, 10 источников, 1 приложение.

КОМПЬЮТЕРНАЯ ЭКСПЕРТИЗА, ФОРЕНЗИКА, ЛОГИ, QT, XML, GIT, LATEX, MOZILLA THUNDERBIRD, MS OUTLOOK, WINDOWS, PST, MSG, RTF, HTML, БИБЛИОТЕКИ, РЕПОЗИТОРИЙ, ПОЧТОВЫЙ КЛИЕНТ, SQLLITE, РЕЕСТР, МЕТА-ДАННЫЕ, READPST, JPEG, PNG, ID3V1, JFIF, RIFF, CHUNK, DBX, C++, DOXYGEN.

Цель работы — создание программного комплекса, предназначенного для проведения компьютерной экспертизы.

Задачей, поставленной на данный семестр, стало написание программного комплекса, имеющего следующие возможности:

- 1) сбор и анализ информации из журналов истории браузеров;
- 2) сбор и анализ информации из мессенджеров;
- 3) сбор и анализ информации из почтовых приложений;
- 4) поиск медиа-файлов (аудио, видео, изображение) и извлечение мета-данных из них;
- 5) сбора информации об установленном ПО по остаточным файлам.
- 6) сбор и анализ информации из реестра Windows.

Результаты работы в данном семестре:

– реализован импорт истории посещений, поисковых запросов, загруженных файлов, списка установленных расширений, информации о версии программы и подключенном аккаунте Google из приложения Google Chrome;

– реализован алгоритм сканирования директорий ОС Windows на содержание файлов, оставшихся при установке или после удаления различных программ;

– реализован алгоритм извлечения адресата, отправителя, темы, даты и текста сообщения из файлов формата «.dbx», используемого MS Outlook для хранения сообщений;

– реализован алгоритм извлечения времени, даты, отправителя, получателя и текст сообщения почтового клиента Mozilla Thunderbird;

– реализован алгоритм извлечения мета-данных из файлов формата ID3v1, JFIF и RIFF;

– реализован алгоритм извлечения информации из реестра Windows (.reg-файлов).

Пояснительная записка выполнена при помощи системы компьютерной вёрстки L^AT_EX.

Список исполнителей

Муковкин Д.Е. – программист, ответственный исполнитель, ответственный за развертывание инфраструктуры, настройку платформы, тестирование платформы на возможные ошибки, за выбор языка программирования, за внутреннюю архитектуру платформы, её внутренние связи, подключение модулей, работу с базой данных, за добавление в Front-end необходимых модулей для управления игрой, за разработку API функций.

Койшинов Т.С. – программист, ответственный за разработку структуры БД, ее сущности и связи в ней, за разработку различных компонентов ядра, за тестирование ядра платформы, её функционала на предмет ошибок, за добавление в существующий Front-end необходимых модулей для управления игрой.

Содержание

Введение	6
1 Назначение и область применения	6
2 Постановка задачи	6
3 Инструменты	7
3.1 Система контроля версий Git	7
3.2 Система компьютерной вёрстки \TeX	7
3.3 Система документирования Doxygen	8
3.4 Qt - кроссплатформенный инструментарий разработки ПО	8
3.4.1 Автоматизация поиска журнальных файлов	10
3.4.2 Реализация сохранения результатов работы программного комплекса в XML	10
4 Технические характеристики	11
4.1 Требования к аппаратному обеспечению	11
4.2 Требования к программному обеспечению	11
4.3 Выбор единого формата выходных файлов	11
5 Разработка программного обеспечения	12
5.1 Архитектура	12
5.1.1 Основной алгоритм	12
5.1.2 Описание основных функций модуля системы	14
5.2 Сбор информации из браузера Google Chrome	15
5.2.1 База данных Login Data Chrome	15
5.2.2 Расширения браузера Chrome (Extensions)	18
5.2.3 Изменения, добавленные в программный модуль в течение текущего семестра	20
5.3 Плагин SearchProgram	23
5.3.1 Директории в ОС, где программы могут оставить след	23
5.3.2 Список эталонных каталогов для Windows XP x32	26
5.3.3 Список эталонных каталогов для Windows XP x64	26
5.3.4 Список эталонных каталогов для Windows 7 x32	27
5.3.5 Список эталонных каталогов для Windows 7 x64	28
5.3.6 Список эталонных каталогов для Windows 8 x32	30
5.3.7 Список эталонных каталогов для Windows 8 x64	31
5.3.8 Блок-схема алгоритма работы программного модуля SearchProgram	33
5.3.9 Описание плагина TaskSearchProgram	37
5.4 Сбор информации из почтового клиента MS Outlook	40
5.4.1 Реализация программного модуля для почтового клиента MS Outlook	42
5.4.2 Задачи на следующий семестр	42
5.5 Сбор информации из почтового клиента Mozilla Thunderbird	45
5.5.1 Реализация программного модуля	45
5.5.2 Алгоритм работы модуля	46
5.5.3 Структура XML-файла	47
5.6 Поиск медиа-файлов и извлечение мета-данных	48

5.6.1 Введение	48
5.6.2 ID3v1	48
5.6.3 JFIF	50
5.6.4 RIFF	52
5.6.5 Задачи на следующий семестр	54
5.7 Сбор и анализ информации из реестра ОС MS Windows	55
5.7.1 Исследование механизма формирования реестра	55
5.7.2 Написание программного модуля, способного извлекать информацию из реестра .	55
5.7.3 Результаты работы за семестр	57
Заключение	60
Список использованных источников	61
Приложение А Компакт-диск	62

Введение

Компьютерно-техническая экспертиза – это самостоятельный род судебных экспертиз, относящийся к классу инженерно-технических экспертиз, проводимых в следующих целях: определения статуса объекта как компьютерного средства, выявление и изучение его роли в рассматриваемом деле, а так же получения доступа к информации на электронных носителях с последующим всесторонним её исследованием [1]. Компьютерная экспертиза помогает получить доказательственную информацию и установить факты, имеющие значение для уголовных, гражданских и административных дел, сопряжённых с использованием компьютерных технологий. Для проведения компьютерных экспертиз необходима высокая квалификация экспертов, так как при изучении представленных носителей информации, попытке к ним доступа и сбора информации возможно внесение в информационную среду изменений или полная утрата важных данных.

Компьютерная экспертиза, в отличие от компьютерно-технической экспертизы, затрагивает только информационную составляющую, в то время как аппаратная часть и её связь с программной средой не рассматривается.

На протяжении предыдущих семестров разработчиками данного проекта были рассмотрены такие направления компьютерной экспертизы, как исследование файловых систем, сетевых протоколов, организация работы серверных систем, механизм журналирования событий. Также были изучены основные задачи, которые ставятся перед сотрудниками правоохранительных органов, проводящими компьютерную экспертизу, и набор существующих утилит, способных помочь эксперту в проведении компьютерной экспертизы. Было выявлено, что существует множество разрозненных программ, предназначенных для просмотра лог-файлов системы и таких приложений, как мессенджеры и браузеры, но для каждого вида лог-файлов необходимо искать отдельную программу. Так как ни одна из них не позволяет эксперту собрать воедино и просмотреть все логи системы, браузеров и мессенджеров, было решено создать для этой цели собственный автоматизированный комплекс, не имеющий на данный момент аналогов в РФ.

1 Назначение и область применения

Разрабатываемый комплекс предназначен для автоматизации процесса сбора информации с исследуемого образа жёсткого диска.

2 Постановка задачи

На данный семестр были поставлены следующие задачи:

- изучение архитектуры проекта «Компьютерная экспертиза» новыми участниками проектной группы;
- изучение теоретического материала и основных инструментов разработки;
- определение индивидуальных задач для каждого участника проектной группы;
- исследование предметных областей в рамках индивидуальных задач;
- реализация новых программных модулей и доработка уже существующих;

– изучение инструментов для генерации документации к программному коду проекта.

Задачи по проектированию модулей:

- 1) сбор и анализ информации из реестра Windows;
- 2) сбор и анализ информации из браузера Google Chrome;
- 3) сбор и анализ информации из почтового клиента Mozilla Thunderbird;
- 4) сбор и анализ информации из почтового клиента MS Outlook;
- 5) поиск медиа-файлов (аудио, видео, изображение) и извлечение мета-данных из них;
- 6) сбор информации об установленном ПО по остаточным файлам.

3 Инструменты

3.1 Система контроля версий Git

Для разработки программного комплекса для проведения компьютерной экспертизы было решено использовать Git.

Git — распределённая система управления версиями файлов. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux как противоположность системе управления версиями Subversion (также известная как «SVN») [2].

При работе над одним проектом команде разработчиков необходим инструмент для совместного написания, бэкапирования и тестирования программного обеспечения. Используя Git, мы имеем:

- возможность удаленной работы с исходными кодами;
- возможность создавать свои ветки, не мешая при этом другим разработчикам;
- доступ к последним изменениям в коде, т.к. все исходники хранятся на сервере `git.keva.su`;
- исходные коды защищены, доступ к ним можно получить лишь имея RSA-ключ;
- возможность откатиться к любой стабильной стадии проекта.

Основные постулаты работы с кодом в системе Git:

- каждая задача решается в своей ветке;
- необходимо делать коммит как только был получен осмысленный результат;
- ветка `master` мерджится не разработчиком, а вторым человеком, который производит вычитку и тестирование изменения;
- все коммиты должны быть осмысленно подписаны/прокомментированы.

Для работы над проектом проектной группой был поднят собственный репозиторий на сервере `git.keva.su`. Адреса репозитория следующие:

Исходные файлы проекта:

```
git clone git@git.keva.su:gpo.git gpo.git
```

Репозиторий для тестирования проекта:

```
git clone git@git.keva.su:gpo-testdata.git gpo-testdata.git
```

3.2 Система компьютерной вёрстки \TeX

\TeX — это созданная американским математиком и программистом Дональдом Кнутом система для вёрстки текстов. Сам по себе \TeX представляет собой специализированный язык

программирования. Каждая издательская система представляет собой пакет макроопределений этого языка.

L^AT_EX — это созданная Лэсли Лэмпортом издательская система на базе T_EX'a[3]. L^AT_EX позволяет пользователю сконцентрировать свои усилия на содержании и структуре текста, не заботясь о деталях его оформления.

Для подготовки отчётной и иной документации нами был выбран L^AT_EX так как совместно с системой контроля версий Git он предоставляет возможность совместного создания и редактирования документов. Огромным достоинством системы L^AT_EX то, что создаваемые с её помощью файлы обладают высокой степенью переносимости [4].

Совместно с L^AT_EX часто используется BibT_EX — программное обеспечение для создания форматированных списков библиографии. Оно входит в состав дистрибутива L^AT_EX и позволяет создавать удобную, универсальную и долговечную библиографию. BibT_EX стал одной из причин, по которой нами был выбран L^AT_EX для создания документации.

3.3 Система документирования Doxygen

Doxygen — это кроссплатформенная система документирования исходных текстов, которая поддерживает различные языки программирования (в том числе и C++) [5].

Doxygen генерирует документацию на основе набора исходных текстов и также может быть настроен для извлечения структуры программы из недокументированных исходных кодов. Возможно составление графов зависимостей программных объектов, диаграмм классов и исходных кодов с гиперссылками.

Doxygen имеет встроенную поддержку генерации документации в формате HTML, L^AT_EX map, RTF и XML. Также вывод может быть легко сконвертирован в CHM, PostScript, PDF.

Doxygen — консольная программа в духе классической Unix. Она работает подобно компилятору, анализируя исходные тексты и создавая документацию. Параметры создания документации читаются из конфигурационного файла, имеющего простой текстовый формат.

Автором программы является голландец Димитри ван Хееш (Dimitri van Heesch).

QT Creator — среда разработки, используя разработчиками coex — поддерживает формат комментариев, используемого Doxygen при генерации документации. В ходе работы была сгенерирована документация при помощи данного инструмента, представляющая собой набор связанных html-страниц, описывающих различные классы и функции, используемые в coex. В дальнейшем планируется усовершенствовать полученную документацию, добавить описания для всех программных объектов.

3.4 Qt - кроссплатформенный инструментальный разработки ПО

Qt — это кроссплатформенная библиотека C++ классов для создания графических пользовательских интерфейсов (GUI) от фирмы Digia. Эта библиотека полностью объектно-ориентированная, что обеспечивает легкое расширение возможностей и создание новых компонентов. Ко всему прочему, она поддерживает огромное количество платформ.

Qt позволяет запускать написанное с его помощью ПО в большинстве современных операционных систем путём простой компиляции программы для каждой ОС без изменения исход-

ного кода. Включает в себя все основные классы, которые могут потребоваться при разработке прикладного программного обеспечения, начиная от элементов графического интерфейса и заканчивая классами для работы с сетью, базами данных и XML. Qt является полностью объектно-ориентированным, легко расширяемым и поддерживающим технику компонентного программирования.

Список использованных классов фреймворка QT

- `iostream`
- `QChar`
- `QCryptographicHash`
- `QDateTime`
- `QDir`
- `QDirIterator`
- `QFile`
- `QFileInfo`
- `QIODevice`
- `QList`
- `QRegExp`
- `QString`
- `QTextStream`
- `QtSql/SqlDatabase`
- `QVector`
- `QMap`
- `QXmlStreamReader`
- `QXmlStreamWriter`
- `Conversations`

Класс `QXmlStreamWriter` представляет собой XML писателя с простым потоковым.

Класс `QXmlStreamReader` представляет собой быстрый синтаксически корректный XML анализатор с простым потоковым API.

`QVector` представляет собой класс для создания динамических массивов.

Модуль `QtSql/SqlDatabase` помогает обеспечить однородную интеграцию БД в ваши Qt приложения.

Класс `QTextStream` предоставляет удобный интерфейс для чтения и записи текста.

`QTextStream` может взаимодействовать с `QIODevice`, `QByteArray` или `QString`. Используя потоковые операторы `QTextStream`, вы можете легко читать и записывать слова, строки и числа. При формировании текста `QTextStream` поддерживает параметры форматирования для заполнения и выравнивания полей и форматирования чисел. [6]

Класс `QString` предоставляет строку символов Unicode.

Класс `QMap` — контейнерный класс для хранения элементов различных типов данных.

Класс `QDateTime` используется для работы с форматом даты, в который записывается информация о файле.

`QString` хранит строку 16-битных `QChar`, где каждому `QChar` соответствует один символ Unicode 4.0. (Символы Unicode со значениями кодов больше 65535 хранятся с использованием

суррогатных пар, т.е. двух последовательных QChar.)

Unicode - это международный стандарт, который поддерживает большинство использующихся сегодня систем письменности. Это расширение US-ASCII (ANSI X3.4-1986) и Latin-1 (ISO 8859-1), где все символы US-ASCII/Latin-1 доступны на позициях с тем же кодом.

Внутри QString использует неявное совместное использование данных (копирование-при-записи), чтобы уменьшить использование памяти и избежать ненужного копирования данных. Это также позволяет снизить накладные расходы, свойственные хранению 16-битных символов вместо 8-битных.

В дополнение к QString Qt также предоставляет класс QByteArray для хранения сырых байт и традиционных нультерминальных строк. В большинстве случаев QString - необходимый для использования класс. Он используется во всем API Qt, а поддержка Unicode гарантирует, что ваши приложения можно будет легко перевести на другой язык, если в какой-то момент вы захотите увеличить их рынок распространения. Два основных случая, когда уместно использование QByteArray: когда вам необходимо хранить сырые двоичные данные и когда критично использование памяти (например, в Qt для встраиваемых Linux-систем).[7]

Класс QRegExp предоставляет сопоставление с образцом при помощи регулярных выражений.

Регулярное выражение, или "regex", представляет собой образец для поиска соответствующей подстроки в тексте. Это полезно во многих ситуациях, например:

Проверка правильности – регулярное выражение может проверить, соответствует ли подстрока каким-либо критериям, например, целое ли она число или не содержит ли пробелов. Поиск – регулярное выражение предоставляет более мощные шаблоны, чем простое соответствие строки, например, соответствие одному из слов mail, letter или correspondence, но не словам email, mailman, mailer, letterbox и т.д. Поиск и замена – регулярное выражение может заменить все вхождения подстроки другой подстрокой, например, заменить все вхождения & на &;, исключая случаи, когда за & уже следует amp;. Разделение строки – регулярное выражение может быть использовано для определения того, где строка должна быть разделена на части, например, разделяя строку по символам табуляции.

QFileInfo - Во время поиска возвращает полную информацию о файле.

Класс QDir обеспечивает доступ к структуре каталогов и их содержимого.

QIODevice представляет собой базовый класс всех устройств ввода/вывода в Qt.

Класс QCryptographicHash предоставляет способ генерации криптографических хэшей. QCryptographicHash могут быть использованы для генерации криптографических хэшей двоичных или текстовых данных. В настоящее время MD4, MD5, и SHA-1 поддерживаются.[7]

QChar обеспечивает поддержку 16-битных символов Unicode.

3.4.1 Автоматизация поиска журнальных файлов

Для сканирования образа на наличие интересующих лог файлов использовался класс QDirIterator. После вызова происходит поочередный обход по каждому файлу в директории и поддиректории. Проверка полученного полного пути к файлу осуществляется регулярным выражением, если условие выполняется, происходит добавление в список обрабатываемых фай-

ЛОВ.

3.4.2 Реализация сохранения результатов работы программного комплекса в XML

Сохранение полученных данных происходит в ранее выбранный формат XML(Extensible Markup Language). Для этого используется класс QXmlStreamReader и QxmlStreamWriter. Класс QXmlStreamWriter представляет XML писателя с простым потоковым API.

QXmlStreamWriter работает в связке с QXmlStreamReader для записи XML. Как и связанный класс, он работает с QIODevice, определённым с помощью setDevice().

Сохранение данных реализованно в классе WriteMessage. В методе WriteMessages, структура которого представлена на UML диаграмме в разделе Архитектура.

4 Технические характеристики

4.1 Требования к аппаратному обеспечению

Минимальные системные требования:

- процессор 1ГГц Pentium 4;
- оперативная память 512 Мб;
- место на жёстком диске – 9 Гб.

