Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии
Направленность (профиль) образовательной программы Безопасность информационных систем

ДОПУСТИТЬ К ЗАЩИТЕ
Зав. кафедрой
_____ А.В. Бушманов
«____»_____2022 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка информационно-обучающего программного обеспечения по дисциплине «Криптографические методы защиты информации»

Исполнитель
студент группы 855-об _____ С.О. Капитонов
(подпись, дата)

Руководитель
доцент, канд. техн. наук _____ Л.В. Никифорова
(подпись, дата)

Консультант
по безопасности и экологичности
доцент, канд. техн. наук _____ А.Б. Булгаков
(подпись, дата)

Нормоконтроль
инженер кафедры _____ В.Н. Адаменко
(подпись, дата)

Благовещенск 2022

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики

Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Зав. кафедрой

_____А.В. Бушманов

«_____» _____ 2022 г.

З А Д А Н И Е

К бакалаврской работе студента Капитонова С.О.

1. Тема выпускной квалификационной работы: Разработка информационно-обучающего программного обеспечения по дисциплине «Криптографические методы защиты информации»
(утверждена приказом от 05.04.2022 № 679-уч)

2. Срок сдачи студентом законченной работы: _____

3. Содержание бакалаврской работы (перечень подлежащих разработке вопросов): анализ объекта исследования; разработка и эксплуатация программного продукта; проектирование программного продукта; безопасность и экологичность.

4. Перечень материалов приложения (UML-диаграмма классов, UML диаграмма использования, наличие чертежей, таблиц, графиков, схем, программных продуктов, иллюстративного материала и т.п.):

5. Дата выдачи задания: 07.02.2022

Руководитель бакалаврской работы: доцент, канд.техн.наук Л.В. Никифорова.

Задание принял к исполнению(дата): 07.02.2022 С.О. Капитонов

РЕФЕРАТ

Бакалаврская работа содержит 58с., 21 рисунок, 28 источников, 1 приложение.

ОБРАЗОВАТЕЛЬНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON, БИБЛИОТЕКА PYSIDE6, ФРЕЙМВОРК QT, АЛГОРИТМЫ ШИФРОВАНИЯ, КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ.

Работа посвящена разработке и введению в эксплуатацию информационно-обучающего программного продукта по дисциплине «Криптографические методы защиты информации». Работа проходила в несколько этапов:

- Постановка задачи;
- Анализ требований и определение спецификаций;
- Проектирование;
- Реализация.

Содержание

Введение

В последнее время важную роль в подготовке кадров играют компьютерные технологии, которые позволяют по-новому интерпретировать содержание учебных материалов и организовывать их изучение. В то же время они выступают в качестве новых интерактивных средств обучения, обладающих многими достоинствами и обеспечивающих качественное изменение методов, форм и содержания обучения.

Несмотря на постоянный интерес исследователей к вопросам информатизации образования, следует отметить, что в современной педагогической и методической литературе еще недостаточно освещены многие теоретические и практические вопросы применения электронных технологий в образовательном процессе, в частности, вопросы, касающиеся развития программ электронного обучения для студентов высших учебных заведений.

Практика показала, что использование электронных технологий в процессе обучения учащихся дает ряд преимуществ, повышающих эффективность учебного процесса.

Объект исследования: ФГБОУ ВО «Амурский государственный университет».

Предмет исследования: разработка программного продукта для преподавания дисциплины «Криптографические методы защиты информации».

Целью данной работы является разработка информационно-обучающего программного обеспечения по дисциплине «Криптографические методы защиты информации» для студентов бакалавриата и среднего профессионального образования.

В задачи работы входили:

- анализ предметной области и учебного плана;
- анализ аналогов;

- определение основных требований к программе;
- выбор пакета программного обеспечения и инструментов для разработки программного продукта;
- дизайн графического интерфейса;
- разработка программного продукта.

Создание данного программного обеспечения позволит упростить и модернизировать преподавание дисциплины, используя компьютерные технологии.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.

1.1. Анализ организационной структуры учреждения

Полное наименование образовательной организации: Федеральное государственное бюджетное образовательное учреждение высшего образования «Амурский государственный университет». Сокращенное наименование образовательной организации: ФГБОУ ВО "АмГУ".

Организационная структура учреждения представлена на рисунке 1. Структура имеет традиционный иерархический вид.

