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.

2015

РЕФЕРАТ

Курсовая работа содержит 14 страниц, 3 рисунка, 0 таблицы, 0 источников, 1 приложение.

CTF, ATTACK-DEFENSE, СОРЕВНОВАНИЯ, KEVA, ЗАЩИТА ИНФОРМАЦИИ, ФЛАГИ, ПРИЕМКА ФЛАГОВ, SCOREBOARD, GIT, PYTHON, MONGODB, MONGOOSE, ЧЕКЕРЫ, HTML, CSS, JAVASCRIPT, КОМАНДА, API.

Цель работы — создание программного комплекса, предназначенного для проведения соревнований в области информационной безопасности CTF.

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

– инициализация с помощью графического интерфейса;
– интеграция с существующим API и администраторской панелью;
– инициализация и работа проверяющей системы;
– инициализация и работа приемки флагов;
– инициализация и работа таблицы рейтинга;

Результаты работы в данном семестре:

– выбраны язык программирования, база данных с учетом потребностей;
– доработан существующий API и администраторская панель для управления соревнованиями;
– разработана архитектура ядра и организована работа с модулями;
– разработан модуль работы с чекерами;
– разработан модуль приемки флагов;
– разработан модуль таблицы результатов;
– произведено тестирование платформы в форме соревнований.

Пояснительная записка выполнена при помощи системы компьютерной вёрстки LaTeX.

Список исполнителей

Лобанов О.В. – программист, ответственный исполнитель, ответственный за разработку функций сбора информации из реестра.

Шиповской В.В. – программист, ответственный за написание части системы для сбора и обработки информации из браузера Google Chrome.

Серяков А.В. – программист, ответственный за написание части системы для сбора информации из почтового клиента MS Outlook.

Боков И.М. – программист, ответственный за написание части системы для для нахождения медиа-файлов (аудио, видео, изображение) и извлечения мета-данных из них.

Кучер М.В. – программист, ответственный за написание части системы для сбора информации об установленном ПО по остаточным файлам.

Терещенко Ю.А. – программист, ответственный за написание части системы для сбора информации из почтового клиента Mozilla Thunderbird.

Мейта М.В. – документатор, ответсвенный за верстку необходимой документации в системе LaTeX.

Содержание

Введение

CTF (Capture the flag, Захват флага) - это командные соревнования, целью которых является оценка умения участников атаковать и защищать компьютерные системы.

Каждой команде выдается сервер, на котором имеется ряд сервисов, одинаковые у всех. Сервисами являются программы, которые должны быть запущены на все время соревнований. Раз в минуту жюрейская система проверяет работу сервисов, присылает новый флаг, а также проверяет его наличие на сервере. В роли флага выступает случайно сгенерированная строка, определенной длины. Сервисы заведомо имеют уязвимости, через которые можно украсть флаг. Команда должна найти уязвимости в сервисах и закрыть их. Но также она должна эксплуатировать эти уязвимости на сервисах других команд, похищая флаги.

Команда Keva имеет большой опыт в проведении соревнований CTF. Во время проведения межрегиональных межвузовских соревнований в области информационной безопасности SibirCTF 2014, 2015 использовались наработки Екатеринбургской команды HackerDom. Однако их решение не подходит нам по некоторым критически важным параметрам. Поэтому было принято решение написать собственную платформу для проведения соревнований CTF.

Платформа должна отвечать :

– Низкое потребление ресурсов сервера

– Управление игрой через графический интерфейс

– Новая платформа позволит проводить собственные соревнования не используя готовые решения

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 [**?**]. 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 программно-аппаратный комплекс предназначен для проведения игр в области информационной безопасности Capture the flag по типу Attack-Defense.

2 Постановка задачи

На данный семестр были поставлены следующие задачи:

- выбор инструментов разработки платформы;
- разработка архитектуры;
- определение индивидуальных задач для каждого участника проектной группы;
- исследование предметных областей в рамках индивидуальных задач;
- реализация новых программных модулей и доработка уже существующих;

Задачи, связанные с разработкой программного комплекса:

1) разработка ядра платформы;
2) разработка модуля проверяющей системы;
3) разработка модуля сдачи флага;
4) разработка модуля рейтинговой таблицы;
5) доработка модуля администраторской панели;

Задачи, связанные с тестированием программного комплекса:

1) проверка работы ядра;
2) проверка работы модуля проверяющей системы;
3) проверка работы модуля сдачи флага;
4) проверка работы модуля рейтинговой таблицы;
5) проверка работы модуля администраторской панели;

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») [**?**].