4.2 Требования к программному обеспечению

Для корректной работы разрабатываемого программного комплекса на компьютере должна быть установлена операционная система Debian Squeeze или выше, данная система должна иметь набор библиотек QT.

4.3 Выбор единого формата выходных файлов

Для вывода результата был выбран формат XML-документов, так как с данным форматом легко работать при помощи программ, а результат работы данного комплекса в дальнейшем планируется обрабатывать при помощи программ.

XML - eXtensible Markup Language или расширяемый язык разметки. Язык XML представляет собой простой и гибкий текстовый формат, подходящий в качестве основы для создания новых языков разметки, которые могут использоваться в публикации документов и обмене данными [8]. Задумка языка в том, что он позволяет дополнять данные метаданными, которые разделяют документ на объекты с атрибутами. Это позволяет упростить программную обработку документов, так как структурирует информацию.

Простейший XML-документ может выглядеть так:

```
<?xml version="1.0"?>
<list_of_items>
<item id="1"><first>Первый</item>
<item id="2">Второй <subsub_item>подпункт 1</subsub_item></item>
<item id="3">Третий</item>
<item id="4"><last/>Последний</item>
</list_of_items>
```

Первая строка - это объявление начала XML-документа, дальше идут элементы документа `<list_of_items>` - тег описывающий начало элемента `list_of_items`, `</list_of_items>` - тег конца элемента. Между этими тегами заключается описание элемента, которое может содержать текстовую информацию или другие элементы (как в нашем примере). Внутри тега начала элемента так же могут указывать атрибуты элемента, как например атрибут `id` элемента `item`, атрибуту должно быть присвоено определенное значение.

5 Разработка программного обеспечения

5.1 Архитектура

5.1.1 Основной алгоритм

В ходе разработки был применен видоизменённый шаблон проектирования `Factory method`.

Данный шаблон относится к классу порождающих шаблонов. Шаблоны данного класса - это шаблоны проектирования, которые абстрагируют процесс инстанцирования (создания экземпляра класса). Они позволяют сделать систему независимой от способа создания, композиции и представления объектов. Шаблон, порождающий классы, использует наследование, чтобы изменять инстанцируемый класс, а шаблон, порождающий объекты, делегирует инстанцирование другому объекту. Пример организации проекта при использовании шаблона проектирования `Factory method` представлен на рисунке 5.1.

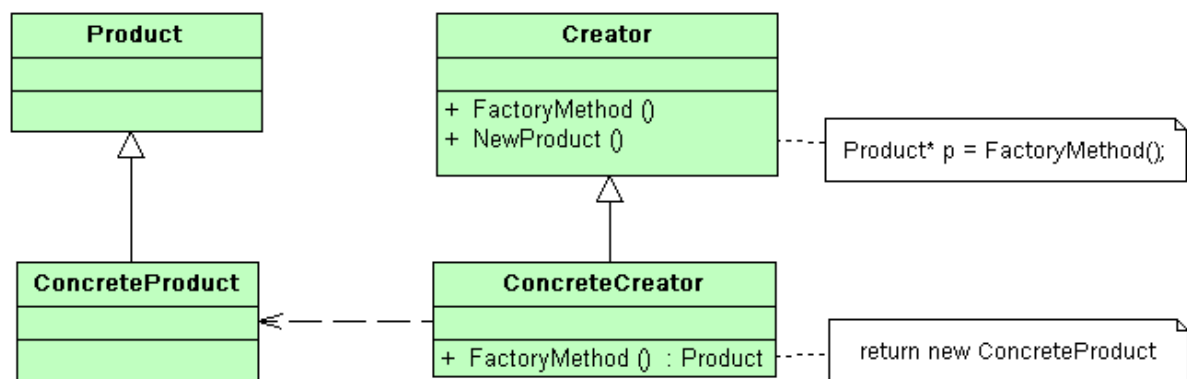


Рисунок 5.1 – Пример организации проекта при использовании шаблона проектирования `Factory method`

Использование данного шаблона позволило разбить проект на независимые модули, что весьма упростило задачу разработки, так как написание алгоритма для конкретного таска не влияло на остальную часть проекта. При разработке был реализован базовый класс для работы с образом диска. Данный класс предназначался для формирования списка настроек, определения операционной системы на смонтированном образе и инстанционировании и накопывание всех необходимых классов-тасков в очереди тасков. После чего каждый таск из очереди отправлялся на выполнение. Блоксхема работы алгоритма представлена на рисунке 5.2.

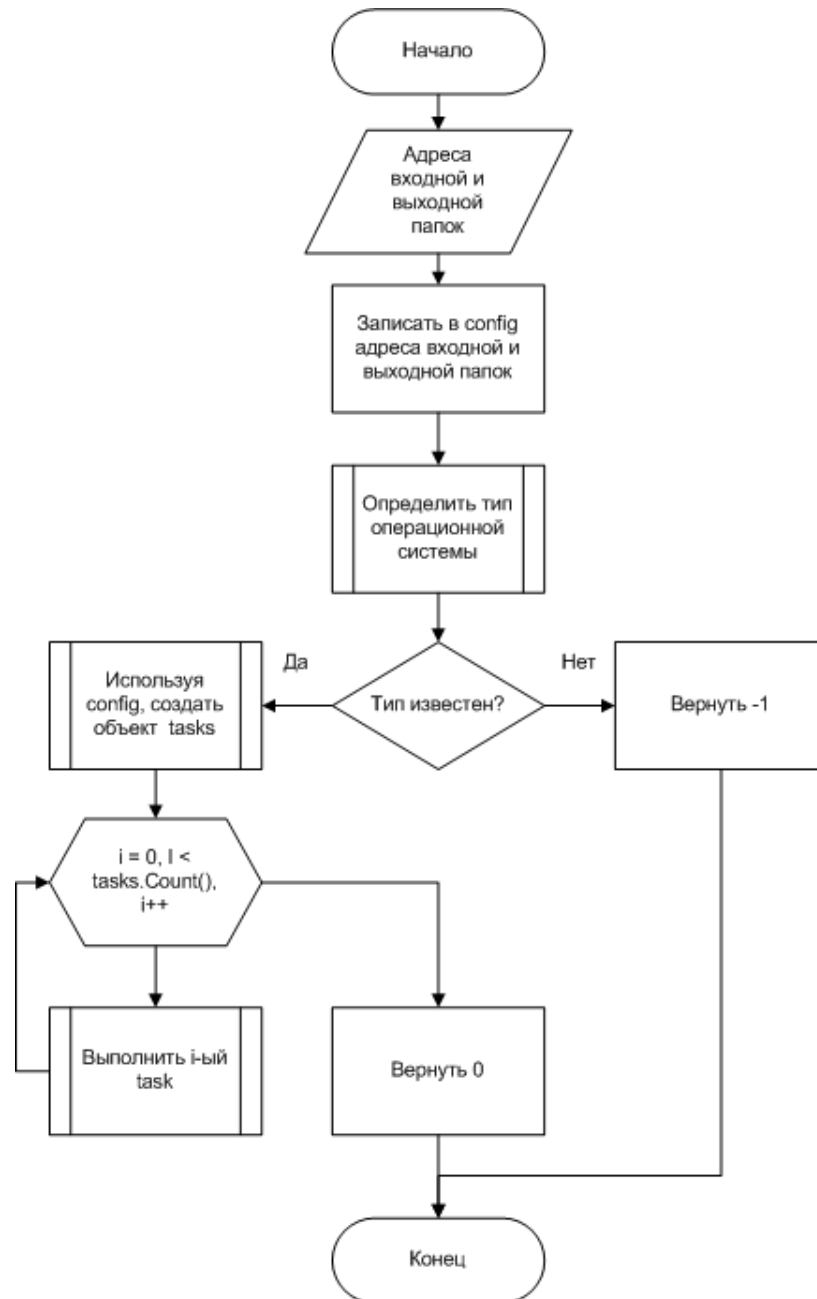


Рисунок 5.2 – Алгоритм работы с образом диска

Каждый класс-задача порождается путем наследования от базового абстрактного класса, который имеет 8 методов и 3 атрибута:

- 1) `QString manual()` - возвращает справку о входных параметрах данной задачи;
- 2) `void setOption(QStringList list)` - установка флагов для поданных на вход параметров;
- 3) `QString command()` - возвращает команду для инициализации задачи вручную;
- 4) `bool supportOS(const coex::typeOS &os)` - возвращает флаг, указывающий на возможность использования данной задачи для конкретной операционной системы;
- 5) `QString name()` - возвращает имя данной задачи;
- 6) `QString description()` - возвращает краткое описание задачи;
- 7) `bool test()` - предназначена для теста на доступность задачи;
- 8) `bool execute(const coex::config &config)` - запуск задачи на выполнение;
- 9) `QString m_strName` - хранит имя задачи;

10) QString m_strDescription - хранит описание задачи;

11) bool m_bDebug - флаг для параметра -debug;

На данный момент в проекте используется восемь классов. UML-диаграмма классов представлена на рисунке 5.3.

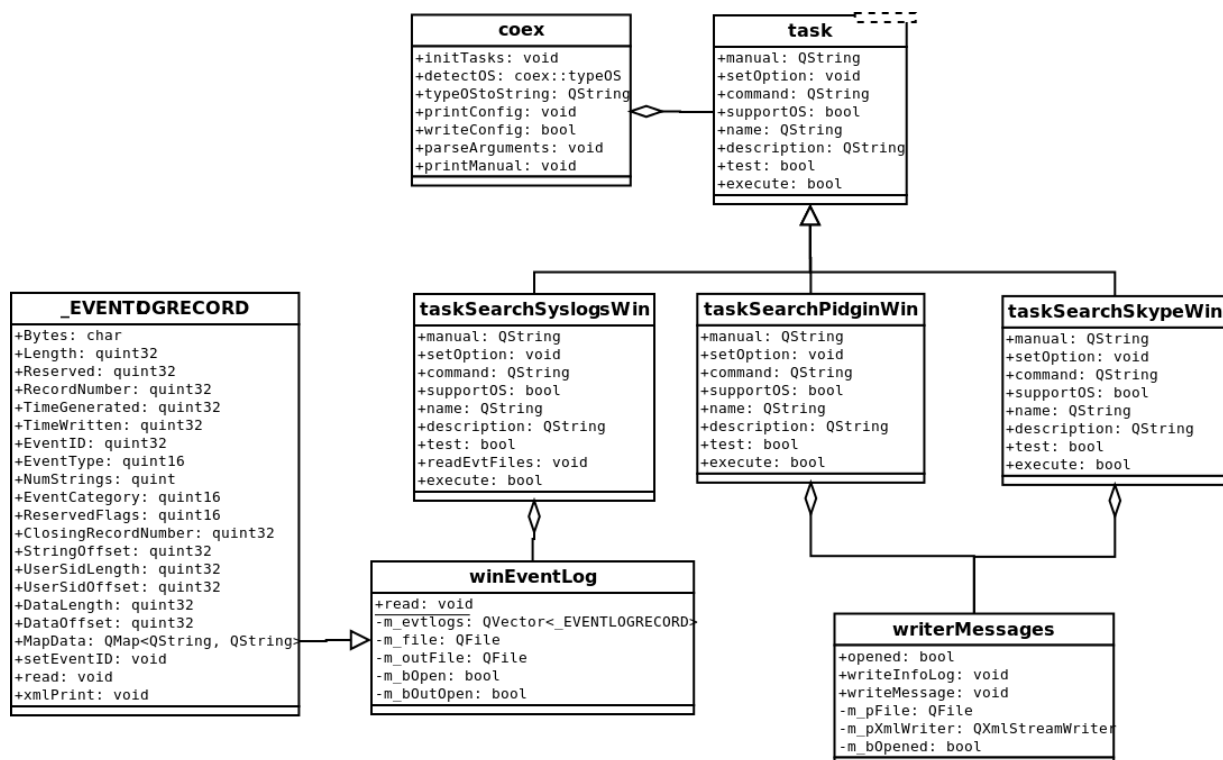


Рисунок 5.3 – UML-диаграмма классов

Классы taskSearchSyslogsWin, taskSearchPidginWin и taskSearchSkypeWin - наследники от класса task являются задачами. Класс winEventLog и _EVENTLOGRECORD предназначены для конвертации журнальных файлов операционной системы Windows XP, а класс writerMessages для преобразования истории переписки.

5.1.2 Описание основных функций модуля системы

Любой модуль системы является классом-наследником от некоторого абстрактного класса используемого как основу для всех модулей программы (шаблон проектирования Factory method). Модуль содержит в себе 8 методов и 3 атрибута:

QString manual() - возвращает справку о входных параметрах данного задания

void setOption(QStringList list) - установка флагов для поданных на вход параметров

QString command() - возвращает команду для инициализации задания вручную

bool supportOS(const coex::typeOS &os) - возвращает флаг указывающий на возможность использования данного задания для конкретной операционной системы

QString name() - возвращает имя данного задания

QString description() - возвращает краткое описание задания

bool test() - предназначена для проверки работоспособности задания

bool execute(const coex::config &config) - запуск задания на выполнение

QString m_strName - хранит имя задания

QString m_strDescription - хранит описание задачи

bool m_bDebug - флаг для параметра `-debug`

5.2 Сбор информации из браузера Google Chrome

Целью работы в текущем семестре являлось исследование журнальных файлов и доработка программного модуля для сбора пользовательских данных приложения Google Chrome и представления их в формате XML.

В ходе изучения работы данного браузера было установлено, что приложение Google Chrome хранит пользовательские данные локально. Адреса директорий, используемых по умолчанию для этих целей Google Chrome можно увидеть в таблице 5.1, пользовательские данные — в таблице 5.2.

Таблица 5.1 – Директории хранения журнальных файлов Chrome

Операционная система	Директория
Linux (Debian)	/home/имя пользователя/.config/google-chrome/Default/
Win7	C:\Users\имя пользователя\AppData\Local\Google\Chrome\User Data\Default\
Win8	C:\Users\имя пользователя\AppData\Local\Google\Chrome\User Data\Default\

Таблица 5.2 – Пользовательские данные

Файл	Содержание
Bookmarks	Закладки
History	История посещений, история запросов, история загруженных файлов
Preferences	Настройки (директория загрузки файлов, версия программы, логин аккаунта Google)
Login Data	Сохраненные логины и пароли
Extensions (папка)	Расширения

5.2.1 База данных Login Data Chrome

Login Data — это реляционная база данных, основанная на СУБД SQLite. Необходимо рассмотреть данную БД, которая содержит 2 таблицы:

- 1) logins;
- 2) meta.

Интерес представляет только таблица logins. Она содержит следующие поля:

- 1) origin_url — адрес ресурса;
- 2) username_value — логин для доступа;
- 3) password_value — пароль, представленный в виде BLOB массива двоичных данных (Binary Large Object);

4) `date_created` — дата сохранения, представленная в следующем виде (пример): 13072972925957814. Это число есть количество секунд, прошедшее с 00:00:00 UTC 1 января, 1601 года (рис. 5.4).

rowid	origin_url	action_url	userna...	username_value	passwor...	passwor...	submit_...	signon_...	ssl_valid	preferred	date_created
5	https://accounts.google.com/ServiceLogin	https://a...	Email	pupkinv086@gmail.com	Passwd	BLOB (Si...		https://a...	1	1	13075215386640954
4	http://pikabu.ru/	http://pi...	email	g2976460@trbvm.com		BLOB (Si...		http://pi...	0	1	13075215016284834
3	https://turbik.tv/Signin	https://t...	login	staber	passwd	BLOB (Si...		https://t...	1	1	13070804018330502
2	https://vk.com/login.php	https://l...	email	sgipovskoi@gmail.com	pass	BLOB (Si...		https://v...	1	1	13070828223000000
1	https://steamcommunity.com/openid/login	https://s...	username	scang9	password	BLOB (Si...		https://s...	1	1	13072972925957814

Рисунок 5.4 – Структура таблицы login

Запрос для импорта данных выглядит следующим образом:

```
SELECT logins.origin_url,
       logins.username_value,
       datetime(logins.date_created/1000000 +
               (strftime('%s', '1601-01-01')), 'unixepoch')
FROM logins;
```

Результат выполнения запроса можно увидеть на рисунке 5.5, блок-схему алгоритма выборки данных из БД Login Data — на рисунке 5.6. Результат выполнения программы в формате XML — рисунок 5.7.