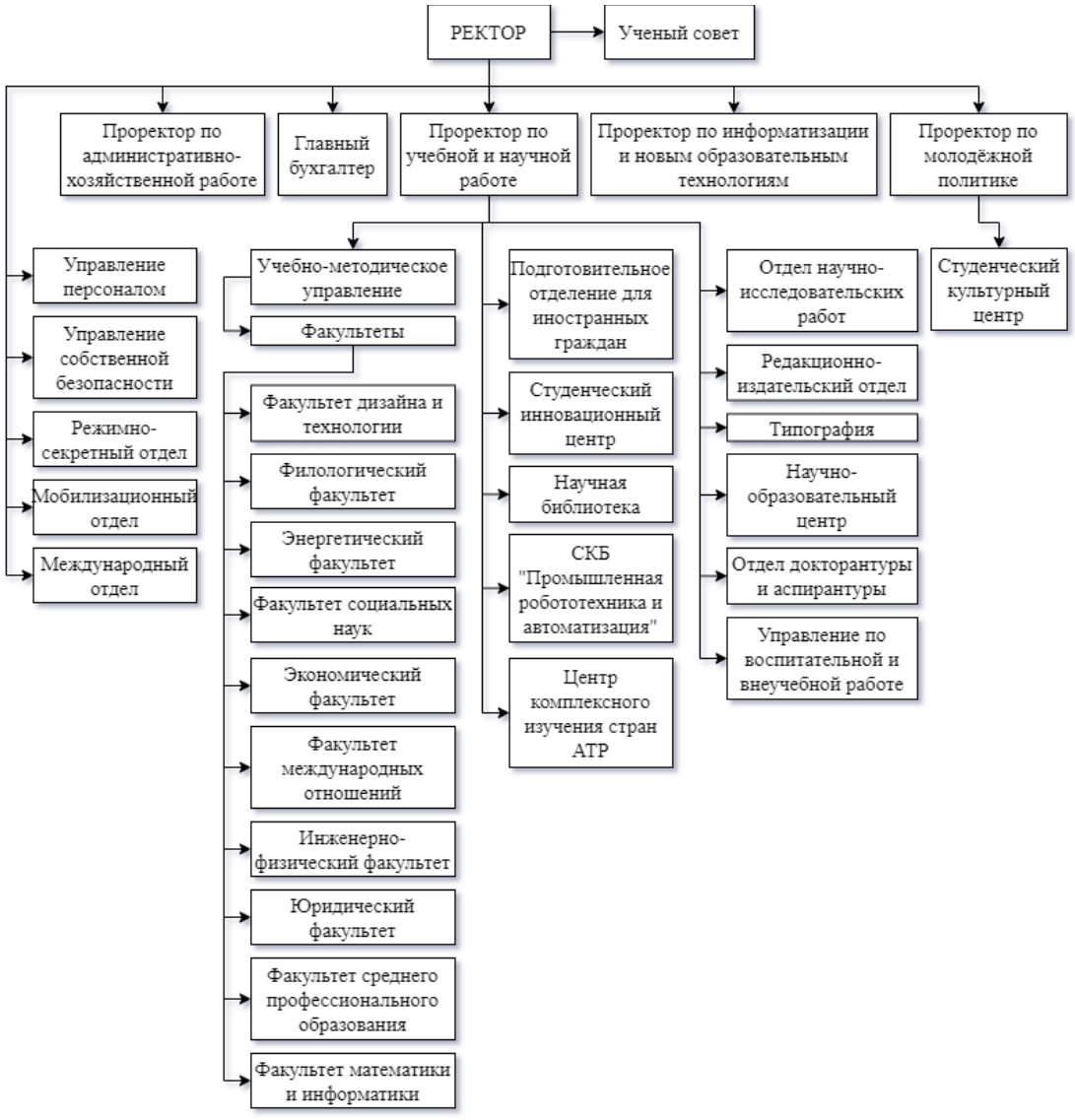


Рисунок 1 – Упрощённая организационная структура ФГБОУ ВО «АмГУ»

Дисциплина «Криптографические методы защиты информации» преподаётся кафедрой информационных и управляющих систем, которая находится в составе факультета математики и информатики.

В настоящее время кафедра выпускает специалистов по трем направлениям подготовки бакалавриата и одно направление подготовки магистратуры:

1. 09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления»
2. 09.03.02 «Информационные системы и технологии», профиль «Безопасность информационных систем»
3. 09.04.04 «Программная инженерия», профиль «Управление разработкой программного обеспечения»

Ведется подготовка аспирантов по направлению 09.06.01 «Математическое моделирование, численные методы и комплексы программ» по профилю «Системный анализ, управление и обработка информации».

На кафедре работает 16 сотрудников:

- Заведующий кафедрой;
- 2 инженера;
- 14 преподавателей (из них 3-совместместители).

1.2. Анализ образовательной программы

В соответствии со статьёй 10 Федерального закона от 29.12.2012 N 273-ФЗ «Об образовании в Российской Федерации» в систему образования Российской Федерации включены федеральные государственные образовательные стандарты (ФГОС) и федеральные государственные требования, образовательные стандарты и самостоятельно устанавливаемые требования, образовательные программы различных вида, уровня и (или) направленности.

Статья 13 указанного выше федерального закона указывает на общие требования к реализации образовательных программ. Среди них можно выделить

возможность использования зачётных единиц для определения структуры профессиональных образовательных программ и трудоемкости их освоения. Зачетная единица представляет собой унифицированную единицу измерения трудоемкости учебной нагрузки обучающегося, включающую в себя все виды его учебной деятельности, предусмотренные учебным планом (в том числе аудиторную и самостоятельную работу), практику.

Количество зачетных единиц по основной профессиональной образовательной программе по конкретным профессии, специальности, направлению подготовки или научной специальности устанавливается соответствующими федеральными государственными образовательными стандартами, федеральными государственными требованиями, образовательными стандартами или самостоятельно устанавливаемыми требованиями.

Преподавание в ФГБОУ ВО «АмГУ» ведётся в соответствии с требованиями данного Федерального закона. Образовательная программа университета составляется на основании Федеральных государственных образовательных стандартов.

Дисциплина «Криптографические методы защиты информации» по ФГОС обязательна к преподаванию по направлению бакалавриата 10.03.01 «Информационная безопасность» и среднего профессионального образования 10.02.04 «Обеспечение информационной безопасности телекоммуникационных систем».

Также данная дисциплина на данный момент преподаётся по направлению 09.03.02 «Информационные системы и технологии» в рамках профиля «Безопасность информационных систем».

В учебном плане по направлению СПО 10.02.04 на дисциплину «Криптографическая защита информации» отводится минимум 265 и максимум 345 академических часов за всё время обучения. Из них:

- 110 часов отводится на лекционные занятия;
- 66 часов отводится на практические занятия;

– 52 часа отводится на лабораторные занятия;
– 20 часов отводится на курсовое проектирования;
– 17 часов отводится на промежуточную аттестацию;
– 46 часов отводится на самостоятельную работу;
– 34 часа отводится на консультации.

В учебном плане бакалавриата 09.03.02 на дисциплину «Криптографические методы защиты информации» отводится 6 зачётных единиц и 216 академических часов за всё время обучения. Из них:

– 36 часов отводится на лекционные занятия;
– 36 часов отводится на лабораторные занятия;
– 36 часов отводится на практические занятия;
– 72 часа отводится на самостоятельные работы;
– 36 часов отводится на контроль (форма контроля – экзамен).

Разрабатываемое программное обеспечение ориентировано на применение при проведении самостоятельных и практических работ, что составляет 190 академических часов.

Из этого можно судить об актуальности разработки и внедрения данного программного продукта в учебный процесс для данных направлений подготовки.

1.3. Обучающее программное обеспечение

1.3.1. Определение обучающего программного обеспечения

Обучающее или образовательное программное обеспечение представляет собой вид программного обеспечения, основная цель которого заключается в реализации образовательных целей. Оно включает в себя различные классы программ, например, интернет-сервисы для изучения языка, программное обеспечение для управления классом, справочное программное обеспечение, образовательные игры и среды и т. д. Цель всего этого программного обеспечения заклю-

чается в повышении эффективности и действенности какого-либо аспекта образования.

1.3.2. История создания обучающего программного обеспечения

История образовательного ПО уходит корнями в 1940-е годы, когда американские разработчики авиасимуляторов использовали аналоговый компьютер для моделирования приборов на борту самолетов.

До середины 1970-х годов образовательное ПО было напрямую связано с мейнфреймами, на которых оно было реализовано. В течение этих лет основоположниками образовательных компьютерных систем были: система PLATO (1960), разработанная в Иллинойском университете, и Система TICCIT (1969). Стоимость этих ранних терминалов превышала 10000 долларов, и образовательные учреждения не могли их себе позволить из-за высокой цены. Языки программирования того времени, такие как BASIC и Logo, можно тоже рассматривать как образовательные системы, разработанные для учащихся и начинающих пользователей компьютеров.

Система PLATO IV, созданная в 1972 году, имела ряд функций, которые позже стали в некотором роде стандартом в образовательном ПО для персональных компьютеров, а именно: растровые изображения, звуковое сопровождение, неклавиатурные устройства ввода, например - сенсорный экран.

Появление первых персональных компьютеров, таких как Altair 8800, созданный в 1975 году, оказало огромное влияние на программное обеспечение в целом. До 1975 года люди могли получить доступ к компьютерам только в вычислительных центрах или некоторых университетах. Но после выхода Altair 8800 компьютеры стали появляться дома и в школах. Рыночная стоимость Altair 8800 в то время не превышала 2000 долларов.

