МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

Кафедра комплексной информационной безопасности электронно-вычислительных систем (КИБЭВС)

К ЗАЩИТЕ ДОПУСТИТЬ

заведующий каф. КИБЭВС

д-р техн. наук, проф.

_____А.А. Шелупанов

«_____»_____2017г.

«ОПРЕДЕЛЕНИЕ АВТОРСТВА ИСХОДНОГО КОДА»

Специалистская работа по направлению 10.05.03 –

Информационная безопасность автоматизированных систем

СОГЛАСОВАНО

Консультант по экономике:
ст. преподаватель каф. КИБЭВС
_____С.В. Глухарева
"_____"_____ 2017г.

Консультант по безопасности жизнедеятельности:
канд. техн. наук, доцент каф. КИБЭВС
_____Е.М. Давыдова
"_____"_____ 2017г.

Студент гр. 722
_____М.В. Мейта
"_____"_____ 2017г.

Руководитель:
канд. техн. наук, доцент каф. БИС
_____ А.С. Романов
«_____»_____2017г.

ТОМСК 2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)

УТВЕРЖДАЮ
Зав. кафедрой КИБЭВС

_____ А.А. Шелупанов
«_____»_____2017 г.

ЗАДАНИЕ
по дипломному проектированию студенту _____
_____ группа _____ факультет _____

1. Тема проекта (работы): _____
_____

2. Срок сдачи студентом законченного проекта
_____

3. Исходные данные к проекту
_____
_____
_____
_____
_____
_____

4. Содержание расчетно-пояснительной записи (перечень подлежащих разработке вопросов)
_____
_____
_____
_____
_____
_____

5. Перечень графического материала (с точным указанием обязательных чертежей)
_____
_____
_____
_____
_____
_____
_____

РЕФЕРАТ

Отчет содержит 63 страницу, 10 рисунков, 6 таблиц, 40 источник, 6 приложений.

СТИЛОМЕТРИЯ, ИСХОДНЫЙ КОД, ДЕАНОНИМИЗАЦИЯ АВТОРА, C/C++, КЛАССИФИКАЦИЯ, PYTHON, SKLEARN, JUPYTER NOTEBOOK, DECISION TREES, RANDOM FOREST CLASSIFIER, CROSS-VALIDATION, ADA BOOST, EXTREMLY RANDOMIZED TREES, GITHUB, LATEX.

Цель работы — разработка программного обеспечения для определения авторства исходного кода программ на на языке C/C++, основанного на методах стилометрического анализа текста, с перспективой его дальнейшего применения в борьбе с киберпреступностью, в области лицензионных, патентных, и иных судебных разбирательств.

В рамках преддипломной практики были поставленны следующие задачи:

– обзор существующих исследований, разработок, методов стилометрического анализа текста, в том числе исходного кода программ;

– построение модели процесса определения авторства исходного кода;

– разработка программного обеспечения для анализа исходного кода программ с применением стилометрии для определения авторства программного обеспечения;

– разработка программного интерфейса;

– подготовка и обработка тестового набора данных;

– исследование эффективности разработанной программы на основе модели анализа исходных кодов, анализ результатов.

Объект исследования: деанонимизация автора программного обеспечения.

Предмет исследования: стилометрия исходного кода программ на языках высокого уровня.

Достигнутые результаты: главным результатом преддипломной практики является программное обеспечение «WhoseCppCode», предназначенное для построения, тестирования и оценки модели классификации авторов исходного кода на языке C/C++, а также визуализации полученных результатов.

Отчет по преддипломной практике выполнен согласно ОС ТУСУР 01-2013 [1] при помощи системы компьютерной вёрстки LaTeX.

ABSTRACT

Research paper contains 63 pages, 10 pictures, 19 tables, 40 sources, 1 appendix.

STYLOMETRY, SOURCE CODE, AUTHORSHIP ATTRIBUTION, C/C++, CLASSIFICATION, PYTHON, SKLEARN, JUPYTER NOTEBOOK, DECISION TREES, RANDOM FOREST CLASSIFIER, CROSS-VALIDATION, ADA BOOST, EXTREMLY RANDOMIZED TREES, GITHUB, LATEX.

Цель работы — разработка алгоритма для определения авторства программного обеспечения, основанного на стилометрическом анализе исходного кода программ на языках высокого уровня.

The aim of this work is a software development of the tool for deanonimization of programmers, based on stylometry analysis of source code written in high-level programming languages.

Содержание

Введение

Задача определения авторства является широко распространенной проблемой в рамках исследования естественных языков, однако в меньшей степени для языков программирования. Тем не менее, с распространением применения компьютерных систем и сетей возросло и количество преступлений в информационной сфере. Существует множество разновидностей кибератак — различные компьютерные вирусы, трояны, несанкционированное копирование данных с кредитных карт, DDoS-атаки и многое другое. Возможность деанонимизации авторов вредоносного программного обеспечения может внести существенный вклад в развитие компьютерной криминалистики.

Считается, что у каждого программиста есть свои специфические профессиональные приемы, привычки, методы написания программного кода, свой так называемый «стиль программирования» и иные признаки, идентифицирующие автора. При этом, как и в случае с естественными языками, на индивидуальный «почерк» программиста может оказывать влияние множество факторов, таких как образование, географическое место проживания, уровень квалификации и другие. «Почерк» также может изменяться с течением времени, развитием технологий и общепринятых норм «хорошего» стиля написания программ. Под «хорошим» стилем обычно понимается набор правил, позволяющих писать код, удобный для чтения, понимания, внедрения дальнейших изменений и рефакторинга. Крупные IT-компании и корпорации обычно вводят свои собственные стандарты кодирования, которые зачастую используются сторонними организациями и индивидуальными программистами в своей работе. Примером могут служить руководства по стилю программирования на языке C++ компаний Google [2] и Geosoft [3].

Определение авторства исходного кода представляет собой актуальную задачу в сфере информационной безопасности, лицензирования в области разработки программного обеспечения, а также может оказать существенную помощь во время судебных разбирательств, при решении вопросов об интеллектуальной собственности и плагиате.

Целью настоящей дипломной работы является разработка программного обеспечения для определения авторства исходного кода программ на языке C/C++, основанного на методах стилометрического анализа текста.

1 Кафедра КИБЭВС

Местом прохождения практики была выбрана кафедра ТУСУРа – КИБЭВС (Кафедра комплексной информационной безопасности электронно- вычислительных систем).

Согласно информации с официального сайта [4], кафедра организована в ТУСУР в 1971 году как кафедра «Конструирования и производства электронно-вычислительной аппаратуры» (КиПЭВА) вскоре переименованной в кафедру «Конструирования электронно-вычислительной аппаратуры» (КЭВА).

21 сентября 1999 г. в связи с открытием новой актуальной специальности 090105 – «Комплексное обеспечение информационной безопасности автоматизированных систем» кафедра КЭВА была переименована в кафедру «Комплексной информационной безопасности электронно-вычислительных систем» (КИБЭВС). Заведующим кафедрой КИБЭВС, а также ректором ТУСУРа на сегодняшний день является ректор ТУСУРа, Александр Александрович Шелупанов, лауреат премии Правительства Российской Федерации, действительный член Международной Академиинаук высшей школы РФ, действительный член Международной Академии информации, Почетный работник высшего профессионального образования РФ, заместитель Председателя Сибирского регионального отделения учебно-методического объединения вузов России по образованию в области информационной безопасности, профессор, доктор технических наук.

Кадровый состав непрерывно укреплялся с момента её образования в 1971 году. В 2011 году, в год 40-летия кафедры, её коллектив состоял из 53 человек, в их числе 34 опытных высококвалифицированных специалиста и 19 аспирантов. Среди сотрудников кафедры члены Академий наук РФ; 4 профессора; 12 доцентов, кандидатов наук; старшие научные сотрудники, кандидаты наук и др.

С 2008 г. кафедра КИБЭВС входит в состав Института «Системной интеграции и безопасности».

На базе кафедры КИБЭВС ТУСУР в 2002 году организовано «Сибирское региональное отделение учебно-методического объединения Вузов России по образованию в области информационной безопасности.

2 Обзор информационных источников

На первом этапе практики необходимо было провести подробный аналитический обзор информационных источников, рассмотреть существующие методы определения авторства исходного кода и различные подходы к решению такого рода задачи.

В работе [5] представлен набор инструментов и техник, используемых для решения задач анализа авторства исходного кода, а также обзор некоторых наработок в данной предметной области. Кроме того, авторы приводят собственную классификацию проблем и подходов к их решению в рамках задачи деанонимизации авторов программного обеспечения.

Frantzeskou [5] выделяет следующие проблемы (задачи) анализа авторства исходного кода:

– идентификация автора — направлена на определение, принадлежит ли определенный фрагмент кода конкретному автору;

– характеристика автора — базируется на анализе стиля программирования;

– определение плагиата — нахождение схожестей среди множества фрагментов файлов исходного кода;

– определение намерений автора — был ли код изначально вредоносным или стал таковым в следствие программной ошибки;

– дискриминация авторов — определение, был ли код написан одним автором или несколькими.

Подходы к решению вышеперечисленных проблем (задач):

– анализ «вручную» — данный подход включает в себя исследование и анализ фрагмента исходного кода экспертом;

– вычисление схожести — базируется на измерении и сравнении различных метрик или токенов для набора файлов исходного кода;

– статистический анализ — в таком подходе используются статистические техники, такие как дискриминантный анализ и стилометрия, позволяющие определить различия между авторами;

– машинное обучение — используются методы рассуждения на основе прецедентов и нейронные сети для классификации автора на базе некоторого набора метрик.

В работе [6] предложен способ определения авторства программного обеспечения. в основе которого лежит статистический подсчет метрик, отражающих «почерк создателя» программного обеспечения. На основе метрик составлен «профиль почерка» программистов и вычисляется отклонение от данного профиля для каждого автора. Преимуществом данного метода является его независимость от языков программирования. Метод получил название SCAP (Source Code Author

Profiles).

В [7] исходный код транслировался в абстрактные синтаксические деревья, после чего разбивался на функции. Дерево каждой функции принималось за отдельный документ с известным автором. Выборка, состоящая из такого рода деревьев подавалась на вход SVM-классификатору, оперирующему данными типа «дерево». Классификатор обучался на файлах исходного кода двух авторов, в результате чего удалось достичь точности около 67-88%.

В работах [8], [9] и [10] рассматривался способ атрибуции исходного кода с использованием метода N-грамм. Вопрос определения авторства программ в данной работе рассматривался с точки зрения определения плагиата. В качестве выборки использовался набор из 1640 файлов исходного кода, написанных 100 авторами. Позднее удалось улучшить точность классификации данной модели до 77% за счет применения рейтинговых схем.

В [11] применялся алгоритм классификации Random Forest [12] и построение абстрактных синтаксических деревьев. Обучение и тестирование производилось для количества авторов от 250 до 1600. При этом удалось добиться высокой точности — 94-98%. Кроме того, авторы статьи выяснили в ходе работы, что сложнее определить авторов более простых примеров, нежели сложных программ, а также значительно выделяются авторы с большим опытом программирования на C/C++. В своей дальнейшей работе [13] авторы предложили применение данного подхода для анализа неполных, некомпилируемых образцов кода.

На основании проведенного исследования было решено опробовать подход, основанный на вычислении статистических метрик, характеризующих авторский стиль написания программ, и методов машинного обучения.

Сравнительная таблица с подробным описанием данных информационных источников и используемых в них методов приведена в приложении Б.

3 Выбор набора признаков, характеризующих автора программы

Задача стилометрического анализа исходного кода состоит в выделении и статистическом подсчете лексических, синтаксических, структурных и или каких-либо иных признаков на основании обработки текста программы.

Перечисленные в данном разделе признаки, по которым идентифицировались авторы, являются характерными для языков C и C++, однако могут быть использованы для исследования C-подобных языков, например, D, Java, Objective C, C#, PHP, perl и другие.