Значение поля `id` — уникальный идентификатор для последующего импорта в solr БД и работы с ним.

origin_url	username_value	datetime(date_created/1000000+(strftime('%s','1601-01-01')),'unixepoch')
https://steamcommunity.com/openid/login	scang9	2015-04-08 13:22:05
https://vk.com/login.php	sgipovskoi@gmail.com	2015-03-14 17:37:03
https://turbik.tv/Signin	staber	2015-03-14 10:53:38
http://pikabu.ru/	g2976460@trbvm.com	2015-05-04 12:10:16
https://accounts.google.com/ServiceLogin	pupkinv086@gmail.com	2015-05-04 12:16:26

Рисунок 5.5 – Результат выполнения запроса

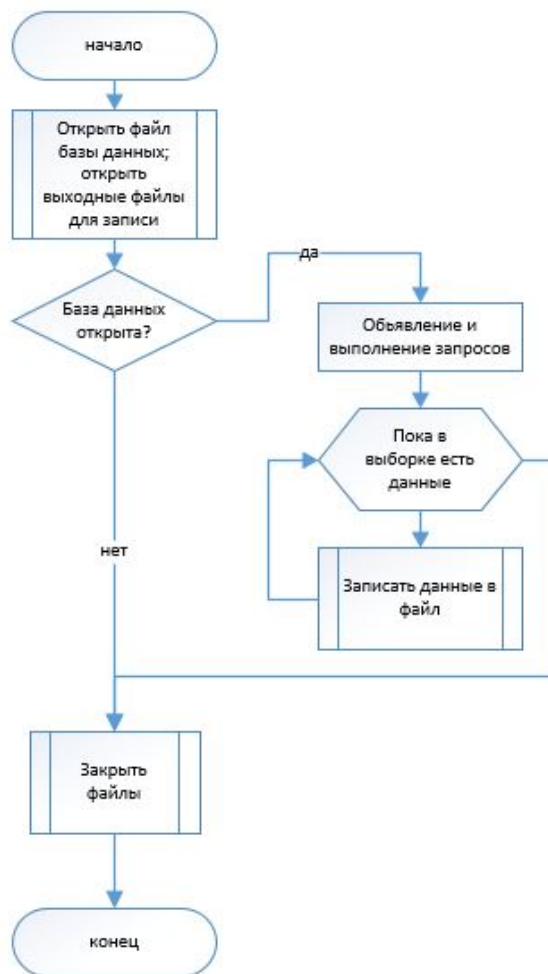


Рисунок 5.6 – Блок-схема алгоритма выборки данных из БД Login Data

```

<?xml version="1.0" encoding="UTF-8"?>
<add>
  <doc>
    <field name="doc_type">login</field>
    <field name="id">chrome_86abe881ff409a9ee1ea0924552ea9fe</field>
    <field name="application">chrome</field>
    <field name="owner">svy</field>
    <field name="login_param_url">https://turbik.tv/Signin</field>
    <field name="login_param_login">staber</field>
    <field name="login_param_date_create">2015-03-14 10:53:38</field>
  </doc>
  <doc>
    <field name="doc_type">login</field>
    <field name="id">chrome_543c82c64865957598bc813e40ccd259</field>
    <field name="application">chrome</field>
    <field name="owner">svy</field>
    <field name="login_param_url">https://steamcommunity.com/openid/login</field>
    <field name="login_param_login">scang9</field>
    <field name="login_param_date_create">2015-04-08 13:22:05</field>
  </doc>
  <doc>
    <field name="doc_type">login</field>
    <field name="id">chrome_235669518bebe0723b1fb367cef73851</field>
    <field name="application">chrome</field>
    <field name="owner">svy</field>
    <field name="login_param_url">https://login.vk.com/</field>
    <field name="login_param_login">sgipovskoi@gmail.com</field>
    <field name="login_param_date_create">2015-03-14 17:37:03</field>
  </doc>
</add>

```

Рисунок 5.7 – Файл login.XML

5.2.2 Расширения браузера Chrome (Extensions)

В папке Extensions (рис. 5.8) находятся данные об установленных в браузере расширениях. Для каждого расширения имеется своя папка, в которой находится различная информация. Также для каждого Extension имеется файл manifest (рис. 5.9) с расширением JSON. JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript. Из данного файла необходима только информация об имени и версии расширения.

Блок-схему алгоритма импорта данных о расширениях можно увидеть на рисунке 5.10.

Имя	Дата изменения	Тип	Размер
аароссclcgogkmnckokdopfmhонfmgоек	04.05.2015 21:13	Папка с файлами	
аohghmighlieiainnegkcijnfilokake	04.05.2015 21:13	Папка с файлами	
apdfllckaahabafndbhieahigklhalf	04.05.2015 21:13	Папка с файлами	
beepbmhgboaoogfdajaanbcjnmnhjmhfн	04.05.2015 21:13	Папка с файлами	
blpcfgoakmgmkcojhhkbfbldkacnbeo	04.05.2015 21:13	Папка с файлами	
cfhdojbkjhnklbpkdaibcdccddlifddb	04.05.2015 21:13	Папка с файлами	
coobgpohoikkipblmiejlniedjppdf	04.05.2015 21:13	Папка с файлами	
cpokhfcmgppipfplgbkiecbpcmplgniam	04.05.2015 21:13	Папка с файлами	
felcaaldnbdncclmgdcncolpebgiejap	04.05.2015 21:13	Папка с файлами	
gkojfkhlkighikafcpjkikfblnmeio	04.05.2015 21:13	Папка с файлами	
gmlllbghnfpflemihjekbapjopfjik	04.05.2015 21:13	Папка с файлами	
hdokiejnpimakedhajhdlcegeplioahd	04.05.2015 21:13	Папка с файлами	
jpniccbojbdjnnnclhelaenfhhbknlан	04.05.2015 21:13	Папка с файлами	
lccekmodgklaepjeofdjpbminlajkg	04.05.2015 21:13	Папка с файлами	
lfpjkcokllnfokkgpkobnkbkmelfej	04.05.2015 21:14	Папка с файлами	
nmmhkkegccagdldgiimedpiccmgmieda	04.05.2015 21:14	Папка с файлами	
pjkljhegncpnkpnbcncohdijeoejaedia	04.05.2015 21:14	Папка с файлами	

Рисунок 5.8 – Папка Extensions

```

1 {
2   "background": {
3     "scripts": [ "background.js" ]
4   },
5   "content_scripts": [ {
6     "css": [ "vk-download_styles.css" ],
7     "js": [ "jquery_min.js", "contentscript.js" ],
8     "matches": [ "https://vk.com/*", "http://vk.com/*" ],
9     "run_at": "document_end"
10  }, {
11    "js": [ "addon.js" ],
12    "matches": [ "http://*/*", "https://*/*" ],
13    "run_at": "document_idle"
14  } ],
15   "description": "Modifies pages with audio (eg. 'My music', 'Suggeste
16   "icons": {
17     "128": "download128.png",
18     "16": "download-icon.png",
19     "48": "download48.png"
20  },
21   "key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAlnimdw6mBo4NhBZ
22   "manifest_version": 2,
23   "name": "VK Music Downloader",
24   "permissions": [ "https://vk.com/*", "http://vk.com/*" ],
25   "short_name": "VK Music Downloader",
26   "update_url": "https://clients2.google.com/service/update2/crx",
27   "version": "1.1",
28   "web_accessible_resources": [ "download-icon.png" ]
29 }
30

```

Рисунок 5.9 – Файл manifest.json

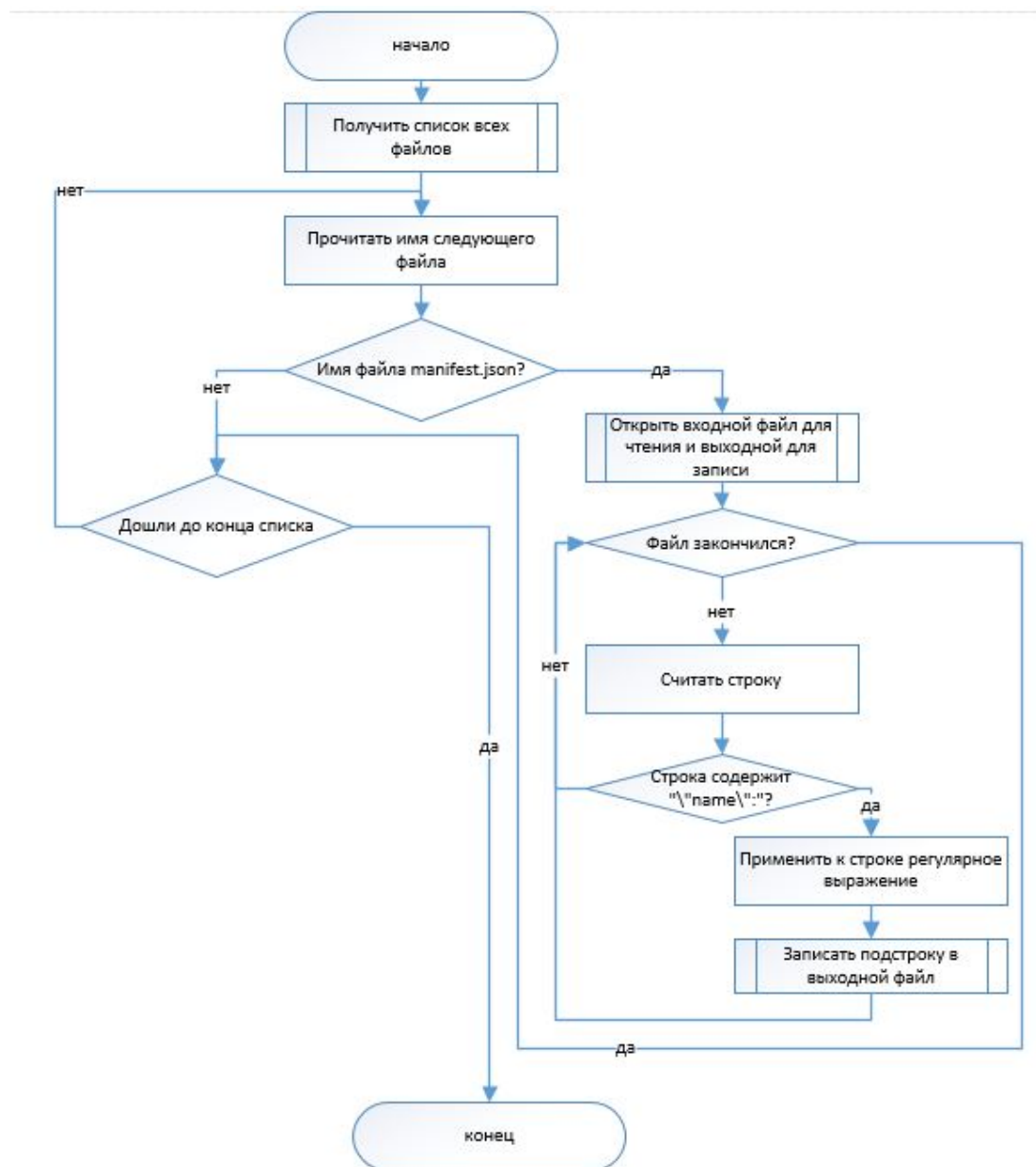


Рисунок 5.10 – Блок-схема алгоритма импорта данных о расширениях

Извлечение подстроки из строки осуществляется с помощью регулярного выражения `\"(.*)\".*(.*)\"`. Например, есть строка «name»: «VK Music Downloader». Данное регулярное выражение возвращает 2 подстроки — «name» и «VK Music Downloader», что и требовалось в ходе работы.

Результат был записан в файл extensions.XML (рис. 5.11).



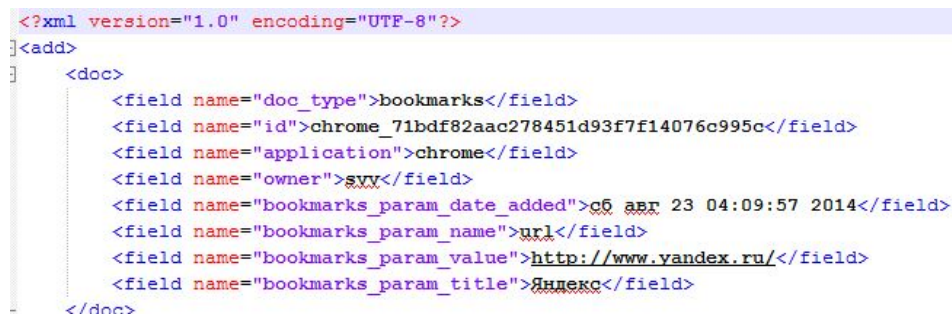
```
<?xml version="1.0" encoding="UTF-8"?>
<add>
  <doc>
    <field name="doc_type">extension</field>
    <field name="id">chrome_008f43a510638330fbfe46f5842cbc4e</field>
    <field name="application">chrome</field>
    <field name="extension_param_owner">svy</field>
    <field name="extension_param_name">_MSG_appName_</field>
  </doc>
  <doc>
    <field name="doc_type">extension</field>
    <field name="id">chrome_10a2696ab52e3e0642b6ba966b09b905</field>
    <field name="application">chrome</field>
    <field name="extension_param_owner">svy</field>
    <field name="extension_param_name">Google Voice Search Hotword (Beta)</field>
  </doc>
  <doc>
    <field name="doc_type">extension</field>
    <field name="id">chrome_35c18af34b33d93c21faffef4aa29c37</field>
    <field name="application">chrome</field>
    <field name="extension_param_owner">svy</field>
    <field name="extension_param_name">Chrome Hotword Shared Module</field>
  </doc>
  <doc>
    <field name="doc_type">extension</field>
    <field name="id">chrome_dd6b43934e9483a2a37c0edb4efedf82</field>
    <field name="application">chrome</field>
    <field name="extension_param_owner">svy</field>
    <field name="extension_param_name">Linkclump</field>
  </doc>
</add>
```

Рисунок 5.11 – Файл extensions.XML

5.2.3 Изменения, добавленные в программный модуль в течение текущего семестра

В файл bookmarks.XML добавлены 2 поля (рис. 5.12):

- 1) дата добавления закладки;
- 2) владелец файла.



```
<?xml version="1.0" encoding="UTF-8"?>
<add>
  <doc>
    <field name="doc_type">bookmarks</field>
    <field name="id">chrome_71bdf82aac278451d93f7f14076c995c</field>
    <field name="application">chrome</field>
    <field name="owner">svy</field>
    <field name="bookmarks_param_date_added">06 apr 23 04:09:57 2014</field>
    <field name="bookmarks_param_name">url</field>
    <field name="bookmarks_param_value">http://www.yandex.ru/</field>
    <field name="bookmarks_param_title">Яндекс</field>
  </doc>
</add>
```

Рисунок 5.12 – Файл bookmarks.XML

В файл history.XML (рис. 5.13) добавлено поле-дата последнего посещения ресурса.

Также было реализовано преобразование данных времени начала и конца загрузки, а также о количестве занимаемого места к читаемому виду (рис. 5.14).



Рисунок 5.13 – Файл history.XML



Рисунок 5.14 – Файл downloads.XML

Помимо этого реализованы следующие задачи:

- 1) присоединение модуля к общей системе соех;
- 2) рекурсивный обход файловой системы для нахождения входных файлов;
- 3) различие выходных данных при обработке входных от нескольких пользователей путём добавления в конец имени файла даты и количества наносекунд, прошедших с начала работы функции.

На данный момент реализован импорт следующих данных:

- 1) история посещений;
- 2) история загруженных файлов;
- 3) история поисковых запросов;
- 4) список установленных расширений;
- 5) информация о версии программы, подключённом аккаунте google;
- 6) сохраненные данные для доступа к ресурсам(только логин).

Список всех выходных XML-файлов приведен на рисунке 5.15.

Имя	Дата изменения	Тип	Размер
bookmarks_2015-05-04_21-08-3976119176	04.05.2015 21:07	Документ XML	2 КБ
bookmarks_2015-05-04_21-08-3976119301	04.05.2015 21:07	Документ XML	51 КБ
download_history_2015-05-04_21-08-3976119198	04.05.2015 21:07	Документ XML	3 КБ
download_history_2015-05-04_21-08-4776127362	04.05.2015 21:07	Документ XML	59 КБ
extensions_2015-05-04_21-08-3976119324	04.05.2015 21:07	Документ XML	6 КБ
history_2015-05-04_21-08-3976119198	04.05.2015 21:08	Документ XML	14 КБ
history_2015-05-04_21-08-4776127362	04.05.2015 21:08	Документ XML	874 КБ
login_2015-05-04_21-08-3976119133	04.05.2015 21:08	Документ XML	2 КБ
preferences_2015-05-04_21-08-3976119183	04.05.2015 21:08	Документ XML	1 КБ
preferences_2015-05-04_21-08-3976119313	04.05.2015 21:08	Документ XML	2 КБ
search_term_2015-05-04_21-08-3976119198	04.05.2015 21:08	Документ XML	1 КБ
search_term_2015-05-04_21-08-4776127362	04.05.2015 21:08	Документ XML	30 КБ

Рисунок 5.15 – Файл downloads.XML

5.3 Плагин SearchProgram

Целью работы в текущем семестре стало написание программного модуля, осуществляющего поиск в ОС Windows (XP, 7, 8) следов установленных или удаленных программ. При этом реестр и директорию меню «Пуск» учитывать было не нужно.

5.3.1 Директории в ОС, где программы могут оставить след

В любой версии Windows (XP, 7, 8) необходимые директории:

- 1) Program Files (рис. 5.16);
- 2) Program Files (x86) (рис. 5.17);
- 3) Program Files\Common Files (рис. 5.18);
- 4) Program Files (x86)\Common Files (рис. 5.19).

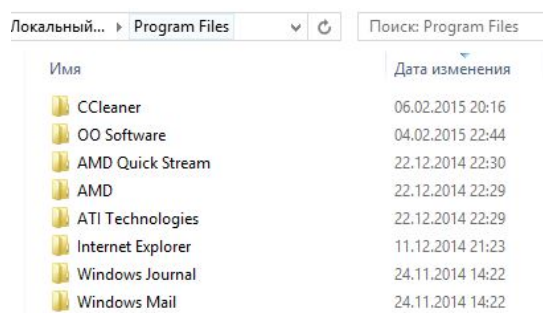


Рисунок 5.16 – Директория Program Files

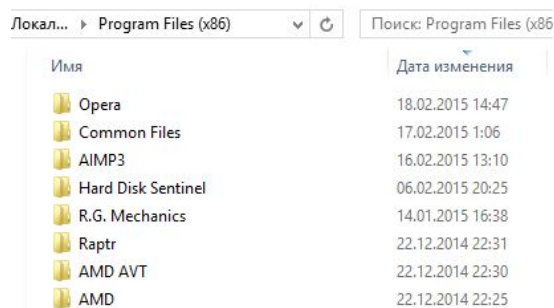


Рисунок 5.17 – Директория Program Files (x86)

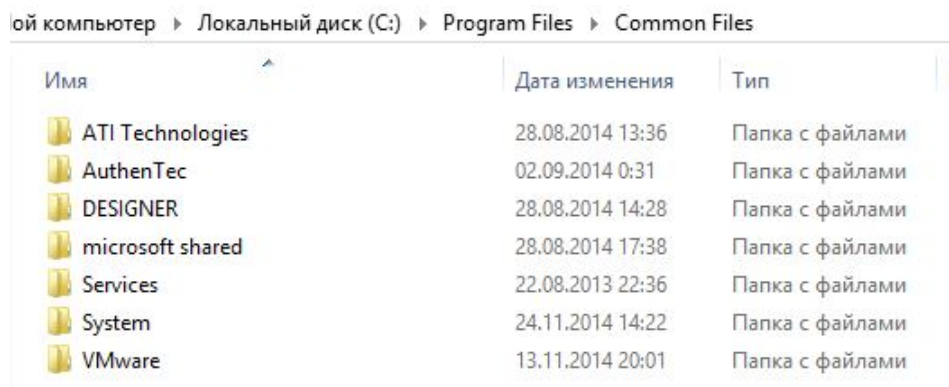


Рисунок 5.18 – Директория Program Files\Common Files

ой компьютер > Локальный диск (C:) > Program Files (x86) > Common Files		
Имя	Дата изменения	Тип
Adobe	28.08.2014 17:07	Папка с файлами
ATI Technologies	22.12.2014 22:30	Папка с файлами
AuthenTec	02.09.2014 0:31	Папка с файлами
Designer	28.08.2014 16:07	Папка с файлами
Merge Modules	28.08.2014 16:32	Папка с файлами
Microsoft	28.08.2014 16:17	Папка с файлами
Microsoft Shared	17.10.2014 20:17	Папка с файлами
Services	22.08.2013 22:36	Папка с файлами
System	24.11.2014 14:20	Папка с файлами
VMware	13.11.2014 20:00	Папка с файлами

Рисунок 5.19 – Директория Program Files (x86)\Common Files

Далее расположение остаточных файлов зависит от ОС.