В начале 1980-х стали доступны такие компьютеры, как Commodore PET и Apple II, что привело к созданию компаний и некоммерческих организаций, про-

изводящих образовательное программное обеспечение. На тот момент существовало несколько ведущих компаний-разработчиков программного обеспечения, включая Brøderbund и The Learning Company. Из некоммерческих организаций можно выделить МЕСС. Эти и другие компании разработали целый спектр всевозможного образовательного ПО, первоначально написанного для компьютеров Apple II.

1.3.3. Классификация обучающего программного обеспечения

Подходов к классификации обучающих компьютерных программ существует множество, но единого мнения и общей классификации не существует.

По одному из подходов разделяют четыре вида обучающих программ:

- Тренировочные и контролирующие;
- Наставнические;
- Имитационные и моделирующие;
- Игровые;

Тренировочные и контролирующие программы обычно используются, когда теоретический материал уже изучен и используются для закрепления знаний. Такие программы предлагают обучающемуся ответить на вопросы и решить задачи. В конце программа подсчитывает количество правильно и неправильно решённых задач и по возможности ставит оценку.

Наставнические программы предлагают ученику теоретический материал, после изучения которого предлагаются контрольные задания для закрепления материала. При неправильных ответах может предлагаться повторное изучение теоретического материала.

Выделяют четыре различных типа наставнического ПО:

- *Линейные* – состоят из небольших познавательных блоков учебной информации и вопросами с вариантами ответов;
- *Разветвлённые* – при неправильном ответе ученику предлагается

ознакомиться с дополнительной информацией, которая поможет ему дать правильный ответ на поставленный вопрос;

- *Адаптивные* – позволяют обучающемуся самостоятельно выбирать уровень сложности преподаваемой информации и, если в этом есть необходимость, получать информацию из дополнительных источников;
- *Комбинированные* – включают в себя компоненты всех предыдущих типов.

Моделирующие программы основываются на вычислительных и графических возможностях ПК. В таких программах ученику даётся возможность наблюдения за определённым процессом, на который он может непосредственно влиять, изменяя установленные параметры, входные данные и др.

Игровые программы представляют собой видеоигру, играя в которую, ученик может расширить свои знания, сформировать познавательные навыки и даже самостоятельно открыть для себя некоторые правила и закономерности.

Разрабатываемый программный продукт относится к моделирующему программному обеспечению, так как позволяет наблюдать за работой шифров криптографической защиты и влиять на процесс, изменяя входные данные.

1.4. Криптография и криптографические шифры

1.4.1. Определение криптографии

Криптография - наука о способах преобразования (шифрования) информации с целью ее защиты от незаконных пользователей.

Криптография исторически использовалась для защиты передаваемых текстовых сообщений от несанкционированного доступа, известного только отправителю и получателю. Все методы шифрования были основаны на этой фундаментальной идее.

С усложнением информационных взаимодействий в человеческом обществе возникли новые задачи по их защите, часть из которых была решена за счет

развития криптографии и возникновению новых подходов и методов.

1.4.2. Классификация криптографических шифров

Классификация криптографических систем строится на основе следующих трех характеристик:

- число применяемых ключей;
- тип операций по преобразованию открытого текста в шифрованный;
- метод обработки открытого текста;

По *числу применяемых ключей* различают:

- Симметричные шифры;
- Асимметричные шифры.

Если отправитель и получатель используют один и тот же ключ, система шифрования называется *симметричной* или *системой с одним ключом*.

Если отправитель и получатель используют разные ключи, система называется *асимметричной*, *системой с двумя ключами* или *схемой шифрования с открытым ключом*.

По *типу операций по преобразованию открытого текста в шифрованный* различают:

- Подстановочные шифры;
- Перестановочные шифры;
- Продукционные шифры.

В *подстановочных шифрах* шифрование основано на замещении каждого элемента открытого текста другим элементом;

В *перестановочных шифрах* шифрование основано на изменении порядка следования элементов открытого текста;

В *продукционных шифрах* шифрование основано на комбинации нескольких операций замены и перестановки.

Продукционные шифры применяются в большинстве реальных современных систем шифрования.

По *методу обработки открытого текста* различают:

- Блочные шифры;
- Поточные шифры.

Блочными называются шифры, в которых логической единицей шифрования является некоторый блок открытого текста, после преобразования которого получается блок шифрованного текста такой же длины.

Поточные шифры подразумевают шифрование всех элементов открытого текста последовательно, одного за другим (бит за битом, байт за байтом).

Блочные шифры были изучены более широко. Считается, что они имеют более широкую область применения, чем поточные. Большинство сетевых приложений, использующих традиционную схему шифрования, используют блочные шифры.

1.5. Анализ аналогов

Чтобы определить актуальность разработки программного обеспечения необходимо рассмотреть доступные аналоги.

1.5.1. Android приложение Cryptoy.

Данное мобильное приложение было разработано британской спецслужбой GCHQ в 2014 году для обучения школьников, студентов и всех остальных основам криптографии.

Программа оснащена интуитивно понятным красочным интерфейсом. Она обучает базовым криптографическим техникам, таким как шифр Цезаря (со сдвигом), шифр подстановки, шифр Виженера, шифр военной криптографической машины «Энигма». Пользователи могут применить эти шифры для кодирования собственных сообщений — и отправлять друзьям на расшифровку.

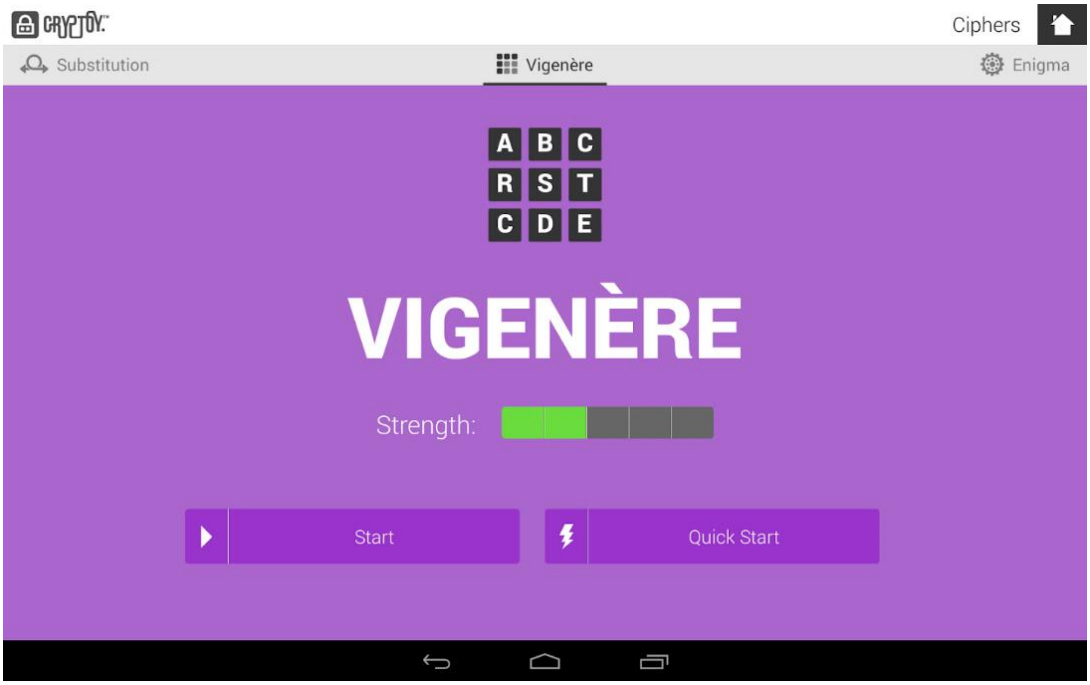


Рисунок 2 – Интерфейс программы Cryptoy

Главными недостатками программы являются:

- отсутствие поддержки русского языка;
- небольшой набор криптографических шифров для изучения;
- отсутствие обновлений с 2014 года;
- *невозможность загрузить программу с официальных источников*, так как она была удалена с официального сайта и магазина приложений Google Play.