3.1 Лексические признаки

Главная особенность данной группы признаков состоит в том, что они могут быть вычислены при непосредственном анализе исходного кода программы в виде текстового файла. При этом код программы может быть некомпилируемым, неполным, содержащим синтаксические или программные ошибки.

Лексические признаки, как правило, улучшают читаемость кода и включают в себя:

– Стиль комментирования — данная подгруппа определяет преобладающий в тексте вид комментариев (однострочные или многострочные), а также общее их количество.

– Стиль расстановки фигурных скобок — к наиболее известным относят «K&R», «Whitesmith», «One True Bracing Style», стиль Алмена и другие. [14]

– Стиль разметки — расстановка пробелов, табуляций, число переносов строки к общей длине файла.

Дополнительно вычисляются:

– Число макросов — использует ли программист директивы препроцессора.[15]

– Средняя длина строки — позволяет также оценить читаемость кода (слишком длинные программные файлы плохо воспринимаются человеком).

В приложении В приведено описание использованных при классификации авторов признаков.

3.2 Ключевые слова C++

Ключевые слова C++ представляют собой список зарезервированных последовательностей символов, используемых языком, недоступных для переопределения.

Для ключевых слов языка C++ вычислялась статистическая мера TF (term frequency), отображающая число вхождения некоторого ключевого слова к общему количеству слов в документе. Подсчет частот ключевых слов может дать представление о предпочтениях автора в определен-

ного рода конструкциях, например, циклов «for» относитльно «while» или «do while», а также об уровне его профессиональной квалификации (определенные конструкции языка C/C++ используются крайне редко, сложны для понимания и предназначены для решения узкоспециализированных задач).

Словарь из 84 ключевых слов C++ (стандарт 11) был взят на сайте с официальной документацией [16] и представлен в таблице 3.1.

Таблица 3.1 – Ключевые слова языка C++ (стандарт 11)

Ключевые слова языка C++					
alignas	char32_t	enum	namespace	return	try
alignof	class	explicit	new	short	typedef
and	compl	export	noexcept	signed	typeid
and_eq	const	extern	not	sizeof	typename
asm	constexpr	false	not_eq	static	union
auto	const_cast	float	nullptr	static_assert	unsigned
bitand	continue	for	operator	static_cast	using
bitor	decltype	friend	or	struct	virtual
bool	default	goto	or_eq	switch	void
break	delete	if	private	template	volatile
case	do	inline	protected	this	wchar_t
catch	double	int	public	thread_local	while
char	dynamic_cast	long	register	throw	xor
char16_t	else	mutable	reinterpret_cast	true	xor_eq

4 Моделирование

Описание процесса определения авторства исходного кода программ в виде модели «черного ящика» согласно методологии IDEF0 представлено на рисунке 4.1, его декомпозиция — на рисунке 4.2.

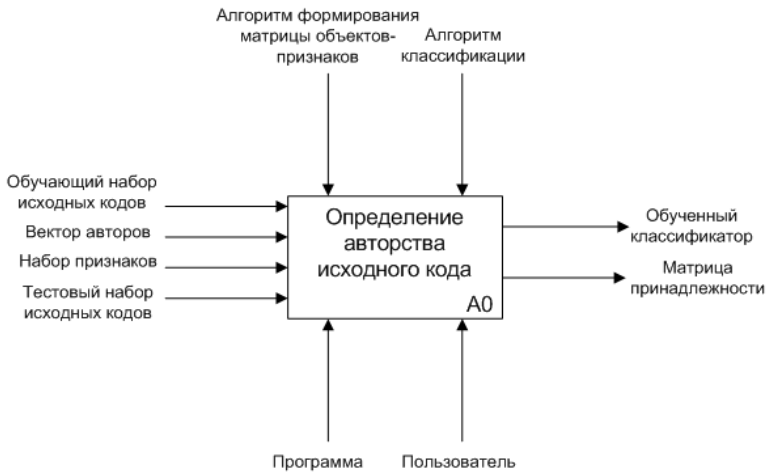

Рисунок 4.1 – Модель «черного ящика» процесса определения авторства исходного кода по методологии IDEF0

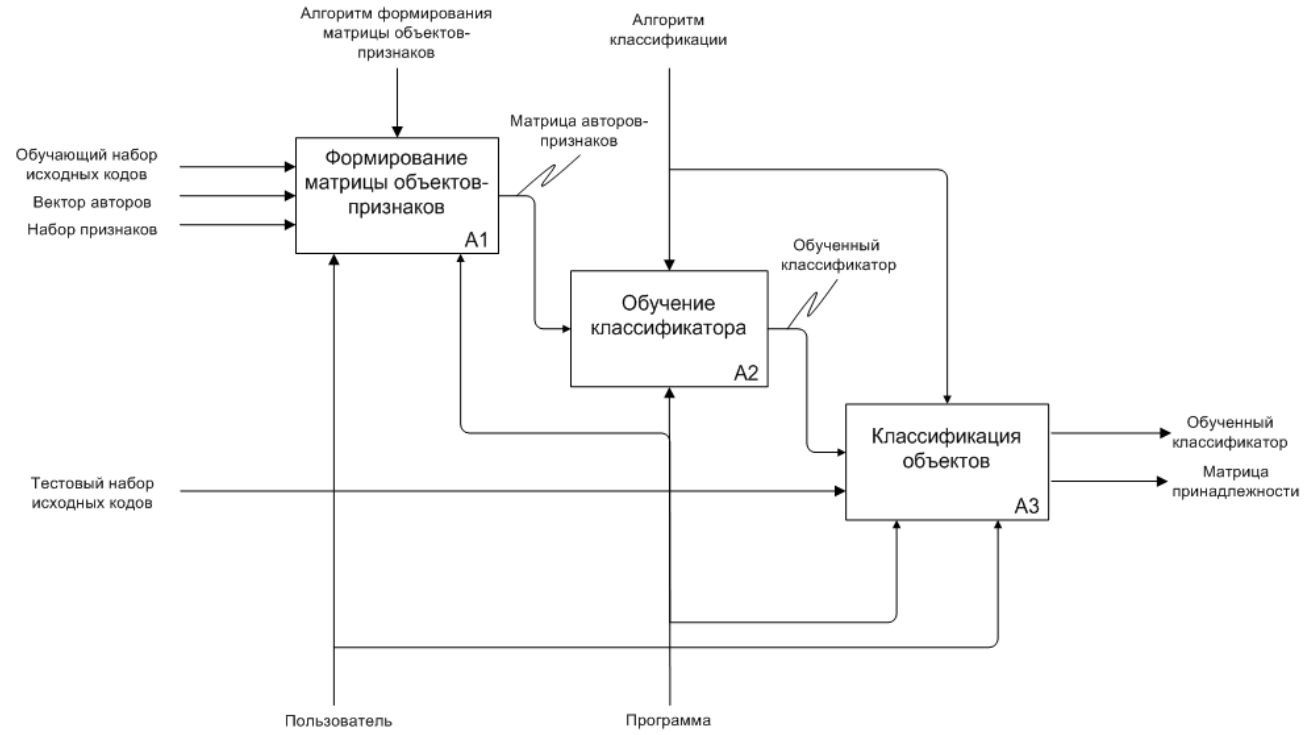

Рисунок 4.2 – Декомпозиция «черного ящика» процесса определения авторства исходного кода по методологии IDEF0

5 Классификация

В данной работе в качестве базового алгоритма для всех классификаторов (см. разделы 5.1, 5.2, 5.3) были выбраны деревья решений (Decision Trees), тестирование и оценка модели производилась на основе 10-фолдовой кросс-валидации (подробнее о тестировании модели в разделе **??**).

Деревья решений (Decision Trees) [17] или деревья принятия решений являются одним из наиболее популярных методов решения задач классификации, регрессии и прогнозирования. Впервые деревья решений были предложены Ховилендом и Хантом (Hoveland, Hunt) в конце 50-х годов прошлого века и в наиболее простом виде представляют собой совокупность правил в иерархической структуре. Основа такой структуры — это ветвление при проверке условий («Да» — «Нет»).

5.1 Random Forest

Алгоритм классификации Random Forest Classifier строится на двух базовых принципах:

– bagging — мета-алгоритм в машинном обучении, при котором на основе большого числа «слабых» классификаторов (в данном случае деревьев решений) строится один «сильный» классификатор (рис. 5.1);

– метод случайных подпространств.

Преимущества данного алгоритма классификации:

– способность эффективно обрабатывать данные с большим числом признаков и классов;

– нечувствительность к масштабированию (к любым монотонным преобразованиям) значений признаков;

– существует методы оценивания значимости отдельных признаков в модели;

– внутренняя оценка способности модели к обобщению (тест out-of-bag);

– высокая параллелизуемость и масштабируемость.

Недостатки алгоритма Random Forest Classifier:

– алгоритм склонен к переобучению на некоторых задачах, особенно на зашумленных, однако для избежания переобучения используется энтропия Шеннона или коэффициент прироста информации (англ. Gain);

– большой размер получаемых моделей приводит к существенным затратам памяти на хранение деревьев, однако данный недостаток решается повышением вычислительных мощностей и распараллеливанием вычислений. [12]

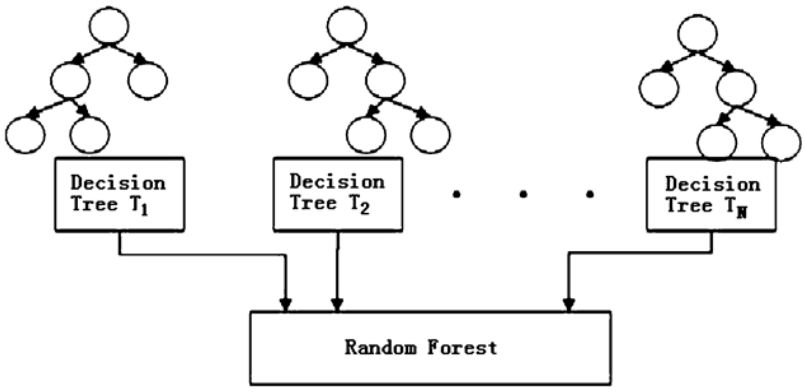

Рисунок 5.1 – Random Forest Classifier

5.2 AdaBoost

Алгоритм AdaBoost (сокр. от adaptive boosting) [18] является мета-алгоритмом, в процессе обучения строит композицию из базовых алгоритмов обучения для улучшения их эффективности.

Достоинства:

– Хорошая обобщающая способность. В реальных задачах (не всегда, но часто) удаётся строить композиции, превосходящие по качеству базовые алгоритмы. Обобщающая способность может улучшаться (в некоторых задачах) по мере увеличения числа базовых алгоритмов.

– Простота реализации.

– Время построения композиции практически полностью определяется временем обучения базовых алгоритмов.

Недостатки алгоритма классификаций AdaBoost:

– Склонен к переобучению при наличии значительного уровня шума в данных.

– Требует достаточно длинных обучающих выборок.

– Бустинг может приводить к построению громоздких композиций, состоящих из сотен алгоритмов. Такие композиции исключают возможность содержательной интерпретации, требуют больших объёмов памяти для хранения базовых алгоритмов и существенных затрат времени на вычисление классификаций.

5.3 ExtraTrees

Алгоритм ExtraTrees (Extremly Randomized Trees) [19] является модификацией алгоритма Random Forest Classifier (см. раздел 5.1), но отличается еще более рандомизированным разделением входного набора данных на подвыборки. Как правило, результаты работы данного алгоритма схожи с результатами Random Forest Classifier, однако в определенных случаях могут давать улучшение точности классификации.

6 Конструкторско-технологическая часть

6.1 Среда разработки и язык программирования

6.2 kmkmk

6.2.1 Программные модули

6.2.2 Интерфейс разрабатываемого программного обеспечения

6.2.3 subsection

6.2.4 ddffd

Программа «WhoseCppCode», разработанная в ходе работы, состоит из двух основных частей:

– программного модуля, реализующего все необходимые функции для сбора, анализа и обработки данных, а также построения модели классификации авторов исходного кода, описанной в разделе 4;

– программного интерфеса на основе технологии Jupyter Notebook [20], предназначенного для визуализации полученных в ходе классификации результатов, сбора необходимых данных с ресурса GitHub [21], построения матрицы объектов-признаков на основе входных данных, проведения вычислительных экспериментов.