Для Windows 7, 8:

- 1) C:\ProgramData (рис. 5.20);
- 2) C:\Users\Имя пользователя\AppData\Local (рис. 5.21);
- 3) C:\Users\Имя пользователя\AppData\Roaming (рис. 5.22);
- 4) C:\Users\Default\AppData\Local;
- 5) C:\Users\Default\AppData\Roaming.

Для Windows XP:

- 1) C:\Documents and Settings\Имя пользователя\Application Data (рис. 5.23);
- 2) C:\Documents and Settings\Имя пользователя\LocalSettings\ApplicationData;

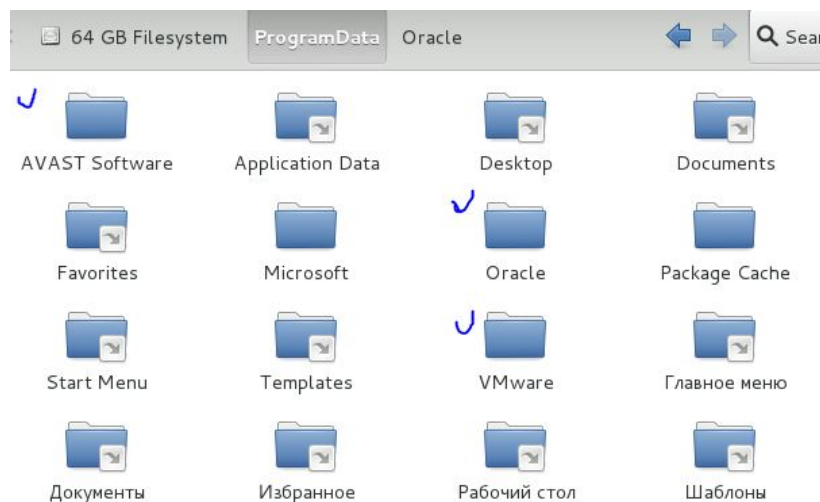


Рисунок 5.20 – Директория ProgramData

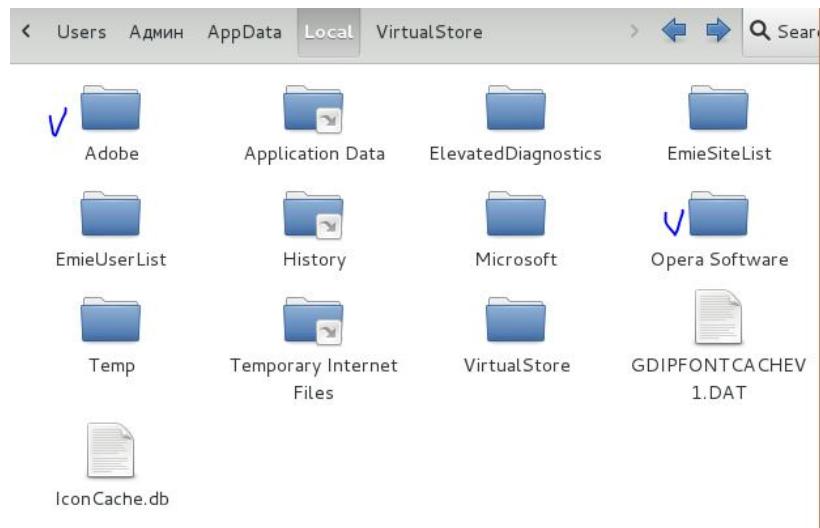


Рисунок 5.21 – Директория Local



Рисунок 5.22 – Директория Roaming

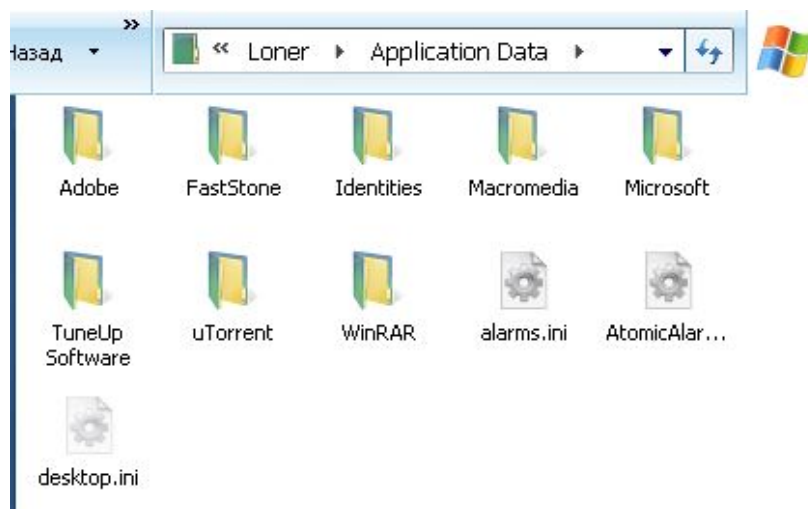


Рисунок 5.23 – Директория Application Data

5.3.2 Список эталонных каталогов для Windows XP x32

C:\Documents and Settings\admin\Application Data\Identities
C:\Documents and Settings\admin\Application Data\Microsoft
C:\Documents and Settings\admin\Local Settings\Application Data\Microsoft
C:\Documents and Settings\All Users\Application Data\Microsoft
C:\Documents and Settings\Default User\Application Data\Microsoft
C:\Documents and Settings\Default User\Local Settings\Application Data\Microsoft
C:\Program Files\Common Files
C:\Program Files\ComPlus Applications
C:\Program Files\Internet Explorer
C:\Program Files\Messenger
C:\Program Files\microsoft frontpage
C:\Program Files\Movie Maker
C:\Program Files\MSN Gaming Zone
C:\Program Files\NetMeeting
C:\Program Files\Online Services
C:\Program Files\Outlook Express
C:\Program Files\Uninstall Information
C:\Program Files\Windows Media Player
C:\Program Files\Windows NT
C:\Program Files\WindowsUpdate
C:\Program Files\xerox
C:\Program Files\Common Files\Microsoft Shared
C:\Program Files\Common Files\MSSoap
C:\Program Files\Common Files\ODBC
C:\Program Files\Common Files\Services
C:\Program Files\Common Files\SpeechEngines
C:\Program Files\Common Files\System

5.3.3 Список эталонных каталогов для Windows XP x64

C:\Documents and Settings\Administrator\Application Data\Identities
C:\Documents and Settings\Administrator\Application Data\Microsoft
C:\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft
C:\Documents and Settings\All Users\Application Data\Microsoft
C:\Documents and Settings\Default User\Application Data\Microsoft
C:\Documents and Settings\Default User\Local Settings\Application Data\Microsoft
C:\Program Files\Common Files
C:\Program Files\ComPlus Applications
C:\Program Files\Internet Explorer
C:\Program Files\Messenger

C:\Program Files\Online Services
 C:\Program Files\Outlook Express
 C:\Program Files\Windows NT
 C:\Program Files\Common Files\Microsoft Shared
 C:\Program Files\Common Files\ODBC
 C:\Program Files\Common Files\Services
 C:\Program Files\Common Files\SpeechEngines
 C:\Program Files\Common Files\System
 C:\Program Files (x86)\Common Files
 C:\Program Files (x86)\Internet Explorer
 C:\Program Files (x86)\microsoft shared
 C:\Program Files (x86)\Movie Maker
 C:\Program Files (x86)\MSN
 C:\Program Files (x86)\MSN Gaming Zone
 C:\Program Files (x86)\NetMeeting
 C:\Program Files (x86)\Outlook Express
 C:\Program Files (x86)\speechengines
 C:\Program Files (x86)\system
 C:\Program Files (x86)\Uninstall Information
 C:\Program Files (x86)\Windows Media Player
 C:\Program Files (x86)\Windows Media Player[Strings]
 C:\Program Files (x86)\Windows NT
 C:\Program Files (x86)\Common Files\Microsoft Shared
 C:\Program Files (x86)\Common Files\ODBC
 C:\Program Files (x86)\Common Files\Services
 C:\Program Files (x86)\Common Files\SpeechEngines
 C:\Program Files (x86)\Common Files\System

5.3.4 Список эталонных каталогов для Windows 7 x32

C:\Program Files\Common Files
 C:\Program Files\DVD Maker
 C:\Program Files\Internet Explorer
 C:\Program Files\Microsoft Games
 C:\Program Files\MSBuild
 C:\Program Files\Reference Assemblies
 C:\Program Files\Uninstall Information
 C:\Program Files\Windows Defender
 C:\Program Files\Windows Journal
 C:\Program Files\Windows Mail
 C:\Program Files\Windows Media Player

C:\Program Files\Windows NT
 C:\Program Files\Windows Photo Viewer
 C:\Program Files\Windows Portable Devices
 C:\Program Files\Windows Sidebar
 C:\Program Files\Common Files\microsoft shared
 C:\Program Files\Common Files\Services
 C:\Program Files\Common Files\SpeechEngines
 C:\Program Files\Common Files\System
 C:\ProgramData\Application Data
 C:\ProgramData\Desktop
 C:\ProgramData\Documents
 C:\ProgramData\Favorites
 C:\ProgramData\Microsoft
 C:\ProgramData\Start Menu
 C:\ProgramData\Templates
 C:\ProgramData\Главное меню
 C:\ProgramData\Документы
 C:\ProgramData\Избранное
 C:\ProgramData\Рабочий стол
 C:\ProgramData\Шаблоны
 C:\Users\admin\AppData\Local\Application Data
 C:\Users\admin\AppData\Local\History
 C:\Users\admin\AppData\Local\Microsoft
 C:\Users\admin\AppData\Local\Temp
 C:\Users\admin\AppData\Local\Temporary Internet Files
 C:\Users\admin\AppData\Local\VirtualStore
 C:\Users\admin\AppData\Roaming\Identities
 C:\Users\admin\AppData\Roaming\Media Center Programs
 C:\Users\admin\AppData\Roaming\Microsoft
 C:\Users\Default\AppData\Local\Application Data
 C:\Users\Default\AppData\Local\History
 C:\Users\Default\AppData\Local\Microsoft
 C:\Users\Default\AppData\Local\Temp
 C:\Users\Default\AppData\Local\Temporary Internet Files
 C:\Users\Default\AppData\Roaming\Media Center Programs
 C:\Users\Default\AppData\Roaming\Microsoft

5.3.5 Список эталонных каталогов для Windows 7 x64

C:\Program Files\Common Files
 C:\Program Files\DVD Maker

C:\Program Files\Internet Explorer
C:\Program Files\Microsoft Games
C:\Program Files\MSBuild
C:\Program Files\Reference Assemblies
C:\Program Files\Uninstall Information
C:\Program Files\Windows Defender
C:\Program Files\Windows Journal
C:\Program Files\Windows Mail
C:\Program Files\Windows Media Player
C:\Program Files\Windows NT
C:\Program Files\Windows Photo Viewer
C:\Program Files\Windows Portable Devices
C:\Program Files\Windows Sidebar
C:\Program Files\Common Files\Microsoft Shared
C:\Program Files\Common Files\Services
C:\Program Files\Common Files\SpeechEngines
C:\Program Files\Common Files\System
C:\Program Files (x86)\Common Files
C:\Program Files (x86)\Internet Explorer
C:\Program Files (x86)\MSBuild
C:\Program Files (x86)\Assemblies
C:\Program Files (x86)\Uninstall Information
C:\Program Files (x86)\Windows Defender
C:\Program Files (x86)\Windows Mail
C:\Program Files (x86)\Windows Media Player
C:\Program Files (x86)\Windows NT
C:\Program Files (x86)\Windows Photo Viewer
C:\Program Files (x86)\Windows Portable Devices
C:\Program Files (x86)\Windows Sidebar
C:\Program Files (x86)\Common Files\microsoft shared
C:\Program Files (x86)\Common Files\Services
C:\Program Files (x86)\Common Files\SpeechEngines
C:\Program Files (x86)\Common Files\System
C:\ProgramData\Application Data
C:\ProgramData\Desktop
C:\ProgramData\Documents
C:\ProgramData\Favorites
C:\ProgramData\Microsoft
C:\ProgramData\Start Menu
C:\ProgramData\Templates
C:\ProgramData\Главное меню
C:\ProgramData\Документы

C:\ProgramData\Избранное
 C:\ProgramData\Рабочий стол
 C:\ProgramData\Шаблоны
 C:\Users\admin\AppData\Local\Application Data
 C:\Users\admin\AppData\Local\History
 C:\Users\admin\AppData\Local\Microsoft
 C:\Users\admin\AppData\Local\Temp
 C:\Users\admin\AppData\Local\Temporary Internet Files
 C:\Users\admin\AppData\Local\VirtualStore
 C:\Users\admin\AppData\Roaming\Adobe
 C:\Users\admin\AppData\Roaming\Identities
 C:\Users\admin\AppData\Roaming\Media Center Programs
 C:\Users\admin\AppData\Roaming\Microsoft
 C:\Users\Default\AppData\Local\Application Data
 C:\Users\Default\AppData\Local\History
 C:\Users\Default\AppData\Local\Microsoft
 C:\Users\Default\AppData\Local\Temp
 C:\Users\Default\AppData\Local\Temporary Internet Files
 C:\Users\Default\AppData\Roaming\Media Center Programs
 C:\Users\Default\AppData\Roaming\Microsoft

5.3.6 Список эталонных каталогов для Windows 8 x32

C:\Program Files\Common Files
 C:\Program Files\Embedded Lockdown Manager
 C:\Program Files\Internet Explorer
 C:\Program Files\Microsoft.NET
 C:\Program Files\MSBuild
 C:\Program Files\Reference Assemblies
 C:\Program Files\Uninstall Information
 C:\Program Files\Windows Defender
 C:\Program Files\Windows Journal
 C:\Program Files\Windows Mail
 C:\Program Files\Windows Media Player
 C:\Program Files\Windows Multimedia Platform
 C:\Program Files\Windows NT
 C:\Program Files\Windows Photo Viewer
 C:\Program Files\Windows Portable Devices
 C:\Program Files\Windows Sidebar
 C:\Program Files\WindowsApps
 C:\Program Files\WindowsPowerShell

C:\Program Files\Common Files\microsoft shared
 C:\Program Files\Common Files\Services
 C:\Program Files\Common Files\System
 C:\ProgramData\Application Data
 C:\ProgramData\Desktop
 C:\ProgramData\Documents
 C:\ProgramData\Microsoft
 C:\ProgramData\regid.1991-06.com.microsoft
 C:\ProgramData\Start Menu
 C:\ProgramData\Templates
 C:\ProgramData\главное меню
 C:\ProgramData\Документы
 C:\ProgramData\Рабочий стол
 C:\ProgramData\Шаблоны
 C:\Users\admin\AppData\Local\Application Data
 C:\Users\admin\AppData\Local\EmieBrowserModeList
 C:\Users\admin\AppData\Local\EmieSiteList
 C:\Users\admin\AppData\Local\EmieUserList
 C:\Users\admin\AppData\Local\History
 C:\Users\admin\AppData\Local\Microsoft
 C:\Users\admin\AppData\Local\Packages
 C:\Users\admin\AppData\Local\Temp
 C:\Users\admin\AppData\Local\Temporary Internet Files
 C:\Users\admin\AppData\Local\VirtualStore
 C:\Users\admin\AppData\Roaming\Adobe
 C:\Users\admin\AppData\Roaming\Microsoft
 C:\Users\Default\AppData\Local\Application Data
 C:\Users\Default\AppData\Local\History
 C:\Users\Default\AppData\Local\Microsoft
 C:\Users\Default\AppData\Local\Temp
 C:\Users\Default\AppData\Local\Temporary Internet Files
 C:\Users\Default\AppData\Roaming\Microsoft

5.3.7 Список эталонных каталогов для Windows 8 x64

C:\Program Files\Common Files
 C:\Program Files\Internet Explorer
 C:\Program Files\MSBuild
 C:\Program Files\Reference Assemblies
 C:\Program Files\Uninstall Information
 C:\Program Files\Windows Defender