1.5.2. Windows приложение CrypTool

CrypTool детально рассказывает о том, что такое криптография, какие существуют криптографические алгоритмы и как они работают. В программе реализовано более 400 криптографических алгоритмов. Пользователи могут настраивать алгоритмы по собственным параметрам. Графический интерфейс, онлайн-документация, аналитические инструменты и алгоритмы проекта CrypTool знакомят пользователей с областью криптографии. Также есть возможность легко комбинировать и выполнять криптографические функции для создания рабочих

процессов в CrypTool самостоятельно с помощью визуального программирования. Благодаря такому подходу сложные процессы можно легко визуализировать и тем самым лучше понять.

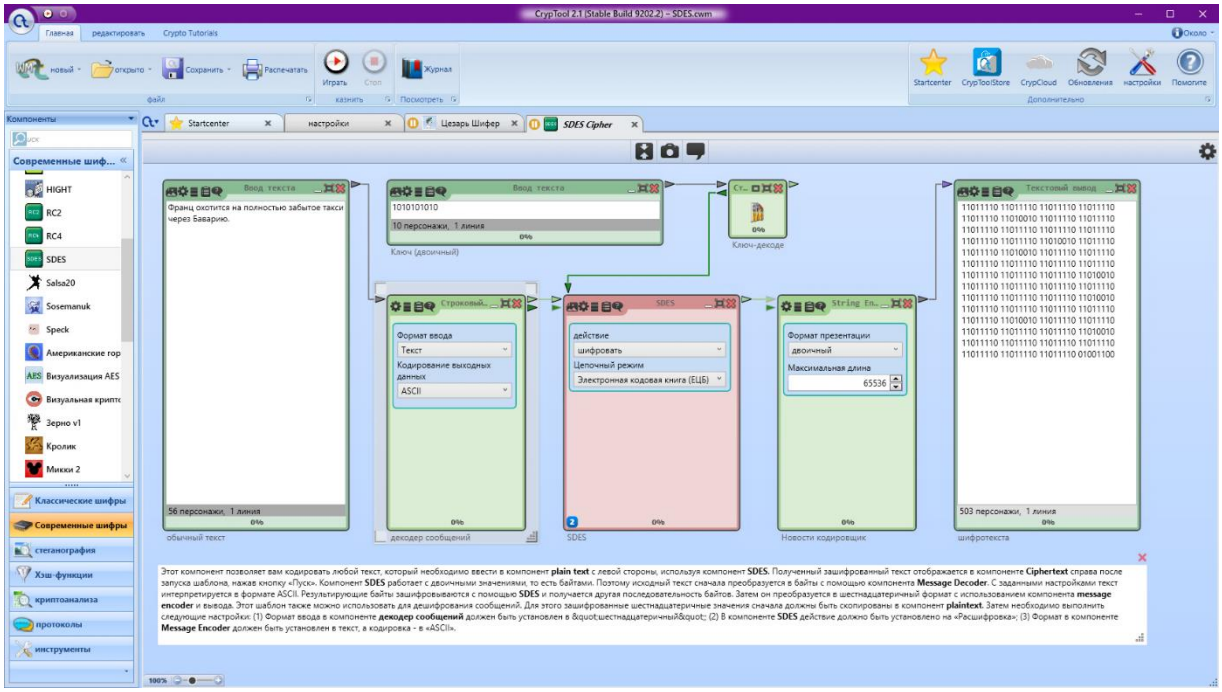


Рисунок 3 – Окно программы CrypTool

Несмотря на обширное количество реализованных в программе алгоритмов, она обладает определёнными недостатками:

– документация доступна только на английском языке;
– интерфейс программы может показаться пользователю слишком сложным;
– перевод на русский полон ошибок, которые указывают на использование машинного перевода при создании русской локализации;
– работа шифра не всегда расписывается по шагам, часто доступна только базовая информация о шифре.

1.4.3 Онлайн сервис CrypTool Online

CrypTool-Online предлагает увлекательную возможность знакомства с миром криптографии. Разнообразные шифры, методы кодирования и инструменты

анализа представлены вместе с иллюстрированными примерами. Делается акцент на том, чтобы объяснения были простыми для понимания для повышения общего интереса к криптографии и криптоанализу.

Основным плюсом сервиса является достаточно обширное количество реализованных шифров, а также пошаговое описание их работы (рисунок 5) и обширное количество настроек.

Из основных минусов можно выделить:

– отсутствие русского языка;
– необходимость в интернет-соединении;
– несмотря на наличие пошагового описания работы многих шифров, введённые данные никак не используются в процессе описания.