При работе над одним проектом команде разработчикоа 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 мержится 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 репозиториев 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.

LATEX — это созданная Лэсли Лэмпортом издательская система на базе TEX'а[**?**]. LATEX позволяет пользователю сконцентрировать свои услия на содержании и структуре текста, не заботясь о деталях его оформления.

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

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

3.3 Язык программирования Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций [**?**].

Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули.

Основными преимуществами языка программирования Python являются большое количество

библиотек, кроссплатформенность, широкие возможности профилирования кода.

Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем. При передаче аргументов в функции Python использует вызов по соиспользованию.

Разработка языка Python была начата в конце 1980-х годов[10] сотрудником голландского института CWI Гвидо ван Россумом.

3.4 База данных MongoDB

Для реализации программного комплекса для проведения соревнований в области информационной безопасноти была использована база данных MongoDB.

MongoDB — документо-ориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Написана на языке C++. [**?**].

Основные возможности MongoDB:

- документо-ориентированное хранение (JSON-подобная схема данных);
- достаточно гибкий язык для формирования запросов;
- динамические запросы;
- поддержка индексов;
- профилирование запросов;
- быстрые обновления «на месте»;
- эффективное хранение двоичных данных больших объёмов, например, фото и видео;
- журналирование операций, модифицирующих данные в базе данных
- поддержка отказоустойчивости и масштабируемости: асинхронная репликация, набор реплик и распределения базы данных на узлы;
- может работать в соответствии с парадигмой MapReduce;
- полнотекстовый поиск, в том числе на русском языке, с поддержкой морфологии.

Архитектура:

СУБД управляет наборами JSON-подобных документов, хранимых в двоичном виде в формате BSON. Хранение и поиск файлов в MongoDB происходит благодаря вызовам протокола GridFS. Подобно другим документо-ориентированным СУБД (CouchDB и др.), MongoDB не является реляционной СУБД. В СУБД:

- нет такого понятия, как «транзакция». Атомарность гарантируется только на уровне целого документа, то есть частичного обновления документа произойти не может;
- Отсутствует понятие «изоляции». Любые данные, которые считываются одним клиентом, могут параллельно изменяться другим клиентом.

В MongoDB реализована асинхронная репликация в конфигурации «ведущий — ведомый» (англ. master — slave), основанная на передаче журнала изменений с ведущего узла на ведомые. Поддерживается автоматическое восстановление в случае выхода из строя ведущего узла. Серверы с запущенным процессом mongod должны образовать кворум, чтобы произошло автоматическое определение нового ведущего узла. Таким образом, если не используется специальный процесс-арбитр (процесс mongod, только участвующий в установке кворума, но не хранящий никаких данных), количество запущенных реплик должно быть нечётным.

3.5 TrackingTime - Система управления проектами и задачами

TrackingTime — это сервис, предоставляющий возможность управлять своими проектами, задачами, персонал.

Основным преимуществом является удобный учет времени ответственного за задачу. Проект декомпозируется на этапы и задачи. Каждая задача назначается на ответственного работника, который в свою очередь при её выполнении устанавливает таймер.

Приложение доступно на всех основных платформах (Windows, Linux, OS X, iOS, Android)

3.6 Flask - микрофреймворк для Python

Flask - это микрофреймворк, написанный на языке программирования Python, основанный на Werkzeug и Jinja 2. Выпускается под BSD лицензией. Особенности фреймворка

- удобная работа с URL;
- богатые возможности шаблонизатора;
- минимальные требования к ресурсам в сравнении с аналогичными фреймворками;

Flask используется в модуле таблицы рейтинга.

4 Технические характеристики

4.1 Требования к аппаратному обеспечению

Необходимо 2 и более компьютера, находящихся в локальной сети.

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 должны быть установлены пакеты: python 3.X, MongoDB.

4.3 Выбор единого формата выходных файлов

Для вывода результата был выбран формат XML-документов, так как с данным форматом лего 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 [**?**]. 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 разарботки был применен видоизменнённый 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.