C:\Program Files\Windows Journal
C:\Program Files\Windows Mail
C:\Program Files\Windows Media Player
C:\Program Files\Windows Multimedia Platform
C:\Program Files\Windows NT
C:\Program Files\Windows Photo Viewer
C:\Program Files\Windows Portable Devices
C:\Program Files\Windows Sidebar
C:\Program Files\WindowsApps
C:\Program Files\WindowsPowerShell
C:\Program Files\Common Files\microsoft shared
C:\Program Files\Common Files\Services
C:\Program Files\Common Files\System
C:\Program Files (x86)\Common Files
C:\Program Files (x86)\Internet Explorer
C:\Program Files (x86)\Microsoft.NET
C:\Program Files (x86)\MSBuild
C:\Program Files (x86)\Reference Assemblies
C:\Program Files (x86)\Windows Defender
C:\Program Files (x86)\Windows Mail
C:\Program Files (x86)\Windows Media Player
C:\Program Files (x86)\Windows Multimedia Platform
C:\Program Files (x86)\Windows NT
C:\Program Files (x86)\Windows Photo Viewer
C:\Program Files (x86)\Windows Portable Devices
C:\Program Files (x86)\Windows Sidebar
C:\Program Files (x86)\WindowsPowerShell
C:\Program Files (x86)\Common Files\Microsoft Shared
C:\Program Files (x86)\Common Files\Services
C:\Program Files (x86)\Common Files\System
C:\ProgramData\Application Data
C:\ProgramData\Desktop
C:\ProgramData\Documents
C:\ProgramData\Microsoft
C:\ProgramData\regid.1991-06.com.microsoft
C:\ProgramData\Start Menu
C:\ProgramData\Templates
C:\ProgramData\главное меню
C:\ProgramData\Документы
C:\ProgramData\Рабочий стол
C:\ProgramData\Шаблоны
C:\Users\agaerg\AppData\Local\Application Data

C:\Users\agaerg\AppData\Local\History
 C:\Users\agaerg\AppData\Local\Microsoft
 C:\Users\agaerg\AppData\Local\Packages
 C:\Users\agaerg\AppData\Local\Temp
 C:\Users\agaerg\AppData\Local\Temporary Internet Files
 C:\Users\agaerg\AppData\Local\VirtualStore
 C:\Users\agaerg\AppData\Roaming\Adobe
 C:\Users\agaerg\AppData\Roaming\Identities
 C:\Users\agaerg\AppData\Roaming\Microsoft
 C:\Users\Default\AppData\Local\Application Data
 C:\Users\Default\AppData\Local\History
 C:\Users\Default\AppData\Local\Microsoft
 C:\Users\Default\AppData\Local\Temp
 C:\Users\Default\AppData\Local\Temporary Internet Files
 C:\Users\Default\AppData\Roaming\Microsoft

5.3.8 Блок-схема алгоритма работы программного модуля SearchProgram

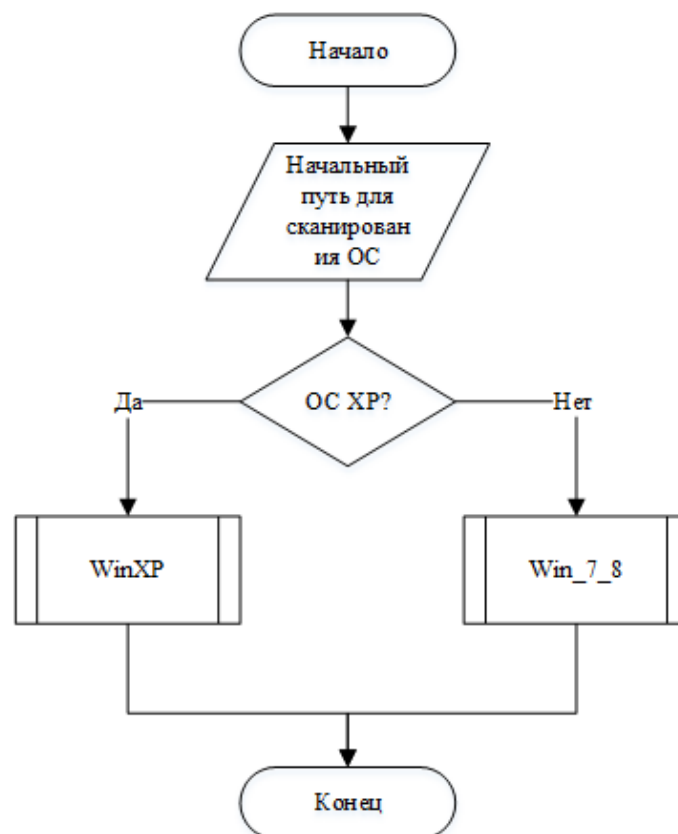


Рисунок 5.24 – Блок-схема основной программы

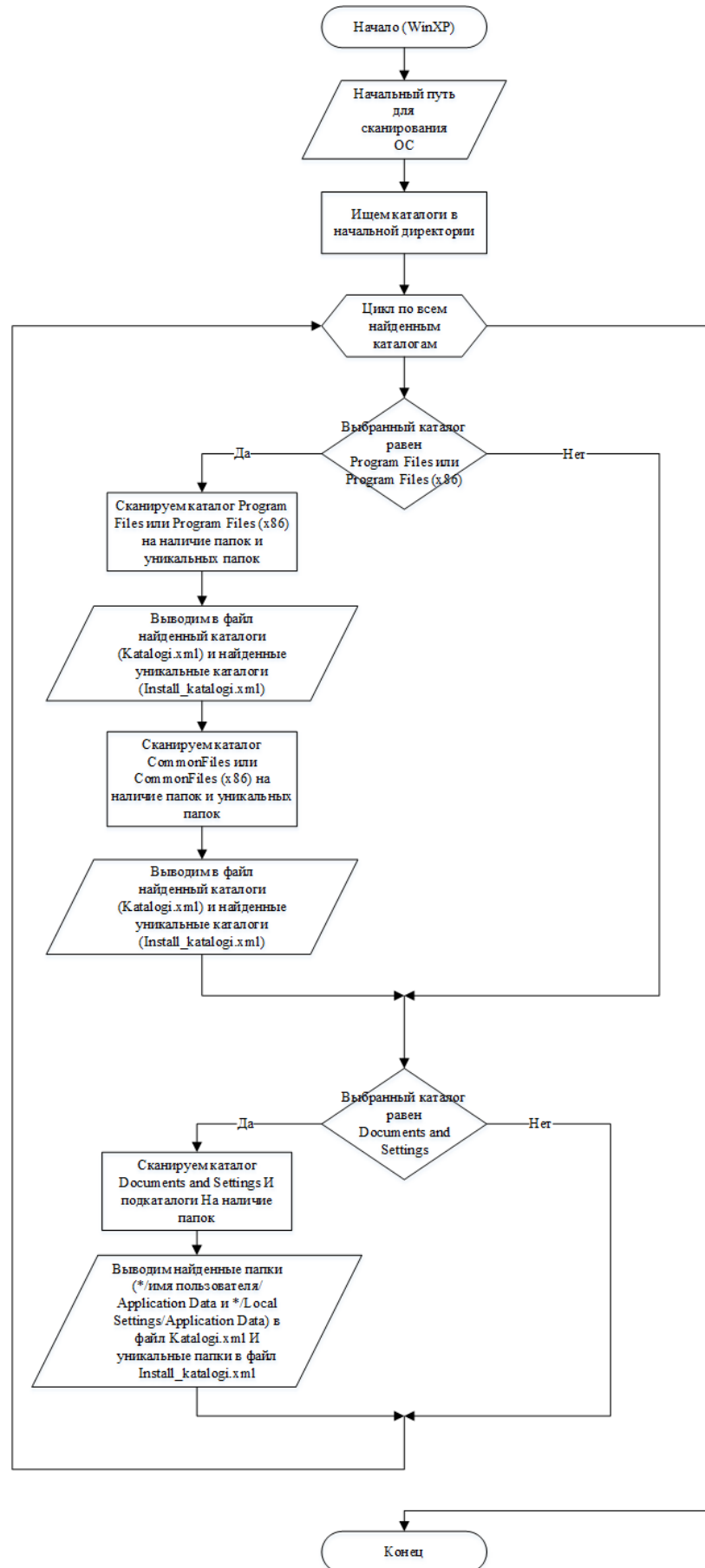


Рисунок 5.25 – Блок-схема функции WinXP

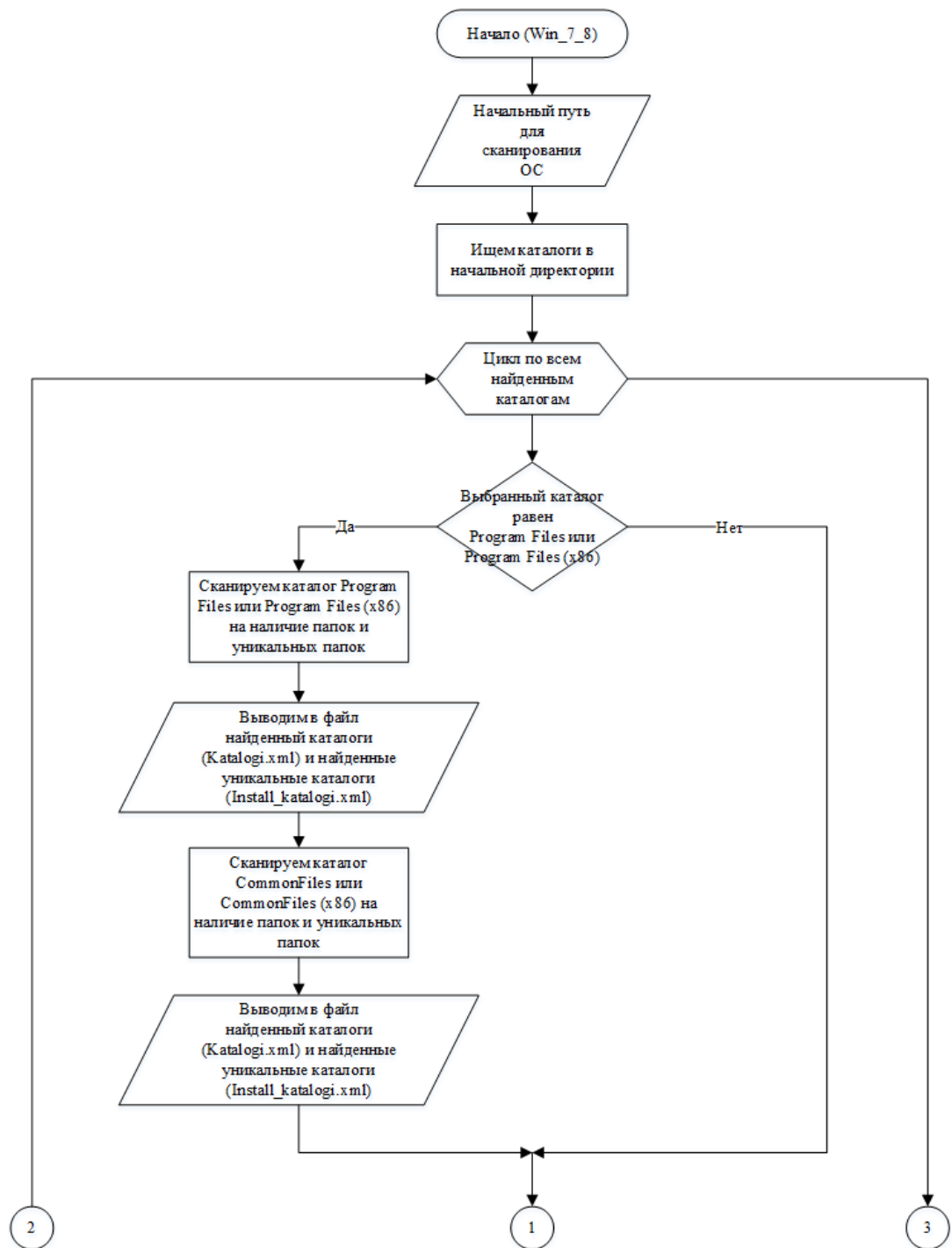


Рисунок 5.26 – Блок-схема функции Win_7_8

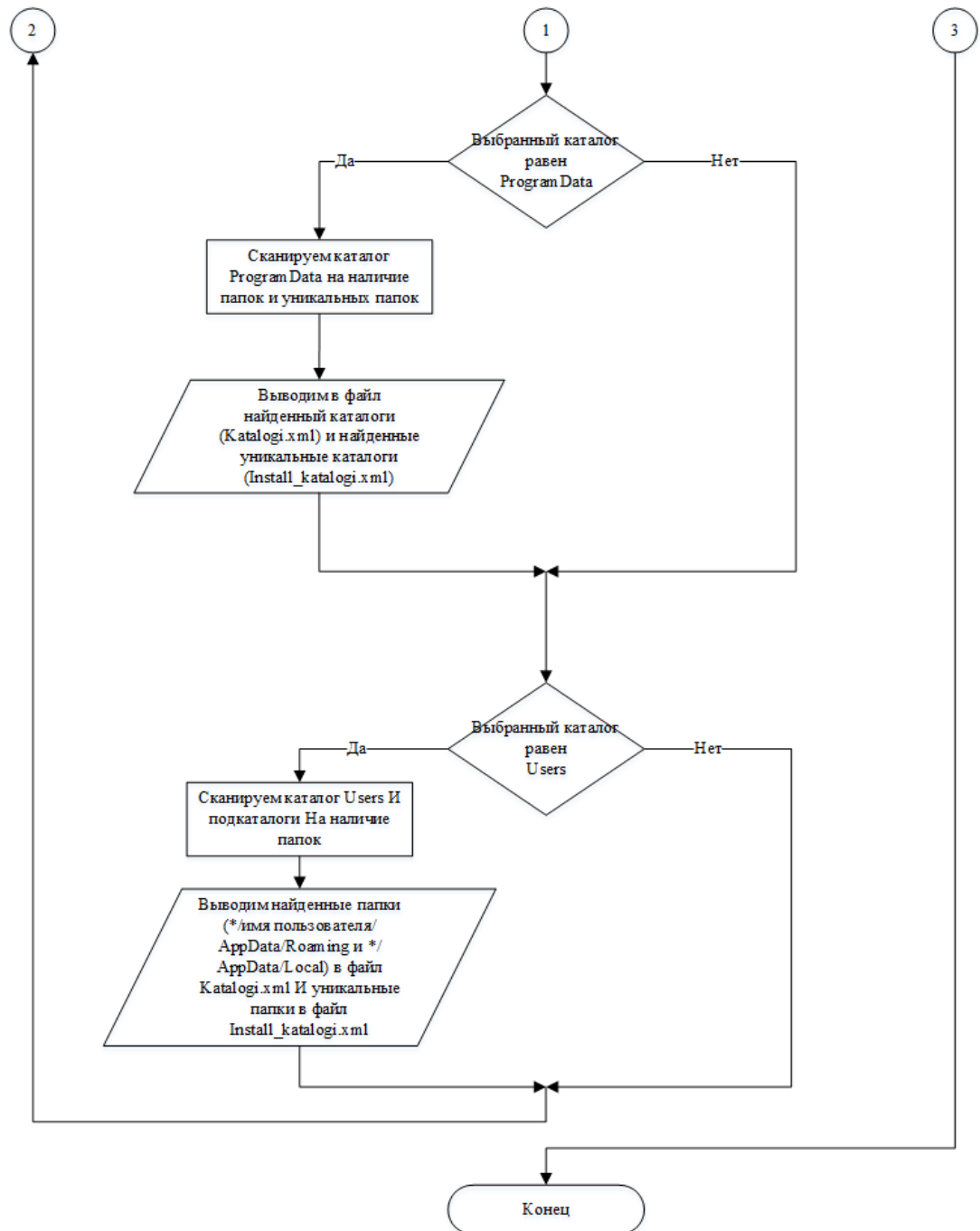


Рисунок 5.27 – Продолжение блок-схемы функции Win_7_8

5.3.9 Описание плагина TaskSearchProgram

Плагин TaskSearchProgram получает начальный путь, с которого он начнет сканировать ОС и путь для сохранения результатов. В зависимости от полученной ОС (на данный момент версия вводится вручную, в дальнейшем это будет автоматизировано) запускается либо функция «WinXP», либо «Win_7_8». Далее происходит сканирование каталогов и подкаталогов, вывод найденных директорий в файл «katalogi.xml». Вдобавок, найденные директории сравниваются с шаблонами, собранными из «чистых» ОС. Те директории, которые не совпали с шаблонами, выводятся в отдельный файл «install_katalogi.xml». Плагин сканирует директории, указанные в предыдущих разделах.