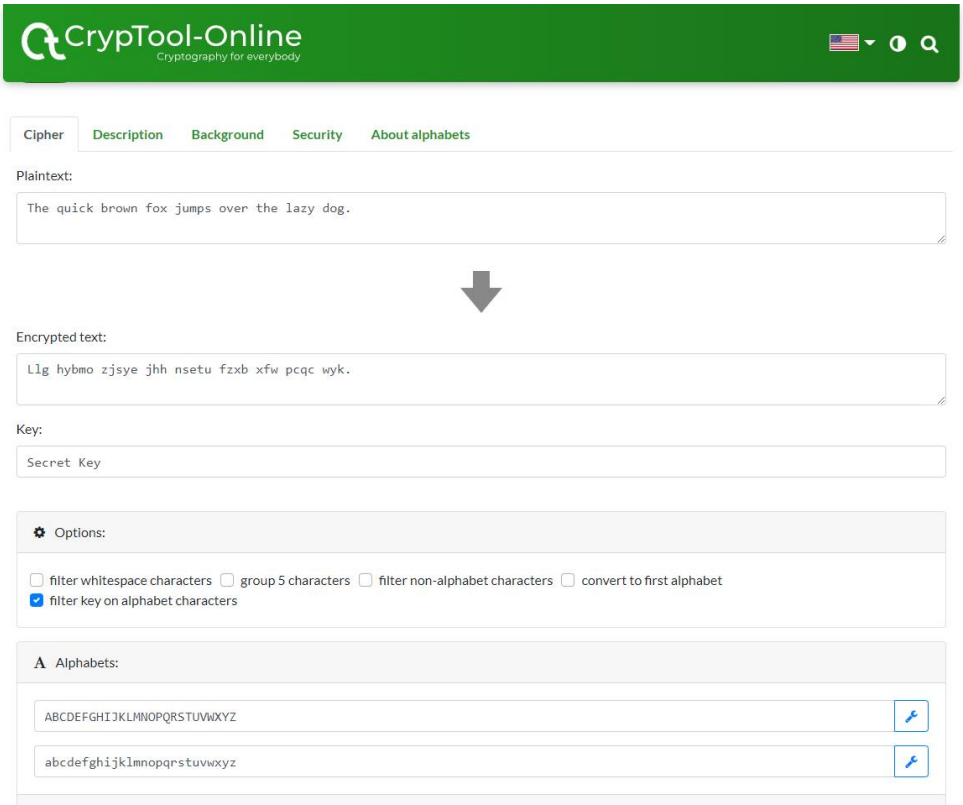
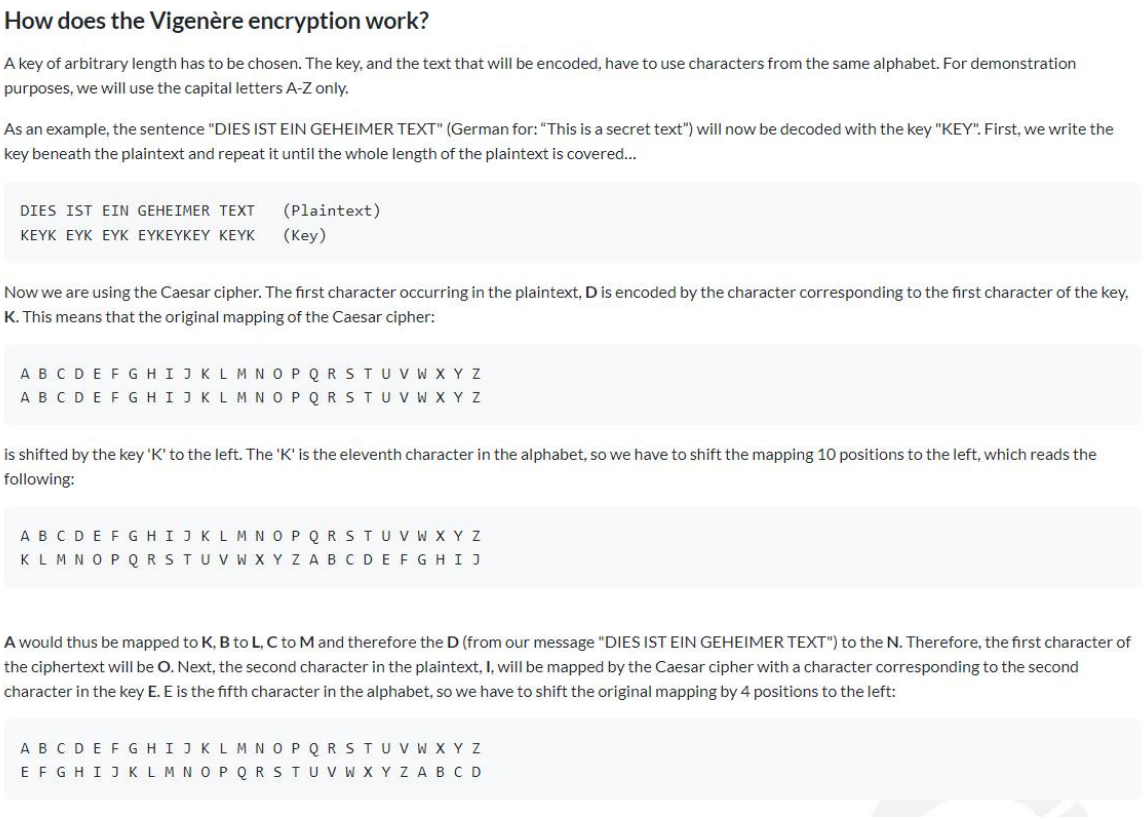


Рисунок 4 – Интерфейс веб-сайта CrypTool Online



How does the Vigenère encryption work?

A key of arbitrary length has to be chosen. The key, and the text that will be encoded, have to use characters from the same alphabet. For demonstration purposes, we will use the capital letters A-Z only.

As an example, the sentence "DIES IST EIN GEHEIMER TEXT" (German for: "This is a secret text") will now be decoded with the key "KEY". First, we write the key beneath the plaintext and repeat it until the whole length of the plaintext is covered...

```
DIES IST EIN GEHEIMER TEXT   (Plaintext)
KEYK EYK EYK EYKEYKEY KEYK   (Key)
```

Now we are using the Caesar cipher. The first character occurring in the plaintext, **D** is encoded by the character corresponding to the first character of the key, **K**. This means that the original mapping of the Caesar cipher:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

is shifted by the key 'K' to the left. The 'K' is the eleventh character in the alphabet, so we have to shift the mapping 10 positions to the left, which reads the following:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
K L M N O P Q R S T U V W X Y Z A B C D E F G H I J
```

A would thus be mapped to **K**, **B** to **L**, **C** to **M** and therefore the **D** (from our message "DIES IST EIN GEHEIMER TEXT") to the **N**. Therefore, the first character of the ciphertext will be **O**. Next, the second character in the plaintext, **I**, will be mapped by the Caesar cipher with a character corresponding to the second character in the key **E**. E is the fifth character in the alphabet, so we have to shift the original mapping by 4 positions to the left:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
```

Рисунок 5 – Описание работы шифра Виженера на сайте CrypTool Online

Таким образом, были сделаны выводы об актуальности разработки обучающей программы для изучения криптографии.

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

2.1. Выбор средств и инструментов разработки

2.1.1. Анализ языков программирования

C++ – Компилируемый, статически типизированный язык программирования общего назначения. Является одним из самых распространенных языков в мире. Отличается быстротой обработки и компиляции.

C# – Объектно-ориентированный язык программирования. Имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов, делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Python – Язык программирования общего назначения высокого уровня, предназначенный для увеличения производительности разработчиков, удобства читаемости кода и разработки веб-приложений. Синтаксис ядра Python минималистичен. Код Python разбит на функции и классы, которые можно объединить в модули.

Java – Строго типизированный объектно-ориентированный язык программирования. Приложения Java обычно транслируются в специальный байт-код. Преимущество этого способа выполнения программ заключается в том, что байт-код абсолютно независим от операционной системы и оборудования, что позволяет запускать Java-приложения на любом устройстве, для которого доступна соответствующая виртуальная машина. Еще одним важным свойством технологии Java является гибкая, свободная от рисков система безопасности, в которой выполнение программы полностью контролируется виртуальной машиной.

2.1.2. Анализ сред программирования

Microsoft Visual Studio – это интегрированная среда разработки. На данной IDE возможно создавать все виды программного обеспечения, от веб-приложений до мобильных приложений и компьютерных игр. Эта линейка программного обеспечения включает в себя множество инструментов для тестирования совместимости.

Поддерживаемые языки: Ajax, ASP.NET, DHTML, ASP.NET, JavaScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML и другие.