Программный модуль реализован на языке прораммирования высокого уровня Python с использованием следующих программных библиотек:

– Scikit-Learn [22] — open-source библиотека для машинного обучения: классификации, регрессии, кластеризации и т.д.

– Plotly [23] — графическая Python-библиотека для построения интерактивных графиков, таблиц, диаграмм.

– Numpy [24] — библиотека для научных вычислений, предоставляющая методы работы с большими массивами данных.

– Scipy [25] — предоставляет среду для проведения математических и научных вычислений.

– Pandas [26] — open-source библиотека, предназначенная для анализа данных.

– Ipywidgets [27] — интерактивные HTML виджеты для Jupyter Notebook.

Интерфейс основан на веб-технологиях, может использоваться для демонстрации возможностей программ на языке Python. Библиотека Jupyter Notebook, с помощью которой был реализован

данный интерфейс, была выбрана за счет ряда преимуществ:

– является свободным ПО;

– поддерживает множество языков программирования;

– позволяет хранить вместе код, изображения, комментарии, формулы и графики;

– не требует знаний и применения веб-технологий, таких как CSS, HTML, JavaScript;

– может быть запущен на любом сервере, необходим только доступ по ssh/http;

– позволяет экспортировать код и сам блокнот в любом формате;

– предназначена для демонстрации разработок на языке Python (в основном в машинном обучении).

Основной модуль программы «WhoseCppCode» может быть использован отдельно от Jupyter Notebook при разработке различного рода программ, систем и интерфейсов лицами, заинтересованными в задаче классификации программистов.

Диаграмма действий в нотации UML, описывющая основной алгоритм работы программы «WhoseCppCode» представлена на рисунке 6.1, примеры ввода и вывода данных в интерфейсе Jupyter Notebook — на рисунках 6.2, 6.3 и 6.4.

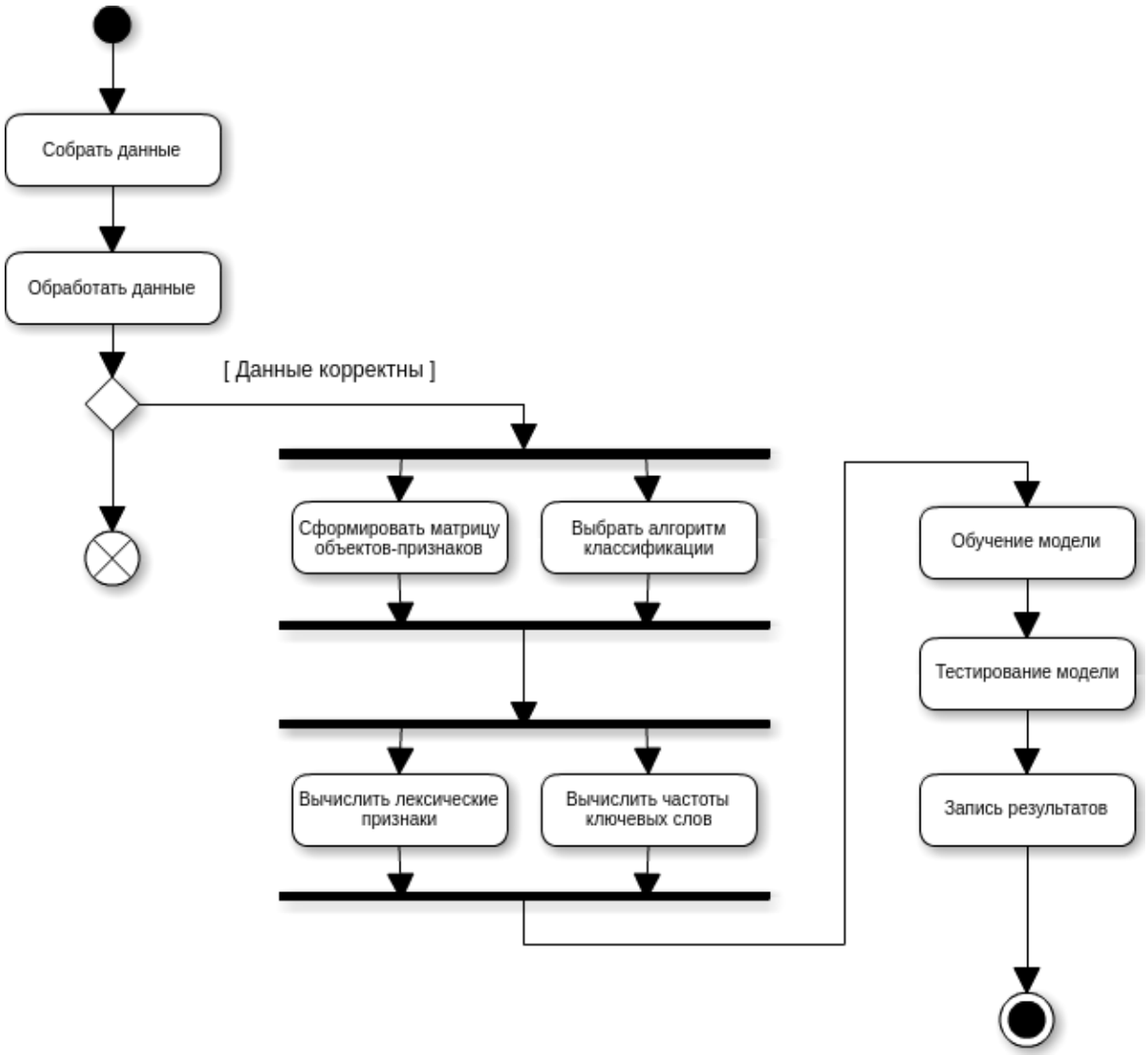

Рисунок 6.1 – Диаграмма действий алгоритма работы программы «WhoseCppCode»

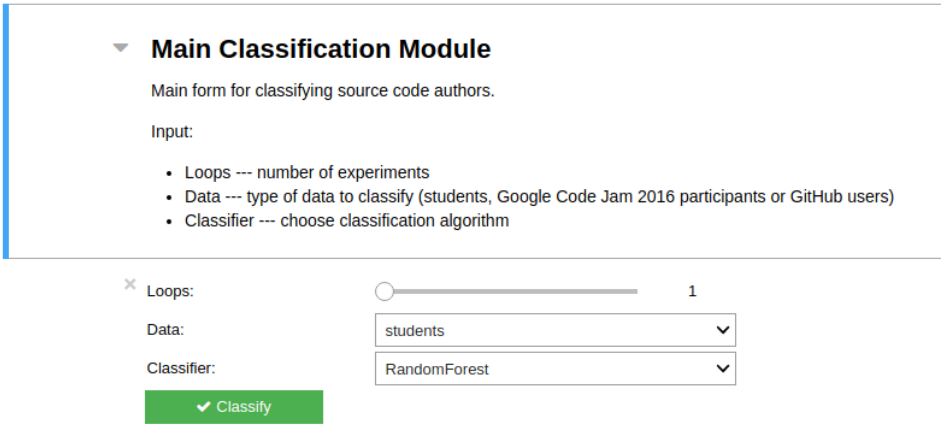

Рисунок 6.2 – Вид основной формы программного интерфейса

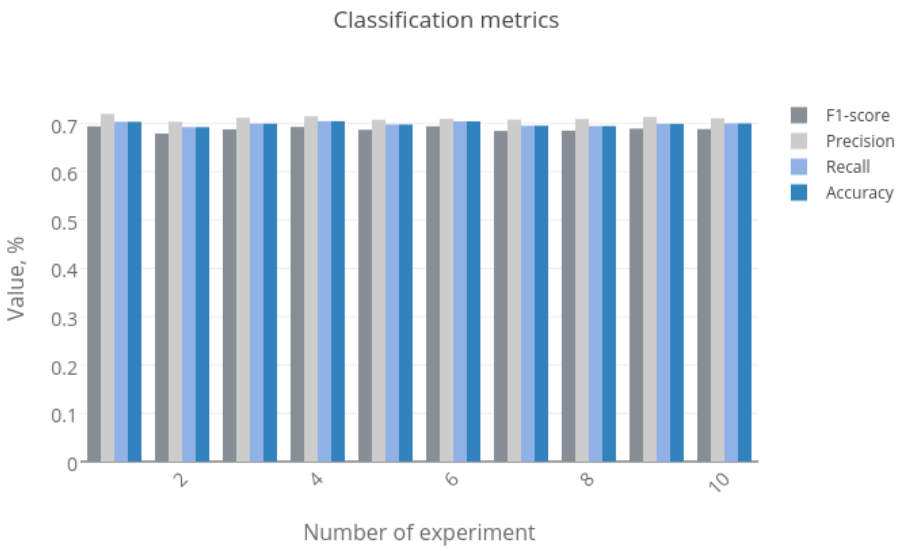

Рисунок 6.3 – Вывод диаграммы результатов классификации

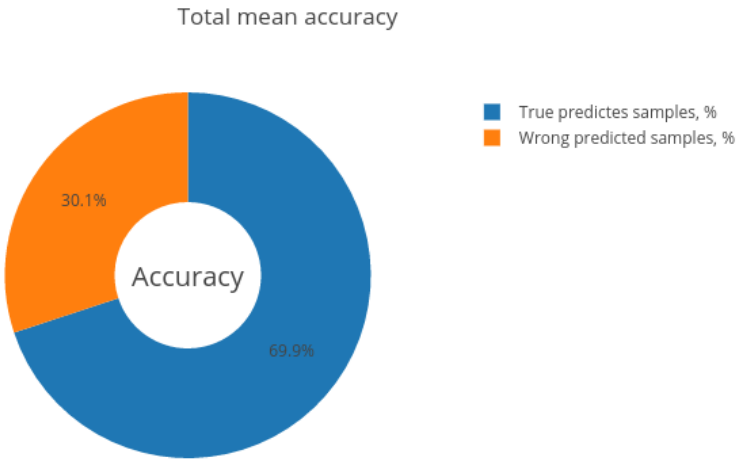

Рисунок 6.4 – Пример вывода диаграммы для средней точности классификации

7 Программа и методика испытаний

Раздел «Программа и методика испытаний» был составлен и оформлен в соответствии с ГОСТ 19.301–79. [28]

7.1 Объект испытаний

7.1.1 Полное наименование системы и ее условное обозначение

Условное обозначение: «WhoseCppCode».

7.1.2 Область применения

Разработанный программный комплекс можно использовать

7.2 Цель испытаний

Испытания системы предназначены для оценки адекватности модели, ее точности

7.3 Требования к программе

7.4 Требования к программной документации

Пояснительная записка к дипломной работе должна включать в себя:

– задание по дипломному проектированию;

– руководство администратора (приложение Г);

– руководство программиста (приложение Д);

– руководство пользователя (приложение Е):

– результаты вычислительных экспериментов.

руководство пользователя должно быть оформлено Согласно ГОСТ РД 50-34.698-90 Автоматизированные системы требования к содержанию документов

7.5 Средства и порядок испытаний

7.5.1 Технические и программные средства, используемые во время испытаний

7.5.2 Порядок проведения испытаний

7.6 Методы испытаний

Для тестирования аналитической модели в машинном обучении применяется процедура скользящего контроля, получившая название кросс-валидации (cross-validation) или перекрестной проверки. [29]

Процедура кросс-валидации включает в себя случайное разбиение на k подгрупп (или фолдов) примерно одинакового размера. Первый фолд служит для тестирования модели, остальные используются для обучения классификатора. Для тестовой подвыборки вычисляется среднеквадратичное отклонение. Процедура повторяется k-1 раз, при этом каждая из подгрупп выступает в роли тестовой выборки.

В данной работе тестирование производилось с применением 10-фолдовой кросс-валидации. Всего было произведено 10 вычислительных экспериментов.