Результаты работы плагина, записанные в выходные XML-файлы, можно увидеть на рисунках 5.28 — 5.33.

```
-<add>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_49fbd9858b14c296c0cb05787a81b9fc</field>
  <field name="application">SearchProgram</field>
  <field name="path">
    ../tmp/test-data/Windows7_Ult/Program Files/Far Manager
  </field>
  <field name="name">Far Manager</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_310a2e689332a7f472668e12675400c3</field>
  <field name="application">SearchProgram</field>
  <field name="path">../tmp/test-data/Windows7_Ult/Program Files/Java</field>
  <field name="name">Java</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_1ccaa35714ad0e2165b30dda274f820e</field>
  <field name="application">SearchProgram</field>
  <field name="path">../tmp/test-data/Windows7_Ult/Program Files/Microsoft IntelliPoint
  </field>
  <field name="name">Microsoft IntelliPoint</field>
</doc>
-<doc>
```

Рисунок 5.28 – Содержимое файла install_katalogi.xml для Windows7

```
-<add>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_a36594184e3edf54053050bb57df3f83</field>
  <field name="application">SearchProgram</field>
  <field name="path">
    ../tmp/test-data/Windows7_Ult/Program Files/Common Files
  </field>
  <field name="name">Common Files</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_49fbd9858b14c296c0cb05787a81b9fc</field>
  <field name="application">SearchProgram</field>
  <field name="path">../tmp/test-data/Windows7_Ult/Program Files/Far Manager
  </field>
  <field name="name">Far Manager</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_0a137205f655f2050bd5e4a679cb3993</field>
  <field name="application">SearchProgram</field>
  <field name="path">../tmp/test-data/Windows7_Ult/Program Files/Internet Explorer
  </field>
  <field name="name">Internet Explorer</field>
</doc>
```

Рисунок 5.29 – Содержимое файла install.xml для Windows7

```
-<add>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_28b55d84e902edb5cf162d659f3fe6eb</field>
  <field name="application">SearchProgram</field>
-<field name="path">
  ../tmp/test-data/Windows8_Pro/Program Files/Far Manager
</field>
  <field name="name">Far Manager</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_438bf22d12cf4e651015ef362d174527</field>
  <field name="application">SearchProgram</field>
  <field name="path">../tmp/test-data/Windows8_Pro/Program Files/Java</field>
  <field name="name">Java</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_1027b68cf6bb404d42ff308ff45ee3d5</field>
  <field name="application">SearchProgram</field>
-<field name="path">
  ../tmp/test-data/Windows8_Pro/Program Files/Microsoft IntelliPoint
</field>
  <field name="name">Microsoft IntelliPoint</field>
</doc>
-<doc>
```

Рисунок 5.30 – Содержимое файла install katalogi.xml для Windows 8

```
- <add>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_7c80a9017a268afe8afcd87bc9ed459e</field>
  <field name="application">SearchProgram</field>
- <field name="path">
  ../tmp/test-data/Windows8_Pro/Program Files/Common Files
  </field>
  <field name="name">Common Files</field>
</doc>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_28b55d84e902edb5cf162d659f3fe6b</field>
  <field name="application">SearchProgram</field>
- <field name="path">
  ../tmp/test-data/Windows8_Pro/Program Files/Far Manager
  </field>
  <field name="name">Far Manager</field>
</doc>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_7fbdd197c94c5da282eafcf142d4f86e</field>
  <field name="application">SearchProgram</field>
- <field name="path">
  ../tmp/test-data/Windows8_Pro/Program Files/Internet Explorer
  </field>
```

Рисунок 5.31 – Содержимое файла install.xml для Windows 8

```
-<add>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_233cbfed5e0c520c38182c54fb1a7a7</field>
  <field name="application">SearchProgram</field>
-<field name="path">
  ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/AllWinnertech
</field>
  <field name="name">AllWinnertech</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_a2d22fd699c9803a550a1a3551bde3f0</field>
  <field name="application">SearchProgram</field>
-<field name="path">
  ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/CollabNet
</field>
  <field name="name">CollabNet</field>
</doc>
-<doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_03d9124ba09af8a07b7421d2637a23e4</field>
  <field name="application">SearchProgram</field>
-<field name="path">
  ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/DIFX
</field>
  <field name="name">DIFX</field>
</doc>
```

Рисунок 5.32 – Содержимое файла install katalogi.xml для Windows XP

```

- <add>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_233cbfe5e0c520c38182c54fb1a7a7</field>
  <field name="application">SearchProgram</field>
  - <field name="path">
    ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/AllWinnertech
  </field>
  <field name="name">AllWinnertech</field>
</doc>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_a2d22fd699c9803a550a1a3551bde3f0</field>
  <field name="application">SearchProgram</field>
  - <field name="path">
    ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/CollabNet
  </field>
  <field name="name">CollabNet</field>
</doc>
- <doc>
  <field name="doc_type">katalog</field>
  <field name="id">katalog_6506938ad46b23668dab8a0ded31f7ae</field>
  <field name="application">SearchProgram</field>
  - <field name="path">
    ../tmp/test-data/WindowsXP_SP3_Pro/Program Files/Common Files
  </field>
  <field name="name">Common Files</field>
</doc>
- <doc>

```

Рисунок 5.33 – Содержимое файла install.xml для Windows XP

5.4 Сбор информации из почтового клиента MS Outlook

В ходе проведения компьютерной экспертизы может возникнуть необходимость проанализировать электронные письма злоумышленника. Подобную информацию можно получить из файлов, сохраняемых программой Outlook на ПК пользователя. Для осуществления данной задачи был разработан программный модуль Outlook.

Почтовая программа (почтовый клиент, клиент электронной почты, мейлер, мейл-клиент) — это ПО, которое устанавливается на компьютер пользователя и предназначено для написания, получения, хранения, отправки электронной почты одного или нескольких пользователей (например, когда имеется несколько учетных записей на компьютере), или нескольких учетных записей пользователя.

Сообщения, синхронизированные с Outlook, имеют самой программе следующий вид (рис 5.34). Задача модуля состоит из поиска данных, отображаемых в программе Outlook в бинарном файле «.dbx» (рис 5.35). В результате работы модуля получаем списки всех тем, дат и тд. сообщения (рис 5.36).

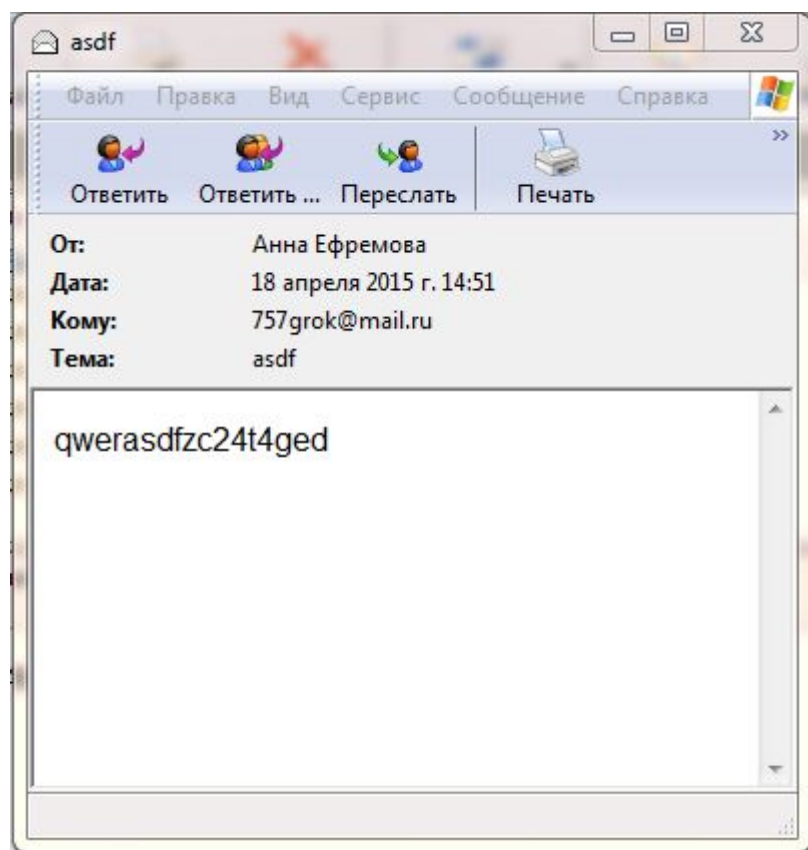


Рисунок 5.34 – Сообщения, синхронизированные с Outlook

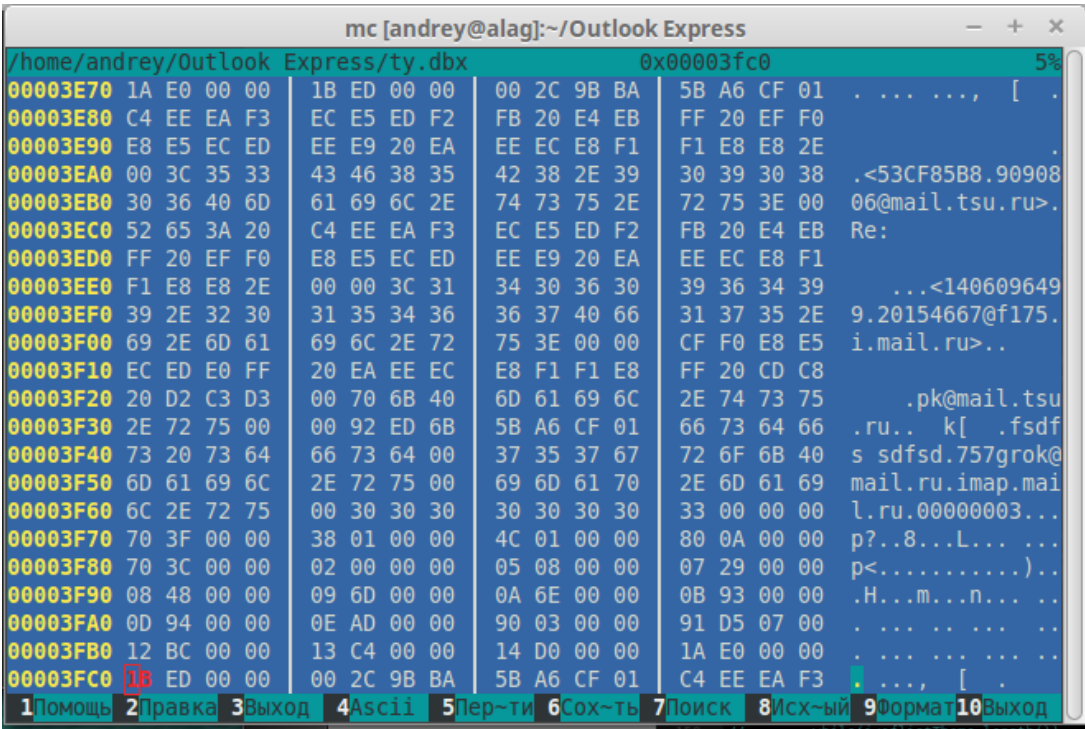


Рисунок 5.35 – Содержание бинарного файла формата «.dbx»

Имя	Значение	Тип
▶ ListDate	<1 элемент>	QStringList
endfile	273108	qint32
▶ file	"/home/andrey/Outlook Express/ty.dbx"	QFile
▶ listAddressOtp	<1 элемент>	QStringList
▶ listText	<1 элемент>	QStringList
▶ listTheme	<1 элемент>	QStringList
▶ listToAddress	<1 элемент>	QStringList
▼ ofListDate	<1 элемент>	QStringList
▶ [0]	"18 Apr 2015 11:51:57 +0300"	QString
▼ oflistAddressOtp	<1 элемент>	QStringList
▶ [0]	"chenneling2013@yandex.ru"	QString
▼ oflistText	<1 элемент>	QStringList
▶ [0]	"\nqwerasdfzc24t4ged"	QString
▼ oflistTheme	<1 элемент>	QStringList
▶ [0]	"\nSubject: asdf"	QString
▼ oflistToAddress	<1 элемент>	QStringList
▶ [0]	"757grok@mail.ru"	QString
smesh	120	qint32
stapseek	273169	qint32
value	""	QString

Рисунок 5.36 – Результат работы модуля

5.4.1 Реализация программного модуля для почтового клиента MS Outlook

Реализация данного программного модуля включала в себя следующие шаги:

- 1) изучение бинарного формата данных «.dbx»;
 - 2) изучение регулярных выражений и библиотек для работы с ними в Qt C++;
 - 3) разработка поиска файлов формата «.dbx» на носителе, на котором установлен Outlook;
 - 4) разработка программы для считывания не всего файла, а только его части, чтобы тем самым уменьшить нагрузку на оперативную память;
 - 5) разработка регулярных выражений для поиска адресата, отправителя, темы, даты и текста сообщения и создание парсера для части информации, извлекаемой из файла;
 - 6) изучение особенностей работы с XML-форматом (языком разметки) и разработка класса для записи данных, полученных из парсера, в XML;
 - 7) изучение системы распределенного контроля версий Git и ее основных возможностей;
 - 8) изучение различных файловых форматов, таких как PST, PAB, MSG, RTF, HTML.
- Трудности, возникшие при написании модуля:

1) В начале написания модуля возникла явная проблема переполнения оперативной памяти из-за добавления всего файла целиком в поток главной программы. Появилась потребность в написании программы, которая делила бы файл на части, запоминала место конца предыдущей части программы и начинала отделять часть такой же длины. Кроме того, данная программа должна была считывать часть файла из любого его места, которую укажут в параметрах, а также преобразовывать последовательность бит в строку юникода. После чего была разработана и написана программа, реализующая данную потребность.

2) Далее стала необходимой разработка парсера (некоего фильтра данных), который бы находил и забирал из выбранной части файла нужные нам последовательности бит. Были изучены основы регулярных выражений, а также синтаксис составления шаблонов для класса QRegex, реализующего работу с регулярными выражениями в QT C++, включая сам класс QRegex. Блок-схема данного парсера представлена на рисунке 5.37.

3) Далее необходимо было разработать алгоритм занесения полученных данных в какой-либо файл для их хранения. В связи с тем, что проект «соех» использует для вывода данных файлы формата XML, был изучен данный формат, а также классы для работы с ним в QT C++. После чего был написан класс «WriteAddress», осуществляющий запись данных из парсера в XML.

Блок-схема исходного модуля после всех преобразований и дополнений приняла следующий вид (рис. 5.38).

5.4.2 Задачи на следующий семестр

В следующем семестре планируется переделать поиск файлов не только в стандартном расположении (месте установки) Outlook, но и в других директориях, за непродолжительное время. Также планируется увеличить скорость работы модуля путем распараллеливания потоков.

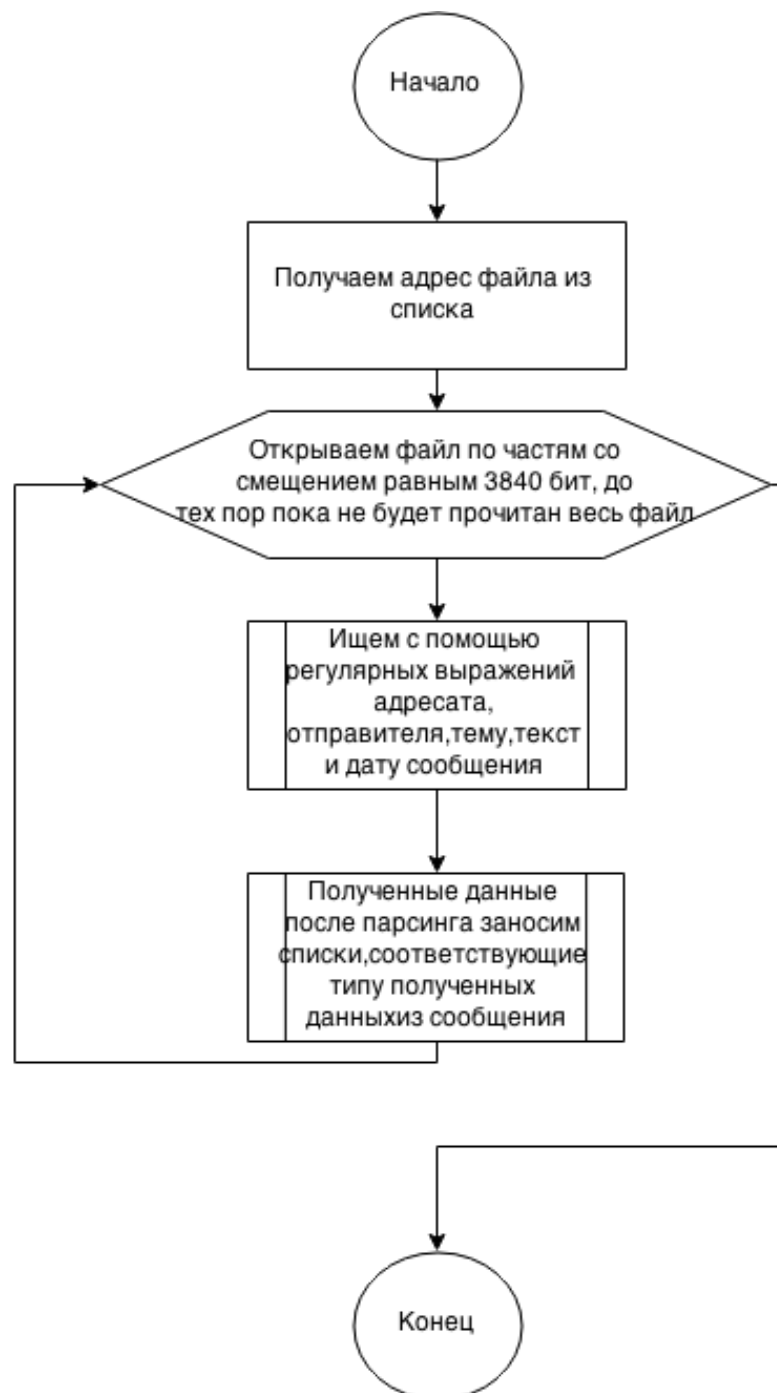


Рисунок 5.37 – Блок-схема парсера для работы с битовыми строками

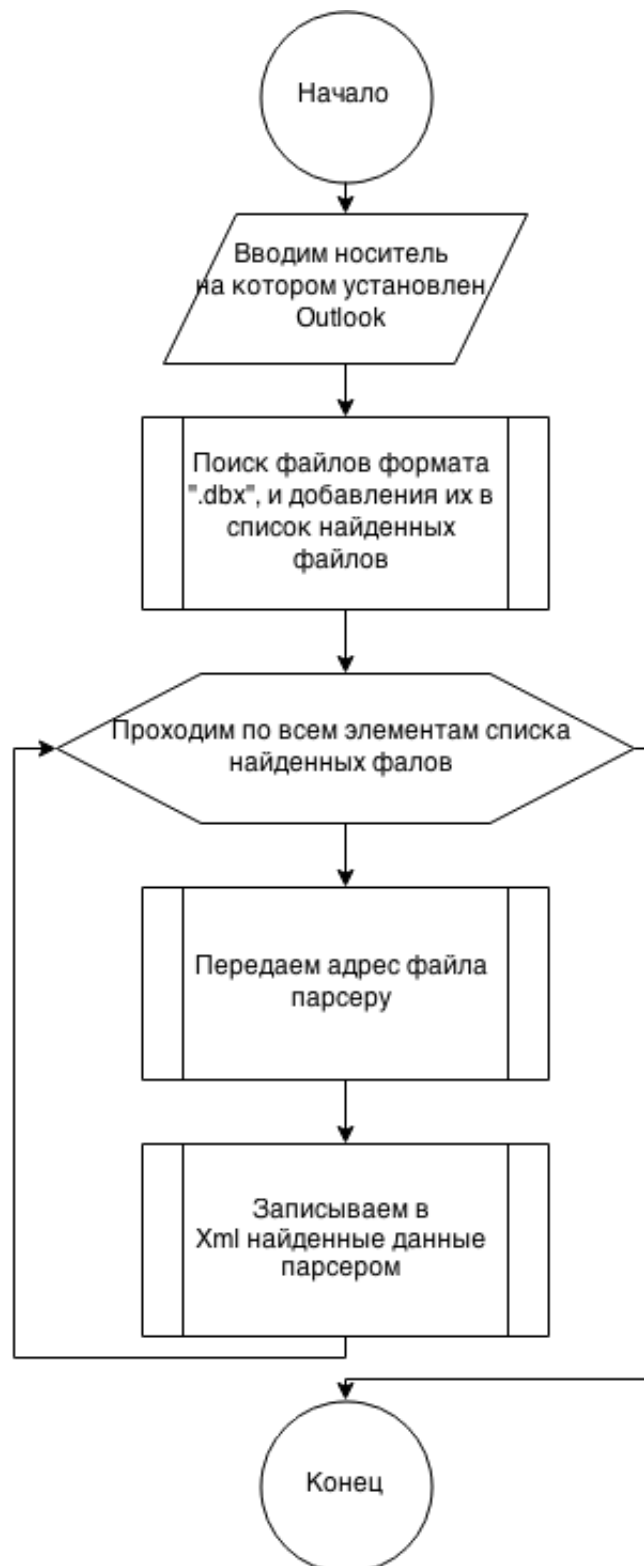


Рисунок 5.38 – Блок-схема исходного модуля Outlook

5.5 Сбор информации из почтового клиента Mozilla Thunderbird

Целью данной работы стало исследование почтового клиента Mozilla Thunderbird, написание программного модуля для сбора сообщений и представления их в формате XML.