Стоимость: от 45 $ в месяц. Есть бесплатная версия (Community) для частного использования, студентов и создателей проектов с открытым исходным кодом.

Особенности:

- огромная коллекция всевозможных расширений, которая постоянно пополняется;
- технология автодополнения intellisense;
- возможность настроить рабочую панель;
- поддержка разделенного экрана (split screen);
- встроенный web-сервер;
- интуитивный стиль кодирования;
- возможности отладки;
- более высокая скорость разработки.

Из недостатков можно выделить тяжеловесность этой IDE и невозможность отладчика (Microsoft Visual Studio Debugger) отслеживать в коде режима ядра.

Даже небольшое редактирование может быть ресурсоемким, поэтому если нужно выполнить простую и быструю задачу, будет удобнее использовать более легкий редактор.

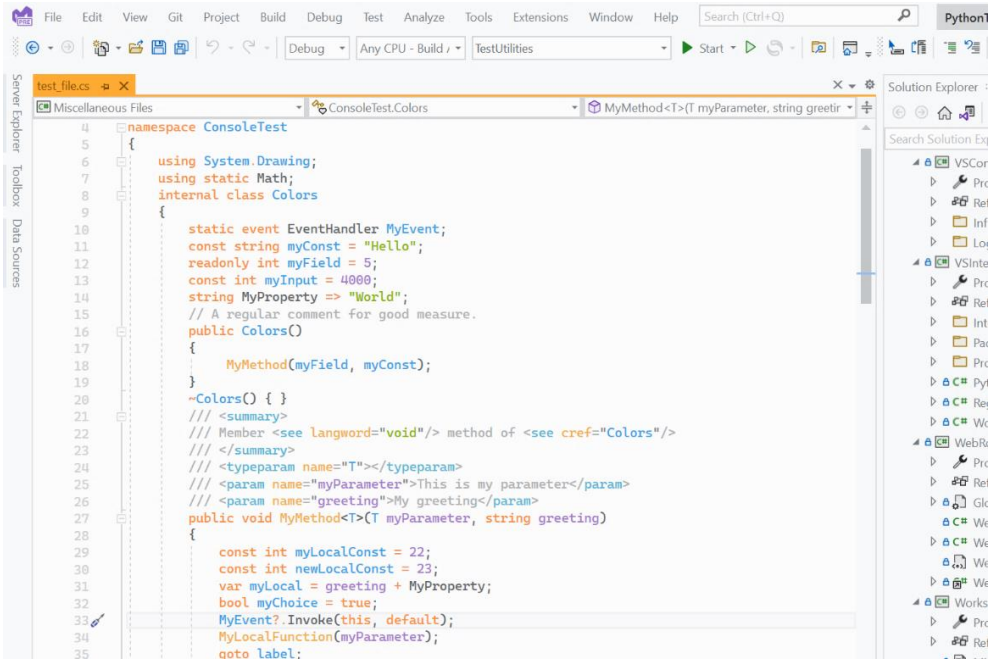


Рисунок 6 – Интерфейс программы MS Visual Studio

PyCharm – интегрированная среда разработки на языке Python, которая была разработана международной компанией JetBrains. Данная IDE распространяется по нескольким лицензиям, включая Community Edition, которая имеет несколько ограниченную функциональность. Сами разработчики характеризуют свой продукт как «самую интеллектуальную Python IDE с полным набором средств для эффективной разработки на языке Python».

Поддерживаемые языки: Python, Jython, Cython, IronPython, PyPy, AngularJS, Coffee Script, HTML/CSS, Django/Jinja2 templates, Gql, LESS/SASS/SCSS/HAML, Mako, Puppet, RegExp, Rest, SQL, XML, YAML и т.д.

Стоимость: от 199 $ в год. Есть бесплатная версия, работающая только с Python.

Преимущества:

- поддержка google app engine; ironpython, jython, cython, pypy wxpython, pyqt, pygtk и др;
- поддержка flask-фреймворка и языков mako и jinja2;
- редактор javascript, coffescript, html/css, sass, less, haml;

– интеграция с системами контроля версий (vcs);
– uml диаграммы классов, диаграммы моделей django и google app engine.

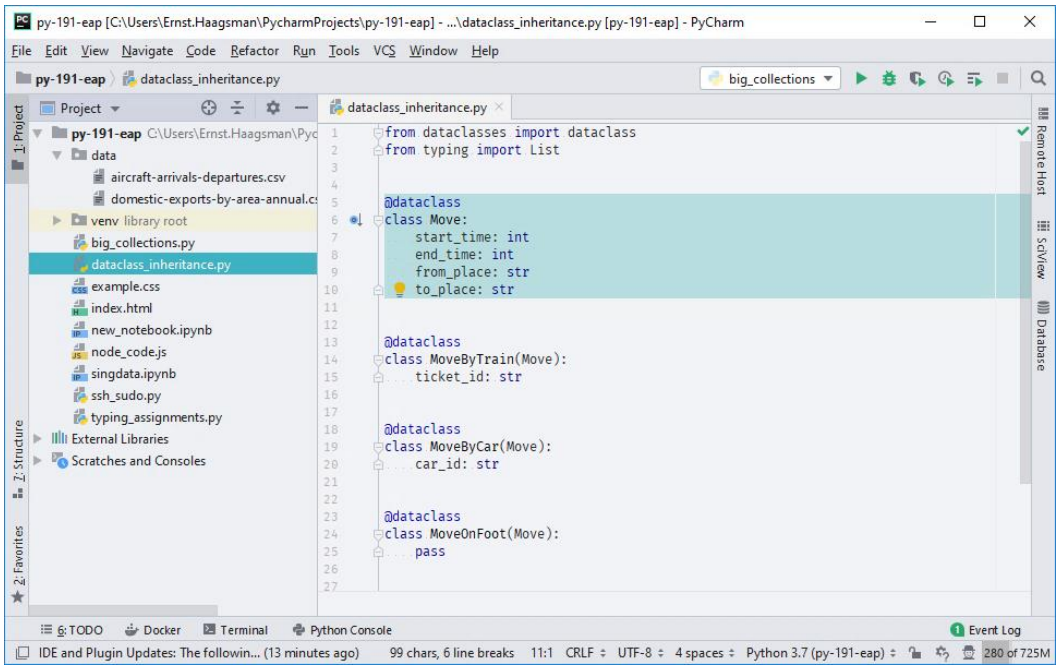


Рисунок 7 – Интерфейс программы PyCharm

Code::Blocks - простая, нетребовательная к ресурсам и очень производительная среда разработки с открытым исходным кодом. Поддерживает огромное количество компиляторов и отладчиков. Расширить функционал можно с помощью бесплатных плагинов.

Поддерживаемые языки: C, C++, Fortran.

Стоимость: бесплатная.

Преимущества:

– удобная структура меню;
– высокая производительность;
– встроенная система быстрой сборки.

Недостатки:

– несколько устаревший интерфейс.