7.7 Описание тестового набора данных

Burrows в работе [9] выделяет следующие ключевые параметры тестовых данных, которые могут влиять на точность классификации:

– Число авторов — с увеличением числа авторов сложность классификации увеличивается, точность — снижается.

– Число экземпляров выборки для каждого автора — желательно соблюдать одинаковым для всех авторов во избежание отклонения в сторону наиболее точно описанных авторов, а также иметь больше экземпляров для увеличения размера тестовой выборки.

– Средняя длина образца кода (количество непустых строк кода) — чем длиннее, тем выше точность классификации. Изменение длины экземпляров выборки может влиять на отклонение в сторону наиболее точно описанных авторов, однако не представляется возможным соблюдать длину экземпляра выборки постоянной.

– «Стилистическая зрелость» (stylistic maturity) авторов — уровень квалификации, личные и профессиональные предпочтения в стиле написания программ.

– Временные метки образцов кода (подразумевается, что со временем программы устаревают, технологии и методы программирования меняются и, как следствие, изменяется стиль программирования).

– Репрезентативность выборки — демографические, социальные и другие факторы.

– Типы авторов — студент, фрилансер, профессиональный разработчик. В идеале система должна включать в себя разные типы.

– Языки программирования — если тестировать несколько языков одновременно, результат будет зависеть от характерных признаков языка.

– Авторство в одном лице — большинство проектов выполняются в сотрудничестве с другими разработчиками.

– Корректное авторство — без плагиата, копирования и т.п.

Burrows упоминает также от том, что характерный стиль программирования нестабилен в начале карьеры программиста, что может существенно отличать начинающего специалиста и профессионала разработки.

Программное обеспечение «WhoseCppCode» тестировалось на трех наборах данных:

1) «Students» — выборка представляет собой работы студентов первого курса обучения по предмету «Основы программирования». Все программы реализуют решения однотипных задач в рамках учебной дисциплины, что исключет их разделение при классификации по функциональному на-

значению вместо стилистических особенностей и снижение точности классификации.

2) «Google Code Jam» — общедоступные данные ежегодной международной олимпиады по программированию Google Code Jam 2016. [30] Так же, как и в первой выборке, авторы решали схожие задачи, используя различные подходы и алгоритмы.

3) «GitHub» — данные, собранные с сайта GitHub [21], крупнейшего [31] веб-сервиса для хостинга IT-проектов и их совместной разработки.

Сбор данных с веб-хостинга GitHub производился по следующему принципу:

1) Выбирались крупные open-source репозитории (удаленные хранилища программного кода и данных), посвященные разработке проектов на C/C++.

2) Просматривался список контрибьюторов (пользователей, вносивших изменения в проект).

3) В качестве авторов выбирались те контрибьюторы, у которых имеются личные проекты, написанные на C/C++.

4) На основе списка пользователей автоматически, средствами программы «WhoseCppCode», производился сбор и сохранение файлов исходного кода для каждого автора.

В таблице 7.1 приводится описание некоторых характеристик каждого набора данных. В данном случае под смешанным типом авторов подразумевается, что разработчики могли быть совершенно разного уровня квалификации и рода деятельности (студенты, фрилансеры, начинающие и профессиональные разработчики, программисты-любители и т.д.).

Таблица 7.1 – Тестовые данные

Набор данных	«Students»	«Google Code Jam»	«GitHub»
Число авторов	3	30	30
исло файлов исходного кода на одного автора	14	9	78
Всего файлов исходного кода	42	278	2334
Минимальное число строк кода	33	36	26
Максимальное число строк кода	160	461	16348
Среднее число строк кода на один файл исходного кода	45	87	234
Тип авторов	Студенты	Смешанный	Смешанный

Каждая выборка представляет собой совокупность файлов исходного кода программ на языке C/C++ с расширениями *.cpp, *.c, *.h, *.hpp, *.cxx, *.cc, *.ii, *.ixx, *.ipp, *.inl, *.txx, *.tpp, *.tpl.

7.8 Критерии оценки эффективности классификации

Критерии оценки работы классификатора [32] представлены в таблице 7.2, где:

– tp — истинно-положительное решение;

– tn — истинно-отрицательное решение;

– fp — ложно-положительное решение;

– fn — ложно-отрицательное решение;

– Accuracy (точность) — отношение количества документов, по которым классификатор принял правильное решение, к общему числу документов (примеров файлов исходного кода);

– Precision (правильность) — доля документов, действительно принадлежащих данному классу, относительно всех документов, которые система отнесла к этому классу;

– Recall (полнота) — доля найденных классфикатором документов, принадлежащих классу, относительно всех документов этого класса в тестовой выборке;

– F1-score (F1-мера) — гармоническое среднее между правильностью и полнотой.

Таблица 7.2 – Критерии оценки работы классификатора

Критерий	Формула	Луч. знач.	Худ. знач.
Accuracy (точность)	(tp + tn) / число примеров * 100 %	100 %	0 %
Precision (правильность)	tp / (tp + fp)	1	0
Recall (полнота)	tp / (tp + fn)	1	0
F1-score (F1-мера)	2 * (precision * recall) / (precision + recall)	1	0

7.9 Результаты классификации

При тестировании классификатора использовались критерии оценки, описанные в разделе 7.8, а также время работы программы. Результаты работы классификатора представлены в таблице 7.3, а также на рисунках 7.1, 7.2 и 7.3.

Таблица 7.3 – Результаты работы

Набор данных «Students»					
Классификатор	Accuracy, %	Precision	Recall	F1-score	Время работы, сек.
RandomForest	89,55	0,90	0,93	0,90	93,61
AdaBoost	70,45	0,70	0,74	0,70	53,22
ExtraTrees	91,85	0,92	0,95	0,92	53,70
Набор данных «Google Code Jam»					
Классификатор	Accuracy, %	Precision	Recall	F1-score	Время работы, сек.
RandomForest	86,66	0,86	0,88	0,87	110,65
AdaBoost	19,43	0,16	0,16	0,19	103,53
ExtraTrees	88,09	0,88	0,90	0,88	60,34
Набор данных «GitHub»					
Классификатор	Accuracy, %	Precision	Recall	F1-score	Время работы, сек.
RandomForest	69,92	0,69	0,71	0,70	223,77
AdaBoost	16,44	0,09	0,11	0,16	451,63
ExtraTrees	70,99	0,70	0,72	0,71	201,13

Наихудшие результаты показал алгоритм AdaBoost, в то время как наиболее точным и быстрым из трех представленных алгоритмов оказался ExtraTrees (см. раздел 5.3).

С использованием метода, основанного на извлечении лексических признаков и классификации с помощью алгоритма ExtraTrees (Extremly Randomized Trees) точность классификации составила 70-71 % на выборке данных из 30 авторов и 2334 неполных, немопилируемых файлов с вебхостинга GitHub.

По результатам классификации можно сделать вывод, что данные методы могут применяться не только в «лабораторных» условиях, когда тестовая выборка генерируется на основе студенческих работ или результатов олимпиад по программированию, где решаются схожие задачи, ограниченные по времени и объему кода, но и в условиях реального мира.

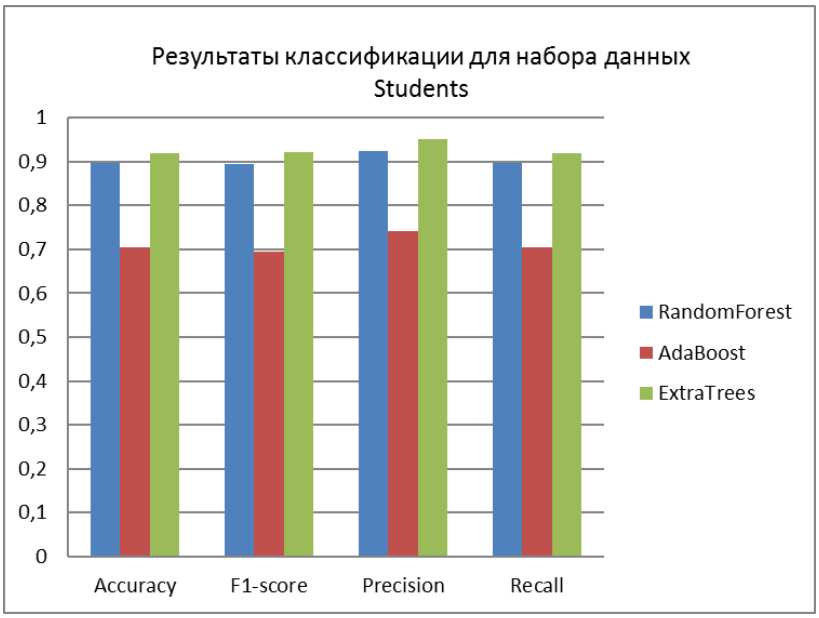

Рисунок 7.1 – «Students»

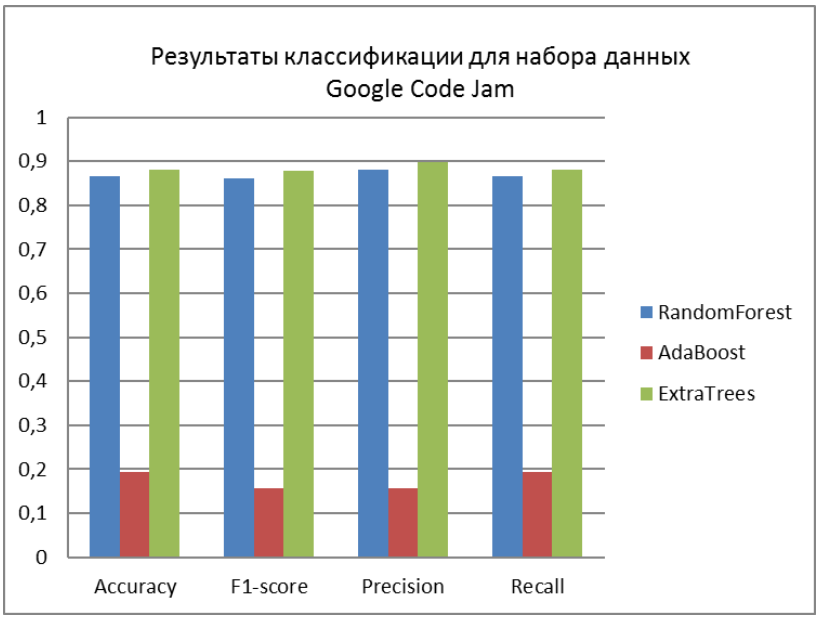

Рисунок 7.2 – «Google Code Jam»

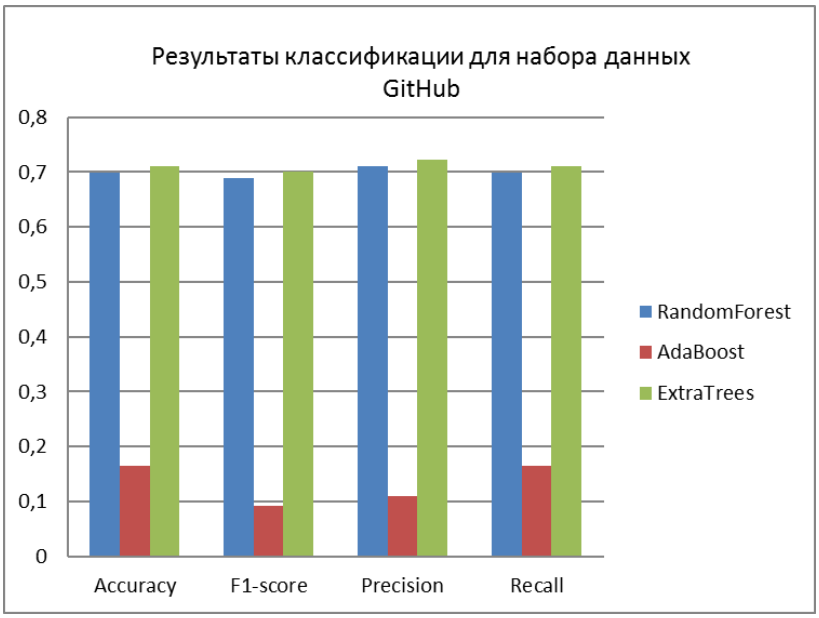

Рисунок 7.3 – «GitHub»

8 Безопасность жизнедеятельности

8.1 Анализ опасных и вредных производственных факторов на рабочем месте

В ходе трудового процесса организм человека может подвергаться различным воздействиям, оказывающим влияние на его здоровье и работоспособность. Подобное воздействие может приводить к различным результирующим последствиям, которые зависят от характера воздействия (прямого или опосредованного), наличия тех или иных факторов производственной среды, а также условий их проявления.