5.5.1 Реализация программного модуля

В ходе изучения приложения было выяснено расположение файлов, хранящих почтовые сообщения. Эти данные представлены в таблице 5.3, где:

- profile_name — может быть любым и генерируется самой программой (например, g5bq66yo.default);
- server_name — название сервера входящей почты (например, imap.yandex.com).

Таблица 5.3 – Местоположение и название файлов

Протокол	Путь	Файл с входящими сообщениями	Файл с исходящими сообщениями
imap	C:\Users\User\AppData\Roaming\Thunderbird\Profiles\profile_name\ImapMail\server_name	INBOX	BB4EQgQ,BEAEMAQyBDsENQQ9BD0ESwQ1-
pop3	C:\Users\User\AppData\Roaming\Thunderbird\Profiles\profile_name\Mail\server_name	Inbox	Sent

Для каждого почтового аккаунта, который подключен в Thunderbird, создается своя папка «server_name». Данные указаны для windows 7, 8, 8.1.

Проводник Windows не может определить расширение файлов, но при открытии любым текстовым редактором можно понять, что файлы имеют формат mbox. Mbox представляет собой текстовый файл, в котором хранятся все сообщения почтового ящика. Начало почтового сообщения определяется строкой из 5 символов: словом «From» с последующим пробелом.

Пример сообщения:

```

From
Message-ID: <55600F73.6020804@yandex.ru>
Date: Sat, 23 May 2015 11:26:11 +0600
From: fgfgsr <art0rias@yandex.ru>
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:31.0)
          Gecko/20100101 Thunderbird/31.7.0
MIME-Version: 1.0
To: yuriy94@hotmail.com
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 7bit

Little girl, little girl, Where have you been?
```

В блоке с сообщением хранятся данные о дате, отправителе, получателе, версии почтового клиента, является ли письмо ответом на другое, а также заголовок и текст письма.

5.5.2 Алгоритм работы модуля

После открытия файл mbox разделяется на отдельные сообщения с помощью регулярного выражения «(From \\r\\n)|(From \\n\\r)|From \\r|From \\n». Затем к каждому сообщению применяются регулярные выражения:

- «\\nDate: ([^\\n]*)\\n» — время отправки/приема сообщения;
- «\\nFrom: .*([a-z][\\w\\.]*\\w@[\\w\\.]*\\.\\w*).*\\nUser-Agent:» — кто отправил сообщения;
- «\\nTo: .*([a-z][\\w\\.]*\\w@[\\w\\.]*\\.\\w*).*\\nSubject:» — кто получил сообщение;
- «\\nContent-Transfer-Encoding: 8bit\\s*(\\S.*\\S)\\s*[0-3]\\d\\.\\[01]\\d\\.\\d4 [0-2]\\d:[0-5]\\d, [^\\n]*\\n» — текст сообщения.

Блок-схема алгоритма представлена на рисунке 5.39.

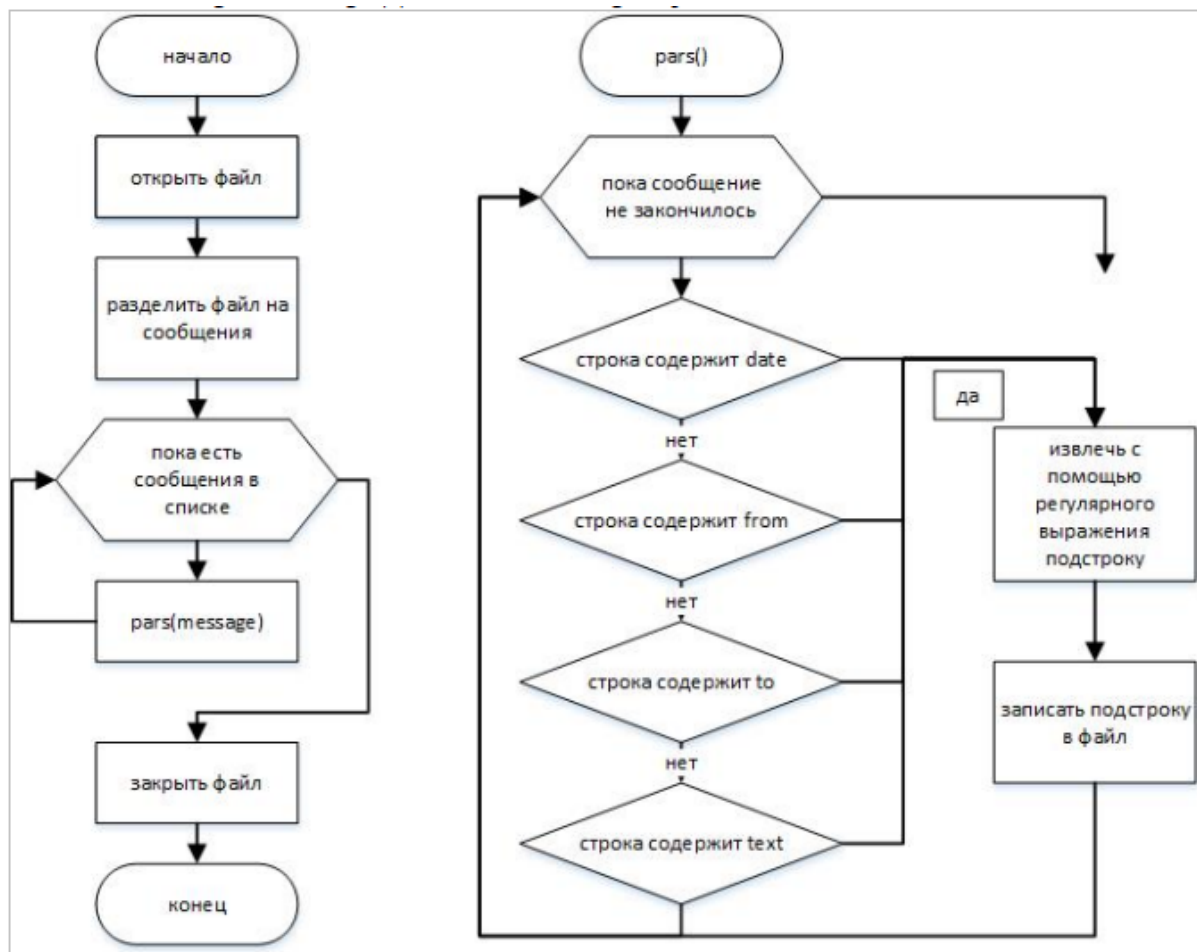


Рисунок 5.39 – Блок-схема алгоритма

5.5.3 Структура XML-файла

Документы XML имеют иерархическую структуру и начинаются с элемента `<add>` — это начальный элемент документа (корень). Далее будут записаны n дочерних элементов `<file>`, где n — количество файлов `mbox`. В каждый элемент `<file>` будут записаны m дочерних элементов `<message>`, где m — количество сообщений в одном файле, в теле которых будет записано нужное нам количество полей для записи информации. В последнюю очередь записывается конечный элемент `</add>`.

Пример файла `message_report.xml` приведен на рисунке 5.40.

Значения полей `date`, `from`, `to`, `text` содержат время и дату, отправителя, получателя, текст сообщения соответственно. Поле `name` содержит полный путь к `mbox` файлу.

```

▼<add>
  ▼<file>
    ▼<field name="name">
      C:\Users\ghost\AppData\Roaming\Thunderbird\Profiles\g5bq66yo.default\ImapMail\imap.yandex.com\&BB4EQgQ,BEAEMAQyBDsENQQ9BD0ESwQ1-
    </field>
    ▼<message>
      <field name="date">Thu, 23 Apr 2015 12:15:55 +0600</field>
      <field name="from">art0rias@yandex.ru</field>
      <field name="to">yuriy9494@gmail.com</field>
      <field name="text">пак за руку цап</field>
    </message>
    ▼<message>
      <field name="date">Thu, 23 Apr 2015 12:46:35 +0600</field>
      <field name="from">art0rias@yandex.ru</field>
      <field name="to">yuriy9494@gmail.com</field>
      <field name="text">I've been to see grandmother Over the green.</field>
    </message>
    ▼<message>
      <field name="date">Thu, 23 Apr 2015 12:47:49 +0600</field>
      <field name="from">art0rias@yandex.ru</field>
      <field name="to">yuriy94@hotmail.com</field>
      <field name="text">What did you say for it? Thank you, Grandam.</field>
    </message>
  </file>
</add>

```

Рисунок 5.40 — Пример XML-файла

5.6 Поиск медиа-файлов и извлечение мета-данных

Целью работы в текущем семестре было написание программного модуля для нахождения медиа-файлов (аудио, видео, изображение) и извлечения мета-данных.

5.6.1 Введение

На базе модуля, сканирующего изображения, была построена основа данного модуля. А именно — проход по файлам с нужным расширением, считывание базовой информации о файле и вывод в XML формат.

Далее необходимо было изучить структуру мета-данных в медиа-файлах и возможность извлекать эти данные. В качестве начальных форматов тэгов были выбраны ID3v1, JFIF и RIFF. Эти типы мета-данных принадлежат к аудио, изображениям и видео соответственно.

Самым простым в реализации оказался тип ID3v1, с которого и было начато исследование.

5.6.2 ID3v1

ID3v1 — формат метаданных, наиболее часто используемый в звуковых файлах в формате MP3. Он содержит данные о названии трека, альбома, имени исполнителя и годе выпуска. Сами эти данные располагаются в конце файла MP3 и имеют структуру, представленную в таблице 5.4).

Блок-схема алгоритма считывания ID3v1 представлена на рисунке 5.41. Как следствие, в XML-файле теперь эти данные имеют следующее представление (рис. 5.42).

Таблица 5.4 – Структура ID3v1

Поле	Размер (в байтах)
Заголовок	3 (всегда равен «TAG»)
Название трека	30
Имя исполнителя	30
Название альбома	30
Год выпуска	4
Комментарии	30
Жанр	1

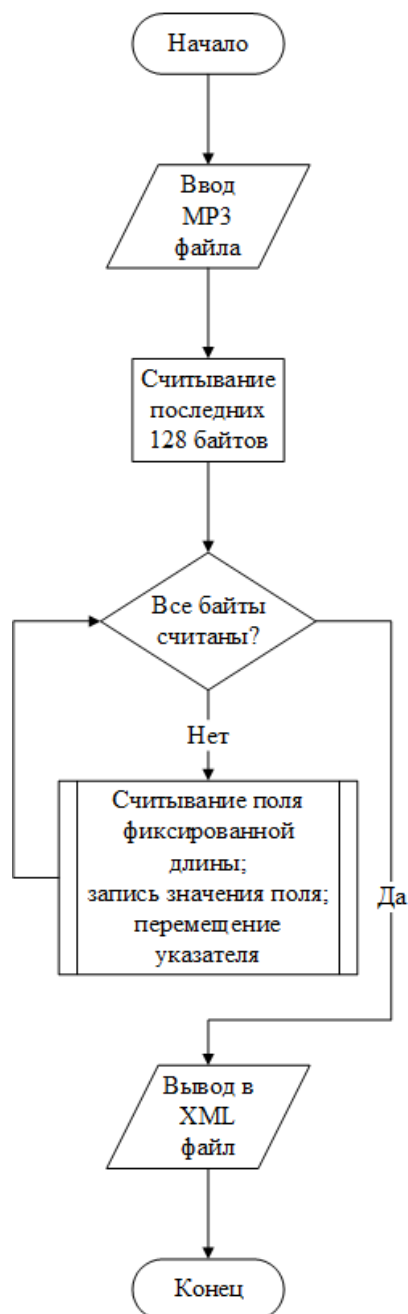


Рисунок 5.41 – Блок-схема алгоритма считывания ID3v1

```

<doc>
  <field name="id">mp3_1cc2b7c5a113d70ff6aeb8d6a82230f3</field>
  <field name="application">media_scanner</field>
  <field name="doc_type">music</field>
  <field name="media_path">D:/University/Testfolder/Cult Of Violence -
    Debris - 04 Sell The King.mp3</field>
  <field name="image_datecreate">Пн май 25 20:34:32 2015</field>
  <field name="image_datemodify">Пн май 25 23:40:31 2015</field>
  - <metadata>
    <field name="title">Sell The King</field>
    <field name="artist">Cult Of Violence</field>
    <field name="album">Debris</field>
    <field name="year">2014</field>
  </metadata>
</doc>

```

Рисунок 5.42 – Результат вывода в XML-файл

5.6.3 JFIF

JPEG File Interchange Format (JFIF) — формат файлов-изображений. Является несколько менее совершенной версией формата EXIF, однако так же позволяет хранить данные об изображении, такие как разрешение изображения, плотность пикселей и данные о миниатюре изображения. Формат представлен в следующем виде (5.5).

Блок-схема алгоритма считывания JFIF представлена на рисунке 5.44, полученный результат в XML-формате — на рисунке 5.43.

Таблица 5.5 – Структура JFIF

Поле	Размер (в байтах)
Маркер APP0	2
Длина	2
Идентификатор	5
Версия	2
Единица измерения плотности	1
Плотность по X	2
Плотность по Y	2
Ширина миниатюры (tw)	1
Высота миниатюры (th)	1
Данные миниатюры	3 * tw * th

```

<doc>
  <field name="id">jpg_7465f0789bcc37a73fbd412e7d9b2672</field>
  <field name="application">media_scanner</field>
  <field name="doc_type">image</field>
  <field name="media_path">D:/University/Testfolder/pic_jpg.jpg</field>
  <field name="image_datecreate">Ср май 20 14:51:12 2015</field>
  <field name="image_datemodify">Вс дек 14 21:01:28 2014</field>
  - <metadata>
    <field name="bits per pixel">8</field>
    <field name="width">1001</field>
    <field name="height">1419</field>
    <field name="density units">no units</field>
    <field name="x resolution">300</field>
    <field name="y resolution">300</field>
  </metadata>
</doc>

```

Рисунок 5.43 – JPG в XML

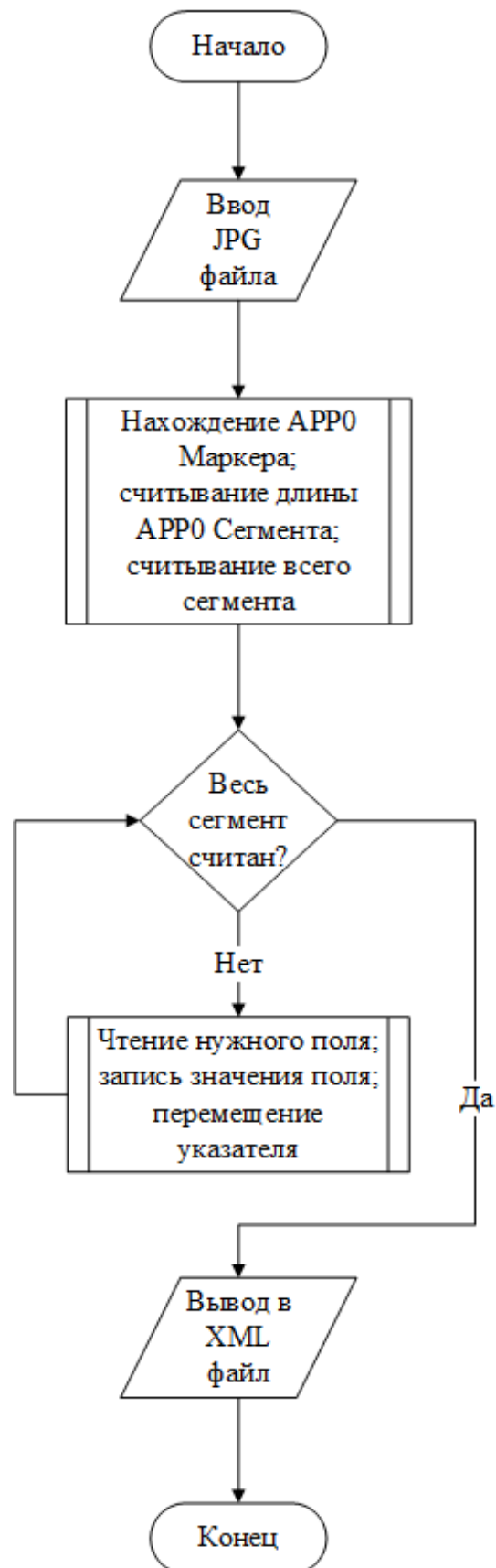


Рисунок 5.44 – Считывание JFIF

5.6.4 RIFF

Resource Interchange File Format (RIFF) — один из форматов файлов-контейнеров для хранения потоковых мультимедиа-данных. Наиболее известными форматами, использующими RIFF в качестве контейнера, является AVI. Основной концепцией формата является chunk — порция данных с заголовком и сигнатурой, указывающей на содержимое chunk'а. В chunk'е strh хранятся общие данные о потоке и определение того, какого типа этот поток (аудио, видео, текст или midi). Соответственно, в его дочернем chunk'е (strf) содержится следующая информация, если это видео (таблица 5.6).