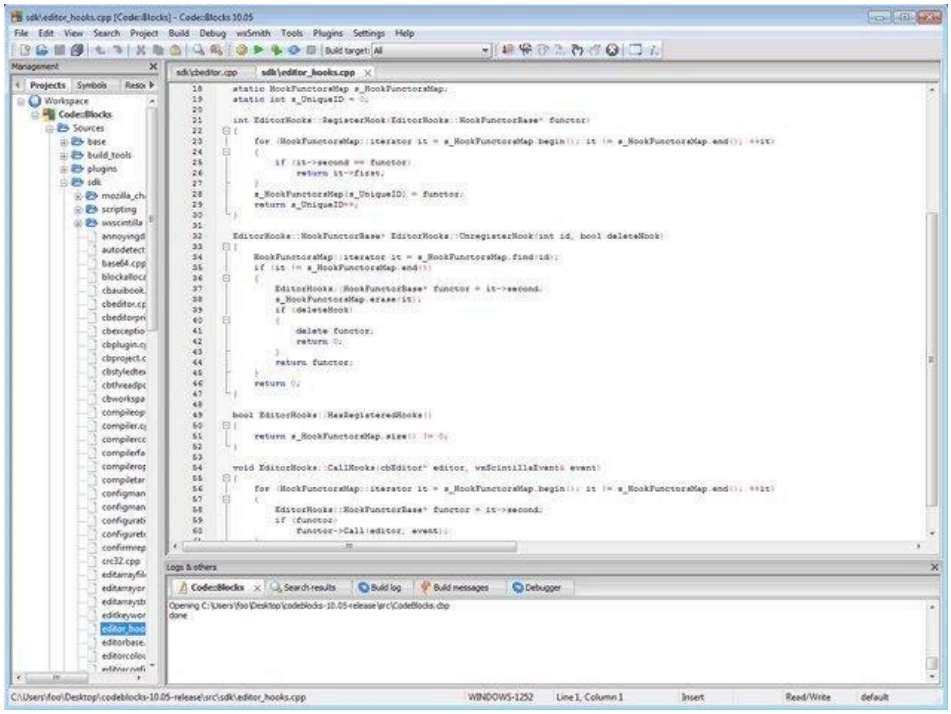


Рисунок 8 – Интерфейс программы Code::Blocks

NetBeans IDE – бесплатная IDE с открытым исходным кодом. По заявлениям многих считается одной из лучших IDE для разработки Java-приложений, в которую можно установить пакеты, обеспечивающие поддержку и других языков программирования.

Поддерживаемые языки: C, C++, C++ 11, Fortan, HTML 5, Java, PHP и другие.

Преимущества:

- интуитивно понятный интерфейс drag-and-drop;
- динамические и статические библиотеки;
- возможность удаленной разработки;
- совместима с windows, linux, macos и solaris;
- поддержка qt;
- поддерживает различные компиляторы, в том числе clang/llvm, cygwin, gnu, mingw и oracle solaris studio.

недостатки:

– netbeans требуется много памяти, поэтому на некоторых машинах эта среда может работать медленно.

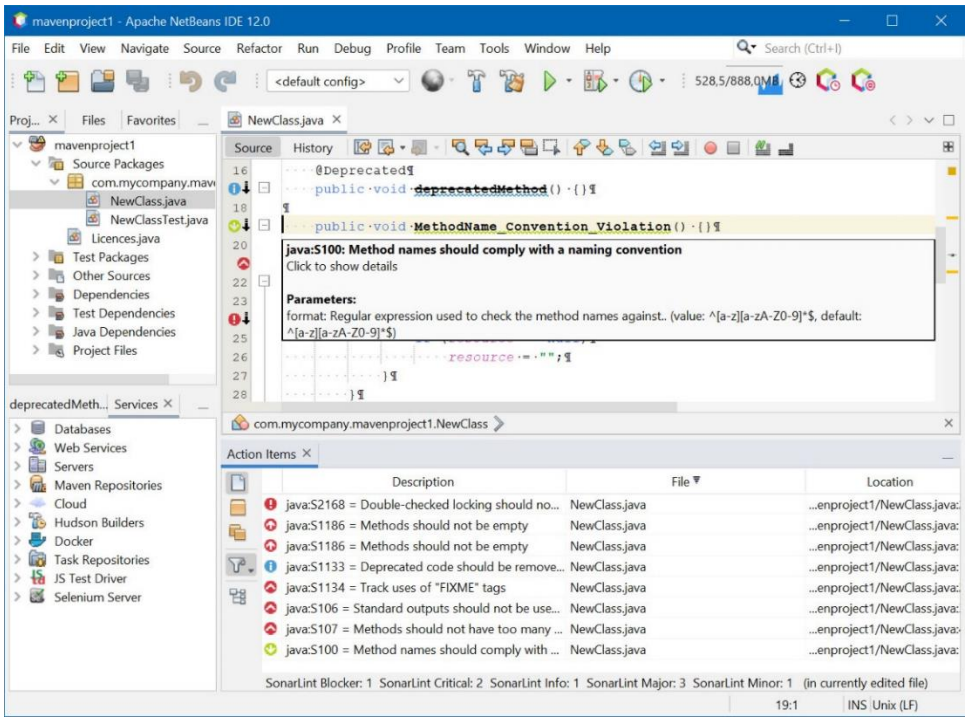


Рисунок 9 – Интерфейс программы NetBeans IDE

2.1.3. Анализ инструментов создания графического интерфейса

Windows Forms — интерфейс программирования приложений (API), который несет ответственность за графический пользовательский интерфейс и является частью Microsoft .NET. Этот интерфейс облегчает доступ к элементам интерфейса Microsoft Windows, создавая обертку для существующего Win32 API с управляемым кодом. Управляемый код — это классы, которые реализуют API для форм Windows, независимо от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, C++, так и на VB.Net, J# и др.

Windows Presentation Foundation (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презента-

ционная) подсистема в составе .NET Framework (начиная с версии 3.0), использующая язык XAML. WPF основана на векторной системе визуализации, которая независима от разрешающей способности выходного устройства и основана на возможностях современного графического оборудования. WPF предоставляет средства для создания визуального интерфейса, включая язык, элементы управления, привязку данных, макеты, двухмерную и трёхмерную графику, анимацию, стили, шаблоны, документы, текст, мультимедиа и оформление.

Swing — библиотека для создания графического интерфейса для программ на языке Java. Swing был разработан компанией Sun Microsystems. Он содержит ряд графических компонентов, таких как кнопки, поля ввода, таблицы и т. д.

Одной из его особенностью является принцип «Lightweight», означающий, что компоненты Swing отрисовываются самими компонентами на поверхности родительского окна, без использования компонентов операционной системы. Приложение Swing может иметь только одно окно, а все остальные компоненты нарисованы на ближайшем родителе с собственным окном.

Netbeans Platform — это фреймворк основанный на Swing, с помощью которого возможно создание больших десктоп приложений. Одноименное Netbeans IDE созданно, как раз, на базе Netbeans Platform. В нём содержится множество API для более легкой работы с окнами, действиями, файлами и т.п.

Каждый элемент приложения на Netbeans Platform может быть представлен отдельным модулем, который в свою очередь сопоставим с плагином. Модуль Netbeans является группой классов Java, которая предоставляет приложению определённый функционал.