Принято разграничивать производственные факторы на две основные группы — опасные производственные факторы (ОПФ) и вредные производственные факторы (ВПФ). При этом однозначно отнести тот или иной фактор к подмножеству опасных или вредных не всегда представляется возможным, поскольку даже нейтральные производственные факторы при наличии определенных условий и обстоятельств могут становиться вредными или опасными для человека, приводить к травмам и заболеваниям, связанным с трудовой деятельностью.

Согласно [33] опасные и вредные производственные факторы производственной среды по природе их воздействия на организм работающего человека подразделяются на:

– факторы, воздействие которых носит физическую природу;

– факторы, воздействие которых носит химическую природу;

– факторы, воздействие которых носит биологическую природу.

При работе за ПЭВМ воздействие на организм человека носит физическую природу. Кроме того работники подвергаются нервно-психическим перегрузкам.

Основываясь на классификации вредных и опасных факторов производства из ГОСТ 12.0.003–2015 [33] можно выделить следующие физические факторы, связанные с работой за ПЭВМ:

– повышенный уровень и другие неблагоприятные факторы шума;

– повышенная или пониженная температура воздуха рабочей зоны;

– повышенная или пониженная влажность воздуха;

– повышенная или пониженная подвижность воздуха;

– повышенное значение напряжения в электрической цепи;

– повышенный уровень статического электричества;

– повышенный уровень электромагнитных излучений;

– отсутствие или недостаток естественного освещения;

– отсутствие или недостаток искусственного освещения;

– повышенная яркость, пульсация света.

К нервно-психическим перегрузкам относят [33]:

– умственное перенапряжение, в том числе вызванное информационной перегрузкой;

– перенапряжение анализаторов, в том числе вызванное информационной перегрузкой;

– монотонность труда, вызывающая монотонию;

– эмоциональные перегрузки.

Под информационной перегрузкой понимается воспринимаемая сенсорными системами организма человека интенсивность поступления информации, воздействующая на центральную нервную систему человека и способная приводить к различным неблагоприятным последствиям для здоровья.

8.1.1 Перечень продукции и контролируемые гигиенические параметры вредных и опасных факторов

Перечень продукции и контролируемых гигиенических параметров огласно [34] приведен в таблице 8.1.

Таблица 8.1 – Перечень продукции, контролируемых гигиенических параметров

Вид продукции	Контролируемые гигиенические параметры
Машина вычислительная электронная цифровая персональная (ПЭВМ)	– уровни электромагнитных полей (ЭМП); – уровни акустического шума; – концентрация вредных веществ в воздухе; – визуальные показатели видеодисплейного терминала
Устройства периферийные: модем, клавиатура, принтер, устройства хранения информации	– уровни ЭМП; – уровни акустического шума; – концентрация вредных веществ в воздухе

8.1.2 Требования к уровням шума на рабочих местах, оборудованных ПЭВМ

Допустимые уровни звукового давления и уровней звука, создаваемых ПЭВМ, не должны превышать значений, представленных в таблице 8.2. Измерение уровня звука и уровней звукового давления проводится на расстоянии 50 см от поверхности оборудования и на высоте расположения источника(ков) звука. Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ. [34]

Таблица 8.2 – Допустимые значения уровней звукового давления

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	50

8.1.3 Требования к уровням электромагнитных полей на рабочих местах, оборудованных ПЭВМ

Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах пользователей представлены в таблице 8.3. [34]

Таблица 8.3 – Временные допустимые уровни ЭМП, создаваемых ПЭВМ на рабочих местах

Наименование параметров		ВДУ
Напряженность электрического поля	в диапазоне частот 5 Гц – 2 кГц	25 В/м
	в диапазоне частот 2 кГц – 400 кГц	2,5 В/м
Плотность магнитного потока	в диапазоне частот 5 Гц - 2 кГц	250 нТл
	в диапазоне частот 2 кГц – 400 кГц	25 нТл
Напряженность электростатического поля		15 кВ/м

8.1.4 Требования к визуальным параметрам устройств отображения информации

Предельно допустимые значения визуальных параметров визуального дисплейного терминала, контролируемые на рабочих местах, представлены в таблице 8.4. [34]

Таблица 8.4 – Визуальные параметры устройств отображения информации

Параметры	Допустимые значения
Яркость белого поля	Не менее 35 кд/кв.м
Неравномерность яркости рабочего поля	Не более $\pm$ 20%
Контрасность (для монохромного режима)	Не менее 3:1
Временная нестабильность изображений (непреднамеренное изменение во времени яркости изображения на экране дисплея)	Не должна фиксироваться
Пространственная нестабильность изображения (непреднамеренные изменения положения фрагментов изображения на экране)	Не более $2 \times 1E(-4L)$, где L — проектное расстояние наблюдения, мм

Для дисплеев на ЭЛТ частота обновления изображения должна быть не менее 75 Гц при всех режимах разрешения экрана, гарантируемых нормативной документацией на конкретный тип дис-

плея и не менее 60 Гц для дисплеев на плоских дискретных экранах (жидкокристаллических, плазменных и т.п.). [34]

8.1.5 Требования к микроклимату. Концентрации вредных веществ, выделяемых ПЭВМ в воздух помещения

Показатели микроклимата должны обеспечивать сохранение теплового баланса человека с окружающей средой и поддержание оптимального или допустимого теплового состояния организма. Несоблюдение оптимальных микроклиматических условий может привести к ухудшению состояния здоровья, снижению работоспособности, ощущению дискомфорта, напряжению механизмов терморегуляции.

Работа за ПЭВМ относится к категории Iа: с интенсивностью энерготрат до 120 ккал/ч (до 139 Вт), производимая сидя и сопровождающаяся незначительным физическим напряжением. [35]

Согласно [35] допустимые величины показателей микроклимата на рабочих местах производственных помещений категории Iа должны соответствовать значениям, приведенным в таблице 8.5.

Амплитуда колебания температуры воздуха в течение смены не должна превышать 4° С.

Таблица 8.5 – Оптимальные величины показателей микроклимата на рабочих местах производственных помещений категории Iа

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Теплый	23-25	22-26	60-40	0,1

8.1.6 Требования к освещению на рабочих местах, оборудованных ПЭВМ

Согласно [34], к освещению рабочих мест, оборудованных ПЭВМ, представляются следующие требования:

1) Рабочие столы следует размещать таким образом, чтобы видеодисплейные терминалы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева.

2) Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В производственных и административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы ком-

бинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

3) Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк.

4) Следует ограничивать прямую блесткость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м^2.

5) Следует ограничивать отраженную блесткость на рабочих поверхностях (экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м^2 и яркость потолка не должна превышать 200 кд/м^2.

6) Показатель ослепленности для источников общего искусственного освещения в производственных помещениях должен быть не более 20. Показатель дискомфорта в административно-общественных помещениях не более 40, в дошкольных и учебных помещениях не более 15.

7) Яркость светильников общего освещения в зоне углов излучения от 50 до 90° с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м^2, защитный угол светильников должен быть не менее 40°.

8) Светильники местного освещения должны иметь непросвечивающий отражатель с защитным углом не менее 40°.

9) Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1-5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

10) Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору.

11) Коэффициент запаса (Кз) для осветительных установок общего освещения должен приниматься равным 1,4.

12) Коэффициент пульсации не должен превышать 5%.

13) Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и про-

водить своевременную замену перегоревших ламп.

Рассчитаем показатели освещенности E на рабочем месте с помощью формул из стандарта [36]:

$$E = \frac{I}{r^2} cos(i),$$

где I — сила света, кд;

i – угол падения лучей света относительно нормали к поверхности;

r – расстояние до источника света, м.

Формула для вычисления силы света:

$$I = \frac{F}{4\pi},$$

где F — номинальный световой поток, лм.

Помещение оборудовано 1-ой люменисцентной лампой и 3-мя лампами накаливания. Световой поток люменисцентной лампы мощностью 10 Вт составляет около 400 Лм, одной лампы накаливания мощностью 60 Вт — 710 Лм.

Сила света люменисцентной лампы:

$$I_1 = \frac{400}{4\pi} = 31,83 \text{ кд.}$$

Сила света ламп накаливания:

$$I_2 = \frac{3 \times 710}{4\pi} = 169,5 \text{ кд.}$$

Показатель освещенности на поверхности стола составляет:

$$E = \frac{31,83}{(0,3)^2} cos(30°) + \frac{169,5}{(1,5)^2} cos(45°) = 359,56 \text{ лк.}$$

Показатель освещенности на поверхности экрана составляет:

$$E = \frac{31,83}{(0,3)^2} cos(60°) + \frac{169,5}{(1,6)^2} cos(60°) = 209,94 \text{ лк.}$$

По результатам расчетов можно сделать вывод, что показатели освещенности соответствуют нормам, принятым в [34].

8.1.7 Требования к организации рабочих мест пользователей ПЭВМ

Организация и оборудование рабочих мест с ПЭВМ для взрослых пользователей согласно [34] включает в себя следующие требования:

1) Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680-800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм.

2) Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм.

3) Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной — не менее 500 мм, глубиной на уровне колен — не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

4) Конструкция рабочего стула должна обеспечивать:

– ширину и глубину поверхности сиденья не менее 400 мм;

– поверхность сиденья с закругленным передним краем;

– регулировку высоты поверхности сиденья в пределах 400-550 мм и углам наклона вперед до 15° и назад до 5°;

– высоту опорной поверхности спинки 300±20 мм, ширину — не менее 380 мм и радиус кривизны горизонтальной плоскости — 400 мм;

– угол наклона спинки в вертикальной плоскости в пределах ±30°;

– регулировку расстояния спинки от переднего края сиденья в пределах 260-400 мм;

– стационарные или съемные подлокотники длиной не менее 250 мм и шириной — 50-70 мм;

– регулировку подлокотников по высоте над сиденьем в пределах 230±30 мм и внутреннего расстояния между подлокотниками в пределах 350-500 мм.

5) Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

6) Клавиатуру следует располагать на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю, или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

Результаты сравнения показали, что значения параметров рабочего места соответствуют значениям рекомендуемых параметров.

8.2 Инструкция по работе на персональном компьютере

Инструкция по охране труда для пользователей ПЭВМ утверждена РД 153-34.0-03.2.98-2001 [37].

8.2.1 Противопожарная безопасность

Для соблюдения противопожарной безопасности запрещается:

– хранить и применять горючие жидкости, взрывчатые вещества, баллоны с газами и др.;

– использовать электронагревательные приборы;

– эксплуатировать провода электроприборов с поврежденной изоляцией;

– применять открытый огонь;

– курить в помещении;

– оставлять без наблюдения включенную в сеть ПЭВМ, оргтехнику, бытовую технику.

По окончании работы необходимо осмотреть помещения на наличие признаков возгорания. При обнаружении возгорания работник обязан:

– немедленно сообщить об этом по телефону «01» в пожарную охрану (при этом необходимо назвать адрес, место возникновения пожара, а также сообщить свою фамилию и должность);

– сообщить руководителю или его заместителю о пожаре;

– принять меры по организации эвакуации людей (эвакуацию начинать из помещения, где возник пожар, а также из помещений, которым угрожает опасность распространения огня и дыма);

– одновременно с эвакуацией людей, приступить к тушению пожара своими силами и имеющимися средствами пожаротушения.