Таблица 5.6 – Структура strf (vids)

Номер байта	Метаданные
0-3	STRF
4-7	Размер
8-11	Ширина
12-15	Высота
16-19	Bit planes
20-23	Кол-во бит на пиксель
24-27	Сжатие бит
28-31	Размер изображения
32-35	Горизонтальное разрешение
36-39	Вертикальное разрешение
40-43	Показатель цвета
44-49	Количество важных цветов

Если же это аудио, то информация будет выглядеть следующим образом (таблица 5.7).

Таблица 5.7 – Структура strf(auds)

Номер байта	Метаданные
0-3	STRF
4-7	Размер
8-9	Компрессор
10-11	Каналы
12-13	Частота дискретизации
14-15	Байты в секунду
16-17	Block align
18-19	Бит на семпл

Алгоритм обработки RIFF — рисунок 5.45, вывод в XML-файл — рисунок 5.46.

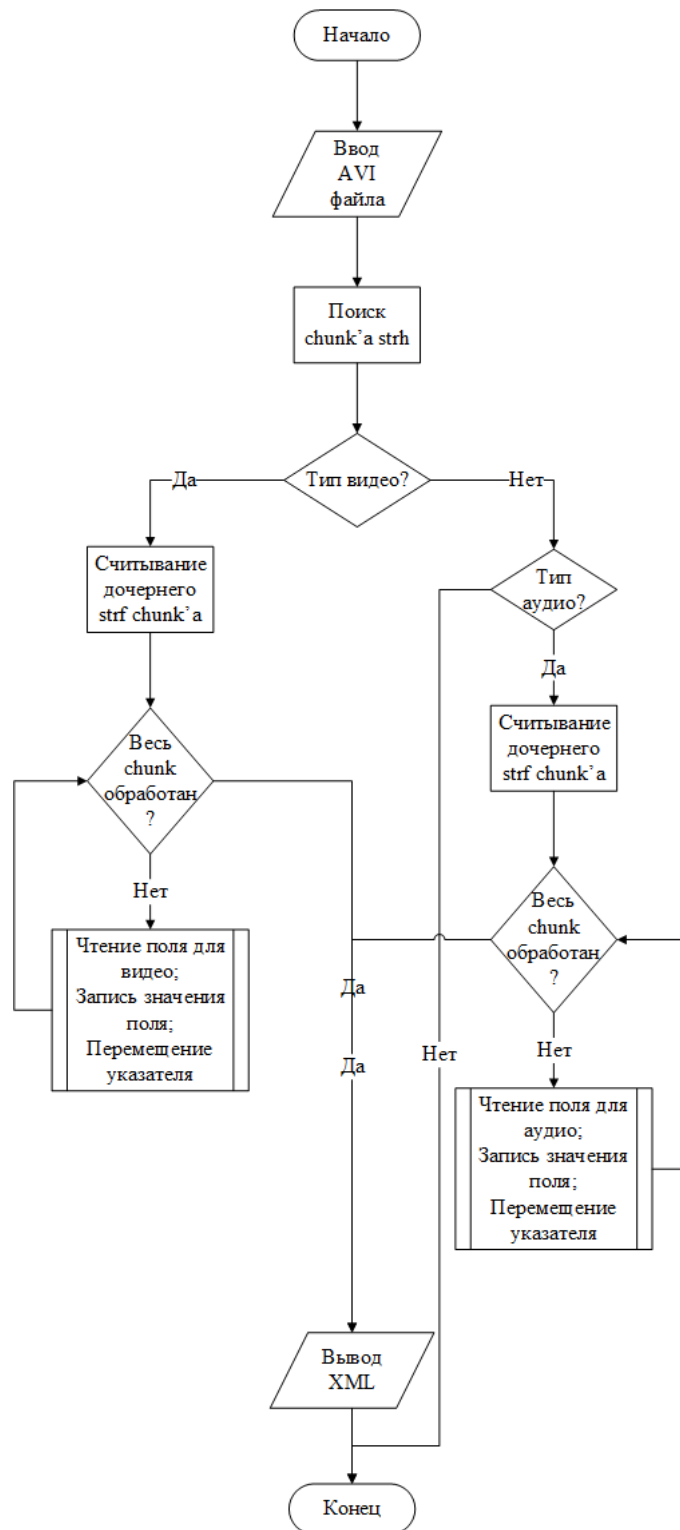


Рисунок 5.45 – Алгоритм обработки RIFF

```

<doc>
  <field name="id">avi_11a5df7d1cf0a6d3304575e992cb0db9</field>
  <field name="application">media_scanner</field>
  <field name="doc_type">video</field>
  <field name="media_path">D:/University/Testfolder/[Shinsen-Subs]
    Gintama_-_03_[0E52C4FF].avi</field>
  <field name="image_datecreate">Вт май 26 04:15:21 2015</field>
  <field name="image_datemodify">Чт янв 2 12:50:20 2015</field>
  - <metadata>
    <field name="video width">640</field>
    <field name="video height">480</field>
    <field name="audio channels">2</field>
    <field name="audio sample rate">48000</field>
  </metadata>
</doc>

```

Рисунок 5.46 – Результат в XML

5.6.5 Задачи на следующий семестр

На данный момент программа является модульной, что позволяет без особого труда добавлять в нее поддержку новых расширений и типов форматов мета-данных. В будущем планируется добавление таких форматов мета-данных, как `mkv`, `exif`, `id3v2`, и поддержка еще большего количества расширений, а также усовершенствование работы программы с уже имеющимися форматами.

В этом семестре реализовано:

- рекурсивный обход директорий;
- чтение формата мета-данных для видео `RIFF`;
- чтение полей тегов формата `JFIF`;
- чтение информации об `mp3` из контейнера `ID3v1`.

5.7 Сбор и анализ информации из реестра ОС MS Windows

Одним из этапов проведения компьютерной экспертизы является анализ установленного программного обеспечения. Общеизвестно, что наличие единого реестра в системе, стало одной из особенностей операционной систем семейства Windows. В большинстве случаев, программы используют его в качестве хранения каких-либо данных.

Реестр Windows представляет собой иерархически построенную базу данных. На этапе загрузки операционная система собирает его из «исходных» файлов, разбросанных по своему дистрибутиву.

Целью работы за этот семестр стояло извлечь полезную информацию из реестра Windows, для достижения которой были выделены следующие задачи:

- 1) исследовать механизм формирования реестра;
- 2) написание программного модуля, способного извлекать информацию из реестра.

5.7.1 Исследование механизма формирования реестра

Семестром ранее был установлен список исходных файлов реестра и было проведено первое знакомство с их структурой.

В ходе исследования одного из файлов было установлено, что он представляет собой «чистый» байтовый массив данных (рис. 5.47). Встал вопрос о структуре данных в нем.

Компания Microsoft разрабатывает операционную систему Windows с конца прошлого века и, несмотря на «закрытость» исходного кода, время от времени выкладывает код отдельных ее модулей. [9] К сожалению, к ним не относится модуль реестра.

Из альтернативных источников можно получить лишь приблизительное представление внутренней структуре (рис. 5.48). [10]

К сожалению, из-за недостатка информации, придется на время отказаться от идеи разбора «сырых» файлов.

00000000	72 65 67 66	67 00 00 00	67 00 00 00	E2 53 12 76	regfg...g... S.v
00000010	F2 95 D0 01	01 00 00 00	03 00 00 00	00 00 00 00
00000020	01 00 00 00	20 00 00 00	00 60 00 00	01 00 00 00`.....
00000030	5C 00 53 00	79 00 73 00	74 00 65 00	6D 00 52 00	\.S.y.s.t.e.m.R.
00000040	6F 00 6F 00	74 00 5C 00	53 00 79 00	73 00 74 00	o.o.t.\.S.y.s.t.
00000050	65 00 6D 00	33 00 32 00	5C 00 43 00	6F 00 6E 00	e.m.3.2.\.C.o.n.
00000060	66 00 69 00	67 00 5C 00	53 00 41 00	4D 00 00 00	f.i.g.\.S.A.M...
00000070	F4 D2 CE 6C	01 6E DE 11	8B ED 00 1E	0B CD 18 24	l.n\$.
00000080	F4 D2 CE 6C	01 6E DE 11	8B ED 00 1E	0B CD 18 24	l.n\$.
00000090	00 00 00 00	F5 D2 CE 6C	01 6E DE 11	8B ED 00 1E l.n . ..
000000A0	0B CD 18 24	72 6D 74 6D	00 00 00 00	00 00 00 00	. .\$.rmtm.....
000000B0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
000000C0	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00

Рисунок 5.47 – Файл SAM в HEX редакторе

5.7.2 Написание программного модуля, способного извлекать информацию из реестра

С помощью встроенной в Windows утилиты regedit был сделан дамп всего древа реестра. Дальнейшая работа будет вестись с этим дампом.

Файл представляет из себя строковый массив с данными 2-х видов:

4.2. Security key

The security key is variable of size and consists of:

offset	size	value	description
0	2	"sk"	Signature
2	2		Unknown
4	4		Previous security key offset The offset value is in bytes and relative from the start of the hive bin data
8	4		Next security key offset The offset value is in bytes and relative from the start of the hive bin data
12	4		Reference count
16	...		NT security descriptor

Рисунок 5.48 – Пример описания одного из «простейших» блоков данных

- 1) описание ветви: [путь_через_узлы];
- 2) описание данных: «название поля» = значение.

```
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.].
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.].
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.\.D.e.s.c.r.i.p.t.i.o.n.].
.
".K.e.y.N.a.m.e.".=.".B.C.D.0.0.0.0.0.0.0.0.".
.
".S.y.s.t.e.m.".=.d.w.o.r.d.:.0.0.0.0.0.0.0.1.
.
".T.r.e.a.t.A.s.S.y.s.t.e.m.".=.d.w.o.r.d.:.0.0.0.0.0.0.0.1.
.
".G.u.i.d.C.a.c.h.e.".=.h.e.x.:.7.0.,.8.7.,.0.d.,.8.d.,.e.7.,.0.b.,.c.f.,.0.1.,.0.c.,.2.7.,.0.0.,.0.0.,.e.8.,.b.d.,.c.1.,.b.c.,.1.e.,.c.f.,.e.0.,.3.5.,.8.0.,.\.
.
.0.1.,.0.0.,.0.0.
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.\.O.b.j.e.c.t.s.].
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.\.O.b.j.e.c.t.s.\.{.0.c.e.4.9.9.1.b.-.e.6.b.3.-.4.b.1.6.-.b.2.3.c.-.5.e.0.d.9.2.5.0.e.5.d.9.}.].
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.\.O.b.j.e.c.t.s.\.{.0.c.e.4.9.9.1.b.-.e.6.b.3.-.4.b.1.6.-.b.2.3.c.-.5.e.0.d.9.2.5.0.e.5.d.9.}.\.D.e.s.c.r.i.p.t.i.o.n.].
.
".T.y.p.e.".=.d.w.o.r.d.:.2.0.1.0.0.0.0.0.
.
.
[.H.K.E.Y._.L.O.C.A.L._.M.A.C.H.I.N.E.\.B.C.D.0.0.0.0.0.0.0.0.\.O.b.j.e.c.t.s.\.{.0.c.e.4.9.9.1.b.-.e.6.b.3.-.4.b.1.6.-.b.2.3.c.-.5.e.0.d.9.2.5.0.e.5.d.9.}.\.E.l.e.m.e.n.t.s.].
```

Рисунок 5.49 – Внутренности .reg файла

Считывание данных происходит построчно, рассмотрим в общем виде, как оно должно



Рисунок 5.50 – Блок-схема алгоритма работы модуля

проходить (рис. 5.51).

Пример работы программы представлен на рисунке 5.52, результат работы программы — на рисунке 5.53.

5.7.3 Результаты работы за семестр

Результаты работы за текущий семестр:

- 1) проведено доисследование структуры исходных файлов реестра;
- 2) написан модуль получения информации из реестра windows. Информация берется из .reg файлов.

Планы на будущее:

- 1) дальнейший поиск описаний структура исходных файлов;
- 2) по возможности, написание модуля извлечения информации из этих файлов.

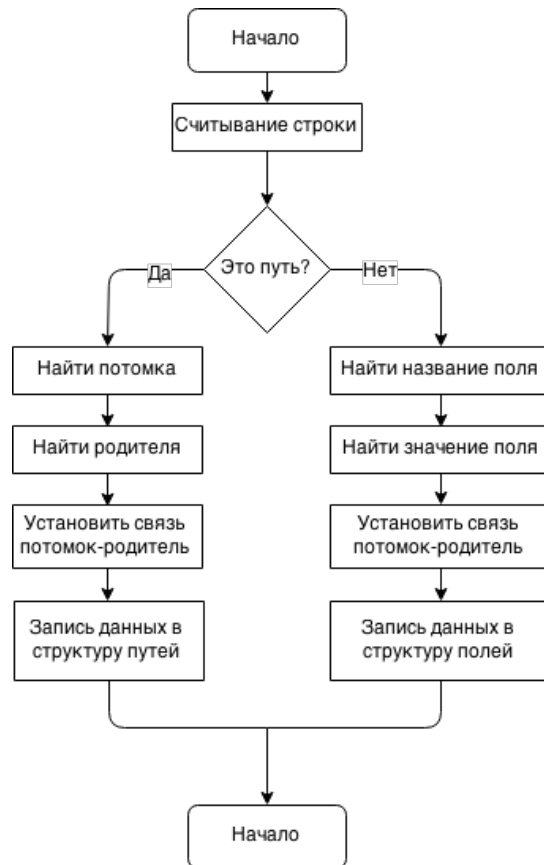


Рисунок 5.51 – Функция разбора одной строки

```

[HKEY_LOCAL_MACHINE]
HKEY_LOCAL_MACHINE
[HKEY_LOCAL_MACHINE\BCD00000000]
-----Current-----
id: 1
content: BCD00000000
children:
-----
-----Parent-----
id: 0
content: HKEY_LOCAL_MACHINE
children: 1
-----
[HKEY_LOCAL_MACHINE\BCD00000000\Description]
-----Current-----
id: 2
content: Description
children:
-----
-----Parent-----
id: 1
content: BCD00000000
children: 2
-----
"KeyName"="BCD00000000"
-----Current_Field----
id: 0
name: KeyName
value: "BCD00000000"
-----
-----Parent-----

```

Рисунок 5.52 – Пример работы программы

```

VIEWING:
<HKEY_LOCAL_MACHINE>
  <BCD00000000>
    <Description>
      KeyName="BCD00000000"
      System=dword:00000001
      TreatAsSystem=dword:00000001
      GuidCache=hex:70,87,0d,8d,e7,0b,cf,01,0c,27,00,00,e8,bd,c1,bc,1e,cf,e0,35,
80,01,00,00
    </Description>
    <Objects>
      <{0ce4991b-e6b3-4b16-b23c-5e0d9250e5d9}>
        <Description>
          Type=dword:20100000
        </Description>
        <Elements>
          <16000020>
            Element=hex:01
          </16000020>
        </Elements>
      </{0ce4991b-e6b3-4b16-b23c-5e0d9250e5d9}>
      <{1afa9c49-16ab-4a5c-901b-212802da9460}>
        <Description>
          Type=dword:20200004
        </Description>
      </{1afa9c49-16ab-4a5c-901b-212802da9460}>
    </Objects>
  </BCD00000000>
</HKEY_LOCAL_MACHINE>

```

Рисунок 5.53 – Результат работы программы

Заключение

В данном семестре нашей группой была выполнена часть работы по созданию автоматизированного программного комплекса для проведения компьютерной экспертизы, проанализированы дальнейшие перспективы и поставлены цели для дальнейшего развития проекта.

Список использованных источников

- 1 Федотов Николай Николаевич. Форензика - компьютерная криминалистика. Юрид. мир, 2007. 432 с.
- 2 Scott Chacon. Pro Git : professional version control. 2011. URL: <http://progit.org/ebook/progit.pdf>.
- 3 С.М. Львовский. Набор и вёрстка в системе L^AT_EX. МЦНМО, 2006. С. 448.
- 4 И. А. Чеботаев, П. З. Котельников. L^AT_EX 2_ε по-русски. Сибирский Хронограф, 2004. 489 с.
- 5 Doxygen : Generate documentation from source code [Электронный ресурс] // www.stack.nl/~dimitri/doxygen/index.html. [2015]. URL: <http://www.stack.nl/~dimitri/doxygen/index.html>.
- 6 Qt Documentation [Электронный ресурс] // qt-project.org/doc. 2013. URL: <http://qt-project.org/doc>.
- 7 Всё о кроссплатформенном программировании - Qt [Электронный ресурс] // doc.crossplatform.ru/qt. 2013. URL: <http://doc.crossplatform.ru/qt>.
- 8 Справочник по XML-стандартам [Электронный ресурс] // [msdn.microsoft.com/ru-ru/library/ms256177\(v=vs.110\).aspx](http://msdn.microsoft.com/ru-ru/library/ms256177(v=vs.110).aspx). URL: [http://msdn.microsoft.com/ru-ru/library/ms256177\(v=vs.110\).aspx](http://msdn.microsoft.com/ru-ru/library/ms256177(v=vs.110).aspx).
- 9 Windows Registry [Электронный ресурс] // forensicswiki.org/wiki/Windows_Registry. [2015]. URL: http://forensicswiki.org/wiki/Windows_Registry.
- 10 Registry [Электронный ресурс] // msdn.microsoft.com/en-us/library/windows/desktop/ms724871. [2015]. URL: <https://msdn.microsoft.com/en-us/library/windows/desktop/ms724871>

Приложение А
(Обязательное)
Компакт-диск

Компакт-диск содержит:

- электронную версию пояснительной записки в форматах *.tex и *.pdf;
- актуальную версию программного комплекса для проведения компьютерной экспертизы;
- тестовые данные для работы с программным комплексом;
- документацию к проекту в html-формате.