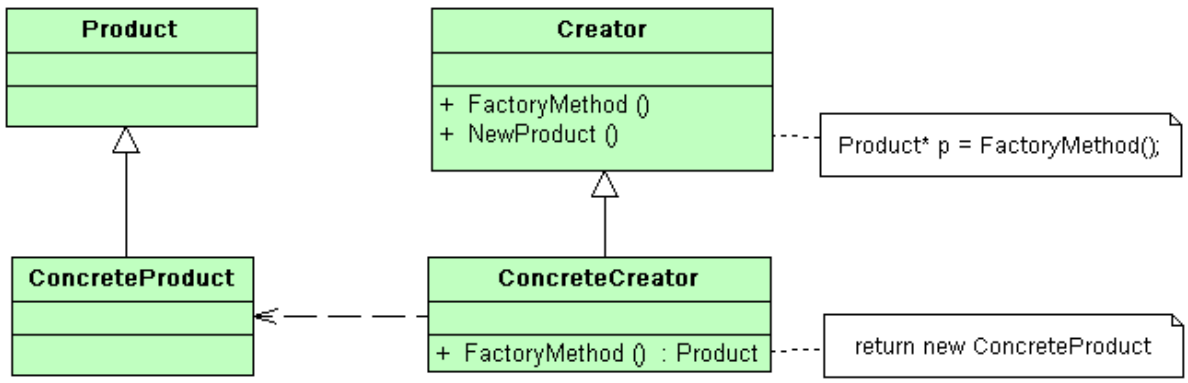

Рисунок 5.1 – Пример организации проекта при использовании шаблона проектирования Factory method

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 клас предназначался для формирования списка настроек, определения операционной системы на смонтированном образе и инстанционировании и накапливание всех необходимых

классов-тасков в очереди тасков. После чего каждый таск из очереди отправлялся на выполнение. Блоксхема работы алгоритма представлена на рисунке 5.2.

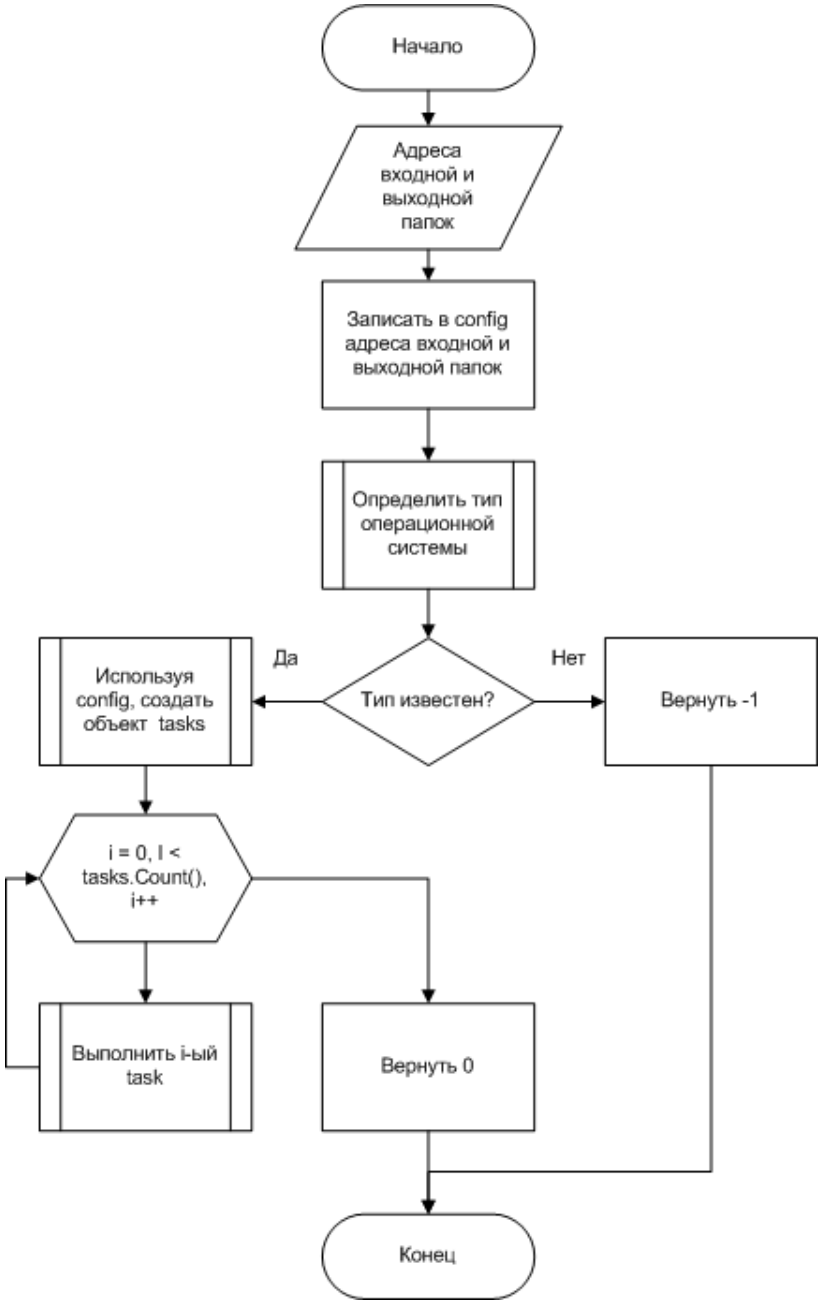

Рисунок 5.2 – Алгоритм работы с образом диска

Каждый класс-таск порождался путем наследования от базового абстрактного класса который имеет 8 методов и 3 атрибута:

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

8) bool execute(const coex::config &config) - запуск таска на выполнение;

9) QString m_strName - хранит имя таска;

10) QString m_strDescription - хранит описание таска;

11) bool m_bDebug - флаг для параметра –debug;

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

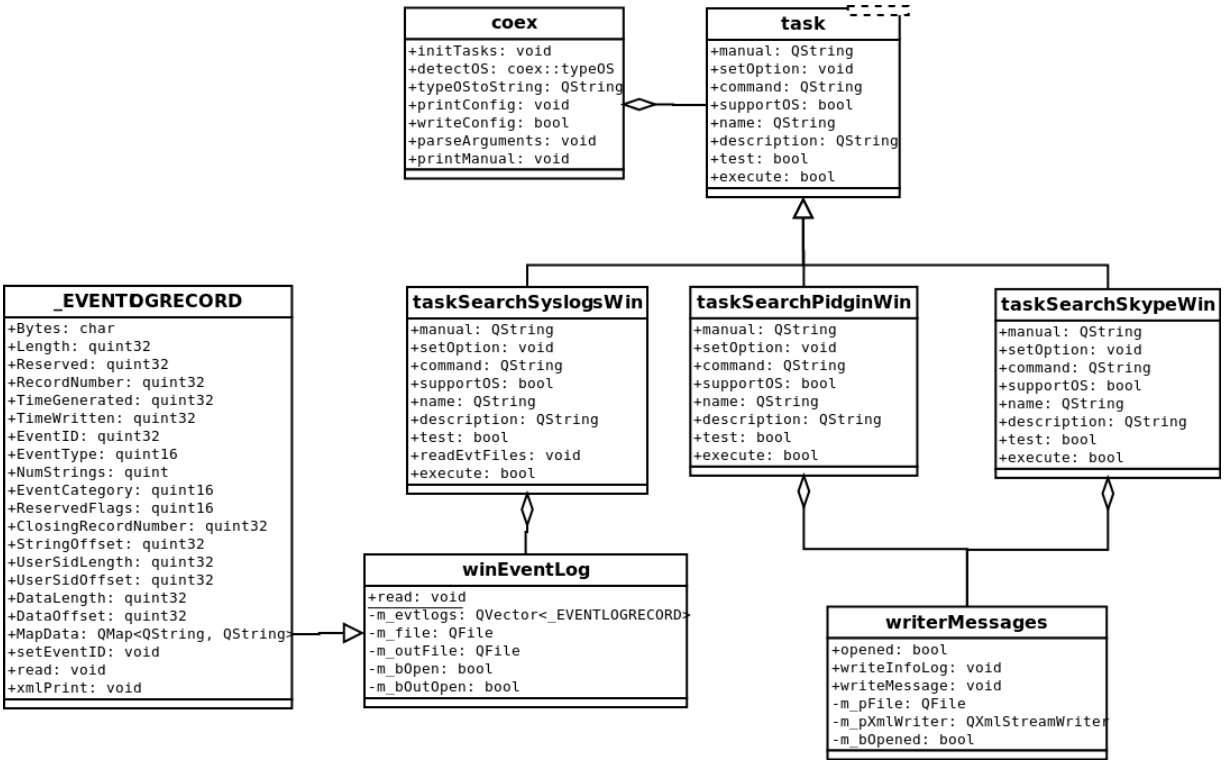

Рисунок 5.3 – UML-диаграмма классов

Классы taskSearchSyslogsWin, taskSearchPidginWin и taskSearchSkypeWin - наследники от класса task являются тасками. Класс winEventLog и _EVENTLOGRECORD предназначины 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 таска

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

QString command() - возвращает команду для инициализации таксса вручную

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

QString name() - возвращает имя данного таска

QString description() - возвращает краткое описание таксса

bool test() - предназначена для проверки работоспособности таска

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

QString m_strName - хранит имя таска

QString m_strDescription - хранит описание таска

bool m_bDebug - флаг для параметра –debug

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.

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.