8.2.2 Электрическая безопасность

Для соблюдения электрической безопасности запрещается:

– прикасаться к проводам и розеткам, открывать электрощитки;

– разбирать и проводить самостоятельно ремонт оборудования, розеток и т.д.;

– пользоваться поврежденными розетками, рубильниками, вилками и прочим электрооборудованием;

– пользоваться неисправной или незаземленной аппаратурой;

– нарушать правила эксплуатации ПЭВМ и оргтехники, а так же инструкции по работе на ПЭВМ и средствах оргтехники;

– включать в сетевые фильтры, блоки бесперебойного питания и специализированные розет-

ки, расположенные в коробах бытовую технику и другое, не относящееся к ПЭВМ оборудование.

По окончании работы необходимо обесточить все электроприборы. При наличии в помещении выделенной сети электропитания для ПЭВМ, необходимо выключить автомат питания в распределительном щите.

8.2.3 Требования охраны труда во время работы. Оказание первой медицинской помощи

При работе с персональным компьютером необходимо:

– соблюдать оптимальное расстояние от экрана видеомонитора до глаз, поддерживать рациональную рабочую позу и оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы;

– осуществлять систематическое проветривание помещения после каждого часа работы с ПЭВМ;

– работу за экраном видеомонитора следует периодически прерывать на регламентированные перерывы, которые устанавливаются для обеспечения работоспособности и сохранения здоровья, или заменять другой работой с целью сокращения рабочей нагрузки у экрана;

– продолжительность непрерывной работы с ПЭВМ без регламентированного перерыва не должна превышать двух часов.

К непосредственной работе на ПЭВМ допускаются лица, не имеющие медицинских противопоказаний. Женщины со времени установления беременности и в период кормления ребенка грудью к выполнению всех видов работ, связанных с использованием ПЭВМ, не допускаются.

Последовательность действий при оказании первой помощи пострадавшему:

– устранение воздействия на организм пострадавшего опасных и вредных факторов (освобождение его от действия электрического тока, гашение горящей одежды и т.п.);

– оценка состояния пострадавшего;

– определение характера травмы, создающей наибольшую угрозу для жизни пострадавшего, и последовательности действий по его спасению;

– выполнение необходимых мероприятий по спасению пострадавшего в порядке срочности (восстановление проходимости дыхательных путей, проведение искусственного дыхания, наружного массажа сердца, остановка кровотечения, иммобилизация места перелома, наложение повязки и т.д.);

– поддержание основных жизненных функций пострадавшего до прибытия медицинского персонала;

– вызов скорой медицинской помощи.

При поражении электрическим током необходимо как можно быстрее освободить пострадавшего от действия тока, так как от продолжительности его действия на организм зависит тяжесть электротравмы. Отключить электроустановку можно с помощью выключателя, рубильника или другого отключающего аппарата. Если отсутствует возможность быстрого отключения электроустановки, то необходимо принять меры к отделению пострадавшего от токоведущих частей, к которым он прикасается. При этом во всех случаях оказывающий помощь не должен прикасаться к пострадавшему без применения надлежащих мер предосторожности, так как это опасно для жизни. Он должен также следить за тем, чтобы самому не оказаться в контакте с токоведущей частью или под напряжением шага, находясь в зоне растекания тока замыкания на землю.

При оказании помощи пострадавшему при ожогах во избежание заражения нельзя касаться руками обожженных участков кожи или смазывать их мазями, жирами, маслами и т. п. Нельзя вскрывать пузыри, так как, удаляя их, можно легко содрать обожженную кожу и тем самым создать благоприятные условия для заражения раны. При небольших ожогах степени нужно наложить на обожженный участок кожи стерильную повязку.

Одежду и обувь с обожженного места нельзя срывать, а следует разрезать ножницами и осторожно снять. Если обгоревшие куски одежды прилипли к обожженному участку кожи, то поверх них необходимо наложить стерильную повязку и направить пострадавшего в лечебное учреждение.

При тяжелых и обширных ожогах необходимо пострадавшего завернуть в чистую простыню или ткань, не раздевая его, укрыть, напоить теплым чаем и создать покой до прибытия врача.

9 Технико-экономическое обоснование

9.1 Обоснование необходимости проводимого исследования

Вследствие активного развития информационных технологий возрастает и количество преступлений в информационной сфере. Зачастую злоумышленники, как внешние, так и внутренние, используют самостоятельно разработанные программы для осуществления атак на информационные ресурсы. Системы, способные идентифицировать разработчиков вредоносного ПО, могут внести существенный вклад в развитие компьютерной криминалистики, а также оказывать помощь в исследовании вопросов интеллектуальной собственности среди разработчиков программного обеспечения.

Цель настроящей дипломной работы — разработать ПО, способное идентифицировать автора программы по исходному коду, с перспективой его дальнейшего применения в борьбе с киберпреступностью, в области лицензионных, патентных, и иных судебных разбирательств.

9.2 Организация и планирование работы

Основные задачи организации и планирования работ:

– определение объема предстоящих работ;

– определение основных этапов работ;

– установление сроков выполнения запланированных работ;

– определение необходимых денежных, материальных и трудовых ресурсов.

При выполнении дипломной работы были задействованы следующие лица:

– руководитель (рук.);

– разработчик (разр.).

Месячный оклад студента, не являющегося дипломированным специалистом, составляет 2324,40 рублей. С учетом 24 рабочих дней и 6-часового рабочего дня стоимость одного часа работ равна 16,14 рублей. Месячный оклад руководителя с ученой степенью кандидата наук и должностью доцента [38] составляет 14800 рублей. Стоимость одного часа работ с учетом 24-ех 6-часовых рабочих дней равна 102,78 рублей.

Руководитель работы оказывает помощь разработчику в планировании работ в период проектирования, рекомендует необходимую литературу, проводит консультации разработчика, осуществляет контроль над выполнением всех намеченных этапов работы. Разработчик реализует объем работ, установленный в техническом задании.

График работ приведен в таблице 9.1. Зная длительность цикла каждого этапа и возможность их параллельно-последовательного выполнения, можно рассчитать срок завершения планируемых

работ и составить ленточный и сетевой графики плана их выполнения (табл. 9.2 и 9.3).

Таблица 9.1 – График выполнения работ

Наименование этапов и содержание работ	Исполнитель (должность)	Трудоемкость		Количество исполнителей, чел.	Стоимость одного часа работ, руб/час	Общая стоимость работы, руб.	Продолжительность рабочего дня, час.	Срок исполнения, дни
		Нормо-часы, н-ч	Процент от общей трудоемкости, %					
1 Постановка задачи	Рук.	6	6	1	102,78	616,68	6	1
	Разр.	6	1	1	16,14	96,84	6	1
2 Обзор информационных источников	Разр.	30	6	1	16,14	484,2	6	5
3 Построение модели процесса определения авторства исходного кода	Разр.	36	7	1	16,14	581,04	6	6
4 Программная реализация разработанной модели	Разр.	96	18	1	16,14	1549,44	6	16
5 Разработка программного интерфейса	Разр.	72	14	1	16,14	1162,08	6	12
6 Сбор и обработка тестовых данных	Разр.	72	14	1	16,14	1162,08	6	12
7 Тестирование и отладка полученной модели	Разр.	72	14	1	16,14	1162,08	6	12
8 Анализ результатов	Рук.	36	35	1	102,78	3700,08	6	6
	Разр.	36	7	1	16,14	581,04	6	6
9 Оформление основной части отчета	Разр.	30	6	1	16,14	484,2	6	5
10 Проверка и исправление основной части отчета	Рук.	36	35	1	102,78	3700,08	6	6
	Разр.	36	7	1	16,14	581,04	6	6
11 Проведение расчетов по технико-экономическому обоснованию	Разр.	6	1	1	16,14	96,84	6	1
12 Проведение расчетов по безопасности жизнедеятельности	Разр.	6	1	1	16,14	96,84	6	1
13 Проверка и исправление пояснительной записки	Рук.	24	23	1	102,78	2466,72	6	4
	Разр.	24	5	1	16,14	387,36	6	4
Всего: 13	Рук.	102	100	1	102,78	10483,56	6	17
	Разр.	522	100	1	16,14	8425,08	6	87

Таблица 9.2 – Ленточный график загрузки участников работ

Этапы работы	Исполнитель	Длительность, дн.	Продолжительность работ, недели														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Рук.	1															
	Разр.	1															
2	Разр.	5															
3	Разр.	6															
4	Разр.	16															
5	Разр.	12															
6	Разр.	12															
7	Разр.	12															
8	Рук.	6															
	Разр.	6															
9	Разр.	5															
10	Рук.	6															
	Разр.	6															
11	Разр.	1															
12	Разр.	1															
13	Рук.	4															
	Разр.	4															

9.3 Определение сметной стоимости проекта

9.3.1 Общие положения

Смета затрат для данной работы состоит из расходов, которые включают в себя следующие статьи:

– затраты на оборудование и амортизацию;

– расходы на оплату труда и отчисления на социальные нужды;

– затраты на основные и вспомогательные материалы;

Таблица 9.3 – Календарный график загрузки участников

Этапы работы	Исполнитель	Длительность, дн.	Календарные даты
1 Постановка задачи	Рук.	1	6.02.2017
	Разр.	1	
2 Обзор информационных источников	Разр.	5	7.02.2017 — 11.02.2017
3 Построение модели процесса определения авторства исходного кода	Разр.	6	13.02.2017 — 18.02.2017
4 Программная реализация разработанной модели	Разр.	16	20.02.2017 — 11.03.2017
5 Разработка программного интерфейса	Разр.	12	13.03.2017 — 25.03.2017
6 Сбор и обработка тестовых данных	Разр.	12	27.03.2017 — 8.04.2017
7 Тестирование и отладка полученной модели	Разр.	12	10.04.2017 — 22.04.2017
8 Анализ результатов	Рук.	6	24.04.2017 — 29.04.2017
	Разр.	6	
9 Оформление основной части отчета	Разр.	5	2.05.2017 — 6.05.2017
10 Проверка и исправление основной части отчета	Рук.	6	8.05.2017 — 13.05.2017
	Разр.	6	
11 Проведение расчетов по технико-экономическому обоснованию	Разр.	1	29.05.2017
12 Проведение расчетов по безопасности жизнедеятельности	Разр.	1	30.05.2017
13 Проверка и исправление пояснительной записки	Рук.	4	31.05.2017 — 3.06.2017
	Разр.	4	

– затраты на электроэнергию.

9.3.2 Затраты на оборудование и амортизацию

Основным оборудованием при проведении работы являются компьютер и принтер, которые постановлением Правительства Российской Федерации от 1.01.02 г. № 1 отнесены ко второй амортизационной группе – «имущество со сроком полезного использования свыше 2 лет до 3 лет включительно» [39]. Месячная норма амортизации составляет 2,8% и для ноутбука, и для принтера.

Результаты расчётов амортизационных отчислений приведены в таблице 9.4.

Таблица 9.4 – Смета затрат на оборудование

Наименование прибора, оборудования	Потребленное количество, шт.	Цена, руб.		Время использования по теме	Месячная норма амортизации, %	Сумма амортизации, руб.
		Единицы	Всего			
Ноутбук	1	25000	25000	4 мес.	2,8	2800
Принтер	1	6500	6500	4 мес.	2,8	728
Итого: 3528 руб.						

9.3.3 Расходы на оплату труда и отчисления на социальные нужды

Статья затрат учитывает выплаты по заработной плате за выполненную работу, вычисленные на основании тарифных ставок и должностных окладов в соответствии с принятой в организации-разработчике системой оплаты труда. В этой статье также отражаются премии, надбавки и доплаты за условия труда, оплата ежегодных отпусков, выплата районного коэффициента и некоторые другие расходы. Отчисления на социальные нужды учитывают страховые взносы.

Результаты расчёта расходов на оплату труда участников проекта представлены в таблице 9.5.

Таблица 9.5 – Расчет расходов на оплату труда участников проекта

Участники проекта	$ЗП_{пр}$	Премия	РН, руб. 30%	ОЗП	ДЗП 15%	ФОТ	Страховые взносы, руб. 30%	Всего
Рук.	10483,56	—	3145,07	13628,63	2044,29	15672,92	4701,88	20374,80
Разр.	8425,08	—	2527,52	10952,60	1642,89	12595,49	3778,65	16374,14
Итого: 36748,94 руб.								

9.3.4 Затраты на основные и вспомогательные материалы

Статья включает расходы по приобретению и доставке основных и вспомогательных материалов, необходимых для опытно-экспериментальной проработки решения, для изготовления макета или опытного оборудования. Сюда включаются и стоимость необходимых материалов для изготовления образцов и макетов, и материалов необходимых для оформления требуемой документации.

Размер транспортно-заготовительных расходов (ТЗР), определяемый в процентах от стоимости, примем 10%. Стоимость вспомогательных материалов принимается 10% от стоимости основных материалов с учётом ТЗР. Результаты расчёта стоимости материалов представлены в 9.6.

Таблица 9.6 – Расчёт затрат на основные и вспомогательные материалы

Наименование материала	Единицы измерения	Потребленное количество	Цена за единицу, руб.	Сумма, руб.
Пачка бумаги	Шт.	1	254	254
CD-диск	Шт.	1	40	40
Конверт для CD-диска	Шт.	1	6	6
Итого затраты на основные (с учетом ТЗР) и вспомогательные материалы: 362,01 руб.				

9.3.5 Расходы на электроэнергию

Статья включает затраты по электроэнергии на технологические нужды. В настоящее время тариф на электроэнергию для населения г. Томска на 2017 год составляет 2,17 руб./ кВт ч. Тариф введен приказом от 23.12.2016 г. № 6-840 «О тарифах на электрическую энергию для населения и потребителей, приравненных к категории население по Томской области на 2017 год» [40], принятый департаментом тарифного регулирования Томской области.

Результаты расчётов приведены в 9.7.

9.3.6 Накладные расходы

Результаты расчёта накладных расходов приведены в таблице 9.8.

9.3.7 Сводная смета затрат

На основании всех произведённых расчётов составим сводную смету затрат на выполнение работы в виде таблицы 9.9.

Таблица 9.7 – Затраты на электроэнергию

Наименование прибора или оборудования	Количество, шт.	Потребляемая мощность, кВт.	Часы работы	Тариф за 1 кВт-час, руб.	Стоимость электроэнергии, руб.
Ноутбук	1	0,05	522	2,17	56,63
Принтер	1	0,1	5	2,17	1,09
Освещение	1	0,6	522	2,17	679,64
Всего: 737,36 руб.					

Таблица 9.8 – Накладные расходы

Услуга	Количество	Стоимость одной единицы, руб.	Сумма затрат, руб.
Переплет	1 шт.	50	50
Транспортные расходы	10 поездок	18	180
Итого: 230 руб.			

Таблица 9.9 – Сводная смета затрат

Наименование статей затрат	Всего, руб.
ФОТ со страховыми взносами	36748,94
Основные и вспомогательные материалы	362,01
Амортизационные отчисления	3528
Затраты на электроэнергию	737,36
Накладные расходы	230
Итоговая себестоимость работ: 41606,31 руб.	

9.4 Научно-технический эффект

Количественная оценка научно-технического уровня может быть произведена путём расчёта результативности участников разработки по формуле:

$$K_{ну} = \sum_{i=1}^{n} (K_{\partial y} \cdot d_i),$$

где $K_{ну}$ – коэффициент научного или научно-технического уровня;

$K_{\partial yi}$ – коэффициент достигнутого уровня *i*-го фактора;

d_i – значимость i-го фактора;

n – количество факторов.

Весовые коэффициенты *d* для каждого из факторов устанавливались экспертным путём. При этом сумма коэффициентов значимости по всем факторам равна единице. Коэффициенты достигнутого уровня факторов также установлены экспертным путём.

Таблица 9.10 – Оценка научно-технического уровня разработки

Показатели	Значимость показателя	Достигнутый уровень	Значение *i*-ого фактора
	d_i	$K_{\partial yi}$	$K_{\partial yi}d_i$
Новизна полученных или предполагаемых результатов	0,375	0,7	0,2625
Перспективность использования результатов	0,300	0,8	0,24
Завершенность полученных результатов	0,125	1	0,125
Масштаб возможной реализации полученных результатов	0,200	0,6	0,12
Результативность	$K_{ну}=\sum_{i=1}^{4}(K\cdot d_i)=0{,}7475$		

Рассчитанный коэффициент научно-технической результативности равен 0,7475. Полученное значение достаточно высоко, что говорит об эффективности проведённых работ выше среднего, однако отмечается необходимость дальнейшего развития проекта для достижения завершённости полученных результатов.

Заключение

В ходе преддипломной практики были выполнены все поставленные задачи:

– проведен обзор актуальных на сегодняшний день информационных источников в области деанонимизации авторов программного обеспечения по его исходному коду;

– построена модель процесса определения авторства исходного кода;

– сформирован и обработан набор данных, состоящий из трех подвыборок, имеющих характерные особенности, которые оказывают влияние на точность классификации;

– произведена программная реализация разработанной модели и ее тестирование;

– разработан интерфейс на основе технологии Jupyter Notebook;

– проведены вычислительные эксперименты и анализ полученных результатов.

Итогом практики является разработанное программное обеспечение «WhoseCppCode» на языке программирования Python, предназначенное для сбора и обработки данных, классификации авторов исходного кода на языке C/C++, визуализации полученных результатов при помощи интерфейса. Основной программный модуль может быть использован отдельно при разработке различных систем, интерфейсов и программ, а также дальнейших научных исследований задачи деанонимизации авторов исходного кода.

Список использованных источников

1 Образовательный стандарт ВУЗа ОС ТУСУР 01-2013 [Электронный ресурс]. — Режим доступа: https://storage.tusur.ru/files/40668/rules_tech_01-2013.pdf, свободный (дата обращения: 28.05.2017).

2 Google C++ Style Guide [Электронный ресурс]. — Режим доступа: https://google.github.io/styleguide/cppguide.html, свободный (дата обращения: 28.04.2017).

3 C++ Programming Style Guidelines [Электронный ресурс]. — Режим доступа: http://geosoft.no/development/cppstyle.html, свободный (дата обращения: 28.04.2017).

4 Кафедра комплексной информационной безопасности электронно-вычислительных систем [Электронный ресурс]. — Режим доступа: доступа: http://kibevs.tusur.ru/pages/kafedra/index, свободный (дата обращения: 01.05.2017).

5 Source Code Authorship Analysis For Supporting The Cybercrime Investigation Process. Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis [Электронный ресурс]. — Режим доступа: http://www.icsd.aegean.gr/lecturers/stamatatos/papers/ICETE2005.pdf, свободный (дата обращения: 17.02.2017).

6 Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method. Georgia Frantzeskou, Efstathios Stamatatos, Stefanos Gritzalis [Электронный ресурс]. — Режим доступа: https://www.semanticscholar.org/paper/Identifying-Authorship-by-Byte-Level-N-Grams-The-Frantzeskou-Stamatatos/3b2531ea2685b9fb9abf071d119974ac3405874d, свободный (дата обращения: 15.02.2017).

7 Using Classification Techniques to Determine Source Code Authorship. Brian N. Pellin Computer Sciences Department University of Wisconsin [Электронный ресурс]. — Режим доступа: https://pdfs.semanticscholar.org/f9aa/790191a50bed02a877e1696c7bb71ea9f33a.pdf, свободный (дата обращения: 25.02.2017).

8 Source Code Authorship Attribution using n-grams. Steven Burrows, S.M.M. Tahaghoghi [Электронный ресурс]. — Режим доступа: https://pdfs.semanticscholar.org/79a2/1998c2f0afe2c616c01d590d6d0f6e16e9eb.pdf, свободный (дата обращения: 23.02.2017).

9 Source Code Authorship Attribution. Steven Burrows [Электронный ресурс]. — Режим доступа: http://researchbank.rmit.edu.au/eserv/rmit:10828/Burrows.pdf, свободный (дата обращения: 23.02.2017).

10 Application of information retrieval techniques for source code authorship attribution. S. Burrows, A. Uitdenbogerd, T. Urpin [Электронный ресурс]. — Режим доступа: https://www.researchgate.net/publication/220787332_Application_of_Information_Retrieval_Techniques_for_Source_Code_Authorship_Attribution, свободный (дата обращения: 23.02.2017).

11 De-anonymizing Programmers via Code Stylometry. Aylin Caliskan-Islam, Drexel University; Richard Harang, U.S. Army Research Laboratory; Andrew Liu, University of Maryland; Arvind Narayanan, Princeton University; Clare Voss, U.S. Army Research Laboratory; Fabian Yamaguchi, University of Goettingen; Rachel Greenstadt, Drexel University [Электронный ресурс]. — Режим доступа: https://www.usenix.org/system/files/conference/usenixsecurity15/sec15-paper-caliskan-islam.pdf, свободный (дата обращения: 18.02.2017).

12 Random Forest. Applied Multivariate Statistics — Spring 2012 [Электронный ресурс]. — Режим доступа: http://stat.ethz.ch/education/semesters/ss2012/ams/slides/v10.2.pdf, свободный (дата обращения: 17.02.2017).

13 Git Blame Who?: Stylistic Authorship Attribution of Small, Incomplete Source Code Fragments. Edwin Dauber, Aylin Caliskan, Richard Harang, Rachel Greenstadt [Электронный ресурс]. — Режим доступа: https://arxiv.org/abs/1701.05681, свободный (дата обращения: 10.03.2017).

14 Хэзфилд Р. Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста / Р. Хэзфилд, Л. Кирби. — М: . ДиаСофт, 2001. – 736 с.

15 Макросы (C/C++) [Электронный ресурс]. — Режим доступа: https://msdn.microsoft.com/ru-ru/library/503x3e3s.aspx, свободный (дата обращения: 23.02.2017).

16 Ключевые слова C++ [Электронный ресурс]. — Режим доступа: http://ru.cppreference.com/w/cpp/keyword, свободный (дата обращения: 12.03.2017).

17 Методы классификации и прогнозирования. Деревья решений. ИНТУИТ [Электронный ресурс]. — Режим доступа: http://www.intuit.ru/studies/courses/6/6/lecture/1740, свободный (дата обращения: 01.05.2017).

18 Алгоритм AdaBoost [Электронный ресурс]. — Режим доступа: http://www.machinelearning.ru/wiki/index.php?title=AdaBoost, свободный (дата обращения: 27.04.2017).

19 Extremely randomized trees. Pierre Geurts, Damien Ernst, Louis Wehenkel [Электронный ресурс]. — Режим доступа: https://pdfs.semanticscholar.org/336a/165c17c9c56160d332b9f4a2b403fccbdbfb.pdf, свободный (дата обращения: 28.04.2017).

20 The Jupyter Notebook [Электронный ресурс]. — Режим доступа: http://jupyter.org/, свободный (дата обращения: 02.04.2017).

21 GitHub [Электронный ресурс]. — Режим доступа: https://github.com/, свободный (дата обращения: 10.05.2017).

22 Scikit-learn — Machine Learning in Python [Электронный ресурс]. — Режим доступа: http://scikit-learn.org/stable/, свободный (дата обращения: 02.03.2017).

23 Plotly Python Library [Электронный ресурс]. — Режим доступа: https://plot.ly/python/, свободный (дата обращения: 25.02.2017).

24 NumPy [Электронный ресурс]. — Режим доступа: http://www.numpy.org/, свободный (дата обращения: 25.02.2017).

25 SciPy [Электронный ресурс]. — Режим доступа: https://www.scipy.org/, свободный (дата обращения: 25.02.2017).

26 Python Data Analysis Library [Электронный ресурс]. — Режим доступа: http://pandas.pydata.org/, свободный (дата обращения: 25.02.2017).

27 ipywidgets: Interactive HTML Widgets [Электронный ресурс]. — Режим доступа: https://github.com/jupyter-widgets/ipywidgets, свободный (дата обращения: 09.03.2017).

28 ГОСТ 19.301-79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению [Электронный ресурс]. — Режим доступа: http://xn—-etbwobqde.xn–p1aiGOST_Edinaya-sistema-programmnoy-dokumentatsii-Programma-i-metodika-ispitaniy-Trebovaniya-k-soderganiyu-i-19301-79_2088.html, свободный (дата обращения: 28.05.2017).

29 Перекрёстная проверка [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/, свободный (дата обращения: 01.05.2017).

30 Code Jam Language Stats [Электронный ресурс]. — Режим доступа: https://www.go-hero.net/jam/16, свободный (дата обращения: 10.02.2017).

31 GitHub Dominates the Forges [Электронный ресурс]. — Режим доступа: https://github.com/blog/865-github-dominates-the-forges, свободный (дата обращения: 10.05.2017).

32 Оценка классификатора (точность, полнота, F-мера) [Электронный ресурс]. — Режим доступа: http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html, свободный (дата обращения: 17.02.2017).

33 ГОСТ 12.0.003-2015 Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация [Электронный ресурс]. — Режим доступа: http://meganorm.ru/Data2/1/4293754/4293754317.pdf, свободный (дата обращения: 31.05.2017).

34 СанПиН 2.2.2/2.4.1340-03 Гигиенические требования к персональным электронно-вычислительным машинам и организации работы [Электронный ресурс]. — Режим доступа: http://docs.cntd.ru/document/901865498, свободный (дата обращения: 31.05.2017).

35 СанПиН 2.2.4.548-96 Гигиенические требования к микроклимату производственных помещений (утв. постановлением Госкомсанэпиднадзора РФ от 1 октября 1996 г. № 21) [Электронный ресурс]. — Режим доступа: http://www.vashdom.ru/sanpin/224548-96/, свободный (дата обращения: 31.05.2017).

36 ГОСТ ИСО 8895-2002 Освещение рабочих систем внутри помещений [Электронный ресурс]. — Режим доступа: http://www.internet-law.ru/gosts/gost/5955/, свободный (дата обращения: 31.05.2017).

37 РД 153-34.0-03.2.98-2001 Типовая инструкция по охране труда для пользователей персональными электронно-вычислительными машинами (ПЭВМ) в электроэнергетике [Электронный ресурс]. — Режим доступа: http://forca.ru/instrukcii/dolzhnostnye/instrukciya-po-ohrane-truda-dlya-polzovateley-pevm.html, свободный (дата обращения: 31.05.2017).

38 Положение об оплате труда работников университета. Приказ ректора от 22.03.2013 г. № 3106. с изменениями от 09.12.2013 г. № 14249 [Электронный ресурс]. — Режим доступа: http://old.tusur.ru/ru/education/documents/inside/doc-table.html, свободный (дата обращения: 29.05.2017).

39 О классификации основных средств, включаемых в амортизационные группы: постановление Правительства РФ № 1 от 1 января 2002 г. [Электронный ресурс]. — Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_34710/, свободный (дата обращения: 29.05.2017).

40 Приказ от 23.12.2016 г. № 6-840 «О тарифах на электрическую энергию для населения и потребителей, приравненных к категории население по Томской области на 2017 год» [Элек-

тронный ресурс]. — Режим доступа: http://energovopros.ru/spravochnik/elektrosnabzhenie/tarify-na-elektroenergiju/tomskaya_oblast/39310/, свободный (дата обращения: 29.05.2017).

Приложение А

(Обязательное)

Компакт-диск

Компакт-диск содержит:

– электронную версию пояснительной записки в форматах *.tex и *.pdf;

– итоговую презентацию результатов работы в форматах *.pptx и *.pdf;

– актуальную версию программы, реализованную на языке программирования Python, для определения авторсва исходного кода программ на языке C/C++;

– тестовые данные для работы с программой.

Приложение Б

(Справочное)

Сравнительный обзор информационных источников

Таблица Б.1 — Обзор источников

Название работы	Авторы, год публикации	Методы, использованные в работе	Описание данных	Достигнутая точность классификации	Язык программирования
Using classification techniques to determine source code authorship [7]	B. Pellin, 2008	ACT, SVM	4 схожие программы, 2 автора	67 — 88 %	Java
Source code authorship attribution using n-grams [8]	S. Burrows, S. Tahaghoghi, 2007	N-граммы	Выборка из 1640 файлов исходного кода и 100 авторов	67 %	C
Identifying Authorship by Byte-Level N-Grams: The Source Code Author Profile (SCAP) Method [6]	G. Frantzeskou, E. Stamatatos, S. Gritzalis, 2007	Составление профиля программиста на основе статистических метрик, подсчет отклонения от профиля	Не указано	88 % для C++, 100 % для Java	Java, C++
Application of information retrieval techniques for source code authorship attribution [10]	S. Burrows, A. Uitdenbogerd, T. Urpin, 2009	N-граммы, рейтинговые схемы	100 авторов, классифицировались по 10, 1579 программных файлов	77 %	C

Продолжение таблицы Б.1

Название работы	Авторы, год публикации	Методы, использованные в работе	Описание данных	Достигнутая точность классификации	Язык программи-рования
De-anonymizing Programmers via Code Stylometry [11]	A. Caliskan-Islam, R. Harang, A. Liu, F. Yamaguchi, 2015	Статистический подсчет признаков, нечеткие АСТ	250 авторов, 1600 файлов	94 — 98 %	C/C++, Python
Git Blame Who?: Stylistic Authorship Attribution of Small, Incomplete Source Code Fragments [13]	A. Caliskan-Islam, E. Dauber, R. Harang, R. Greenstadt, 2017	Калибровочные кривые, нечеткие АСТ, классифика-тор Random Forest	Некомпили-руемые неполные образцы кода с ресурса GitHub	70 — 100 %	C/C++

Приложение В

(Справочное)

Описание лексических признаков

Таблица В.1 — Описание различных лексических признаков

Группа признаков	Признак	Обозначение	Определение
Стиль комментирования	Число однострочных комментариев	ln_inline_ comments	Натуральный логарифм отношения числа однострочных комментариев к длине файла в символах
	Число многострочных комментариев	ln_multiline_ comments	Натуральный логарифм отношения числа многострочных комментариев к длине файла в символах
	Число комментариев	ln_comments	Натуральный логарифм отношения числа комментариев к длине файла в символах
Стиль разметки	Число пробелов	ln_spaces	Натуральный логарифм отношения числа пробелов к длине файла в символах
	Число символов табуляции	ln_tabs	Натуральный логарифм отношения числа символов табуляции к длине файла в символах
	Число переводов строки	ln_newlines	Натуральный логарифм отношения числа переводов строки к длине файла в символах
	Коэффициент пробельных символов	whitespace_ ratio	Натуральный логарифм отношения суммы всех пробельных символов (пробелов, символов табуляции, переводов строки) к длине файла в символах
Стиль расстановки фигурных скобок	Число одиночных раскрывающихся скобок	ln_open_brace_ alone	Натуральный логарифм отношения числа раскрывающихся скобок, одиночных в строке, к длине файла в символах

Продолжение таблицы В.1

Группа признаков	Признак	Обозначение	Определение
Стиль расстановки фигурных скобок	Число раскрывающихся скобок, первых в строке	ln_open_ brace_first	Натуральный логарифм отношения числа раскрывающихся скобок, после которых следует код, к длине файла в символах
	Число раскрывающихся скобок, последних в строке	ln_open_ brace_last	Натуральный логарифм отношения числа раскрывающихся скобок, которым предшествует код, к длине файла в символах
	Число закрывающихся скобок, одиночных в строке	ln_closing_ brace_alone	Натуральный логарифм отношения числа зарывающихся скобок, одиночных в строке, к длине файла в символах
	Число закрывающихся скобок, первых в строке	ln_closing_ brace_first	Натуральный логарифм отношения числа закрывающихся скобок, после которых следует код, к длине файла в символах
Дополнительные признаки	Число закрывающихся скобок, последних в строке	ln_closing_ brace_last	Натуральный логарифм отношения числа закрывающихся скобок, которым предшествует код, к длине файла в символах
	Число макросов	ln_macros	Натуральный логарифм отношения числа макросов к длине файла в символах
	Число строк кода	lines_of_code	Число строк кода, не включающее пустые строки

Приложение Д

(Справочное)

Руководство программиста

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ
ОПРЕДЕЛЕНИЯ АВТОРСТВА ИСХОДНОГО КОДА ПРОГРАММ НА ЯЗЫКЕ C/C++
«WhoseCppCode»

Руководство программиста

Листов 9

Аннотация

В данном руководстве описана структура, принципы работы программного обеспечения «WhoseCppCode». Определены условия, необходимые для эффективного функционирования программного обеспечения. Указаны возможные входные и выходные данные. Описан алгоритм работы программного обеспечения.

Приложение Е

(Справочное)

Руководство пользователя

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ
ОПРЕДЕЛЕНИЯ АВТОРСТВА ИСХОДНОГО КОДА ПРОГРАММ НА ЯЗЫКЕ C/C++
«WhoseCppCode»

Руководство пользователя

Листов 6

1 Введение

1.1 Область применения

Требования настоящего документа применяются при:

- предварительных испытаниях системы;
- опытной эксплуатации;
- приемочных испытаниях;
- эксплуатации на предприятиях.

1.2 Краткое описание возможностей

Программное обеспечение (ПО) «WhoseCppCode» предназначено для определения авторства программ на языке C/C++ по исходному коду и может быть использовано организациями, занимающимися решением вопросов информационной безопасности, лицензирования ПО, интеллектуальной собственности и расследования инцидентов, связанных с применением вредоносного ПО. Программа состоит из программного модуля, реализующего заявленный функционал, и интерфейса, предназначенного для визуализации ввода и вывода данных, а также удобной работы с возможностями основного модуля. При этом интерфейс не является обязательным, программа может быть использована в качестве модуля при разработке иных автоматизированных систем.

ПО «WhoseCppCode» предоставляет следующие возможности:

- обработка файлов исходного кода на языке C/C++;
- сбор данных с ресурса GitHub;
- построение модели классификации авторов программного обеспечения;
- формирование отчетности в форматах *.json и *.csv;
- визуализация результатов классификации.

1.3 Уровень подготовки пользователя

Пользователь ПО «WhoseCppCode» должен иметь опыт работы с ОС Linux, базовые навыки программирования, а также обладать следующими знаниями:

- знать соответствующую предметную область;

– понимать основы машинного обучения, построения и оценки моделей классификации.

Квалификация пользователя должна позволять осуществлять сбор и анализ данных.

1.4 Перечень эксплуатационной документации, с которыми необходимо ознакомиться пользователю

Для начала использования системы пользователю предварительно необходимо ознакомиться с содержанием пояснительной записки к проекту, а также настоящим руководством.

2 Назначение

ПО «WhoseCppCode» предназначено для сбора и анализа файлов исходного кода на языке программирования C/C++ для дальнейшей идентификации авторов соответствующих программ.

Работа с ПО «WhoseCppCode» доступна всем пользователям с доступом к предварительно установленной и настроенной рабочей программной среде, реализованной на ПЭВМ, специально предназначенном сервере или с помощью средств виртуализации.

3 Подготовка к работе

3.1 Состав и содержание дистрибутивного носителя данных

Для работы с ПО «WhoseCppCode» необходимо следующее программное обеспечение:

– виртуальная машина с ОС Ubuntu (16.04 и выше) с доступом к глобальной сети Интернет;

– программная среда с установленными зависимостями (библиотеками);

– браузер (Mozilla FireFox).

3.2 Порядок загрузки данных и программ

3.3 Порядок проверки работоспособности

4 Описание операций

4.1 Описание функций, выполняемых средством автоматизации

. Для каждой операции обработки данных указывают:

1.наименование;

2.условия, при соблюдении которых возможно выполнение операции;

3.подготовительные действия;

4.основные действия в требуемой последовательности;

5.заключительные действия;

6.ресурсы, расходуемые на операцию.

5 Аварийные ситуации

5.1 Действия в случае обнаружения ошибок данных

7 Рекомендации по освоению

описание контрольного примера, правила его запуска и выполнения