Qt — фреймворк для разработки кроссплатформенного программного обеспечения на языке программирования C++.

Для многих языков программирования существуют библиотеки, позволяющие использовать преимущества Qt: Python — PyQt, PySide; Java — Qt Jambi; и др.

Qt позволяет запускать программное обеспечение, написанное с его помощью благодаря компиляции программы для каждой системы без изменения исходного кода. Он включает все базовые классы, которые могут потребоваться для разработки прикладных программ, от элементов графического интерфейса до классов баз данных, XML и работы с сетью. Является полностью объектно-ориентированным, расширяемым и поддерживающим технику компонентного программирования.

Комплектуется визуальной средой разработки графического интерфейса Qt Designer, позволяющей создавать диалоги и формы в режиме визуального редактирования.

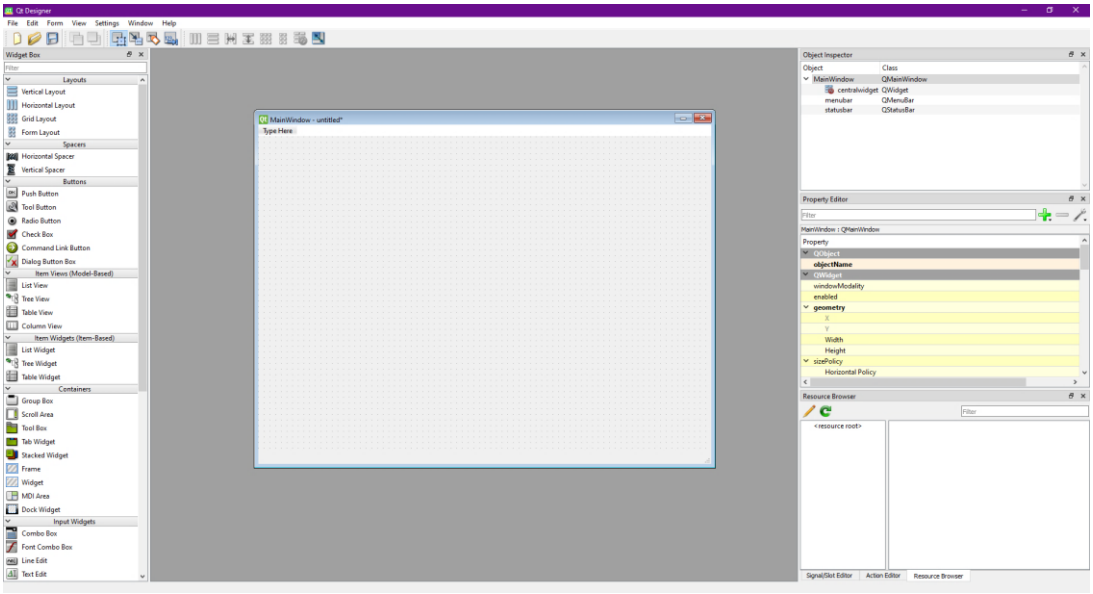


Рисунок 10 – Интерфейс программы Qt Designer.

2.1.4. Обоснование выбора программных средств

В качестве языка программирования был выбран язык Python. На это решение повлияли такие факторы как простота разработки, универсальность и поддержка библиотек для создания графического интерфейса.

В качестве среды программирования был выбран программный продукт JetBrains PyCharm, так как он очень удобен в использовании и обладает рядом специализированных функций для работы с языком Python.

В качестве инструмента для создания графического интерфейса был выбран фреймворк Qt6, из-за своей универсальности и хорошей документации.

Графический интерфейс программы разрабатывается при помощи программного обеспечения Qt Designer так как данная программа позволяет просто и быстро создавать пользовательский интерфейс, используя встроенные виджеты.

Взаимодействие программы и интерфейса происходит с помощью официальной библиотеки Qt6 для языка Python - PySide 6. Для данной библиотеки, как и библиотеки Qt 6, на официальном сайте расположена очень подробная документация, упрощающая процесс разработки.

2.2. Определение требований к программе.

2.2.1. Функциональные требования

Разрабатываемый программный продукт должен обладать следующими функциональными требованиями:

- Программа должна работать под операционной системой Windows;
- Программа должна подробно описывать работу всех алгоритмов шифрования, использующихся в процессе обучения дисциплине «Криптографические методы защиты информации»;
- Программа не должна требовать установки на жесткий диск компьютера;
- В программе должна быть реализована возможность изменения входных данных и повторный запуск алгоритма;
- Работа алгоритмов шифрования должна быть расписана по шагам;
- В описании работы алгоритма должны использоваться входные данные, вводимые пользователем с клавиатуры;
- В программе должна быть реализована возможность сохранения результатов в файл отчёта;
- При необходимости в ограничении алфавита вводимых данных

должна быть реализована возможность выбрать русский или английский язык ввода;

- Программа должна быть написана на русском языке.

2.2.2. Нефункциональные требования

- Должна быть обеспечена стабильная работа и быстродействие программы;
- Графический интерфейс должен быть простым и интуитивно понятным для пользователя;
- Графический интерфейс должен быть построен так, чтобы препятствовать ошибочным действиям пользователя;
- Программа и её графический интерфейс не должны чрезмерно использовать аппаратные ресурсы компьютера;
- Программа не должна занимать много места на жёстком диске компьютера;
- Результат вычислений и работы алгоритмов шифрования должен быть фактически верным.

2.3. Характеристика функциональных модулей

Программа состоит из нескольких модулей, объединённых через главное меню. Каждый модуль позволяет прочитать справочную информацию об алгоритме шифрования, зашифровать открытый текст и расшифровать криптограмму. Процесс шифрования и расшифровывания описывается по шагам, что позволяет в подробностях изучить работу алгоритма.

Существует также возможность сохранить результаты работы алгоритма в текстовый файл.

Каждый модуль программы можно разделить на 3 функциональных модуля:

- модуль пользовательского интерфейса;
- модуль обработки событий;

– модуль шифрования;

Взаимодействие между функциональными модулями представлено на рисунке 11.

Модуль пользовательского интерфейса отвечает за отрисовку графического интерфейса программы.

Модуль обработки событий отвечает на сигналы интерфейса о том, что была нажата кнопка, изменен текст в строке, изменена позиция в списке и др.

Модуль шифрования проводит алгоритм шифрования, используя введённые через графический интерфейс исходные данные и возвращает промежуточные и конечные результаты работы.

На рисунке 12 представлена структура реализованных в программе алгоритмов шифрования.

Взаимодействие пользователя с программой происходит по алгоритму, представленному на рисунке 13 в виде диаграммы последовательностей.

При запуске программы пользователь выбирает в меню модуль с нужным ему алгоритмом шифрования, читает информацию об алгоритме и вводит исходные данные. После нажатия на кнопку далее пользователю выводится результат работы алгоритма. Каждый шаг подробно расписывается. После просмотра результатов можно вернуться назад и ввести новые данные или закрыть окно и выбрать другой алгоритм.

Рисунок 11 – Взаимодействие функциональных модулей программы

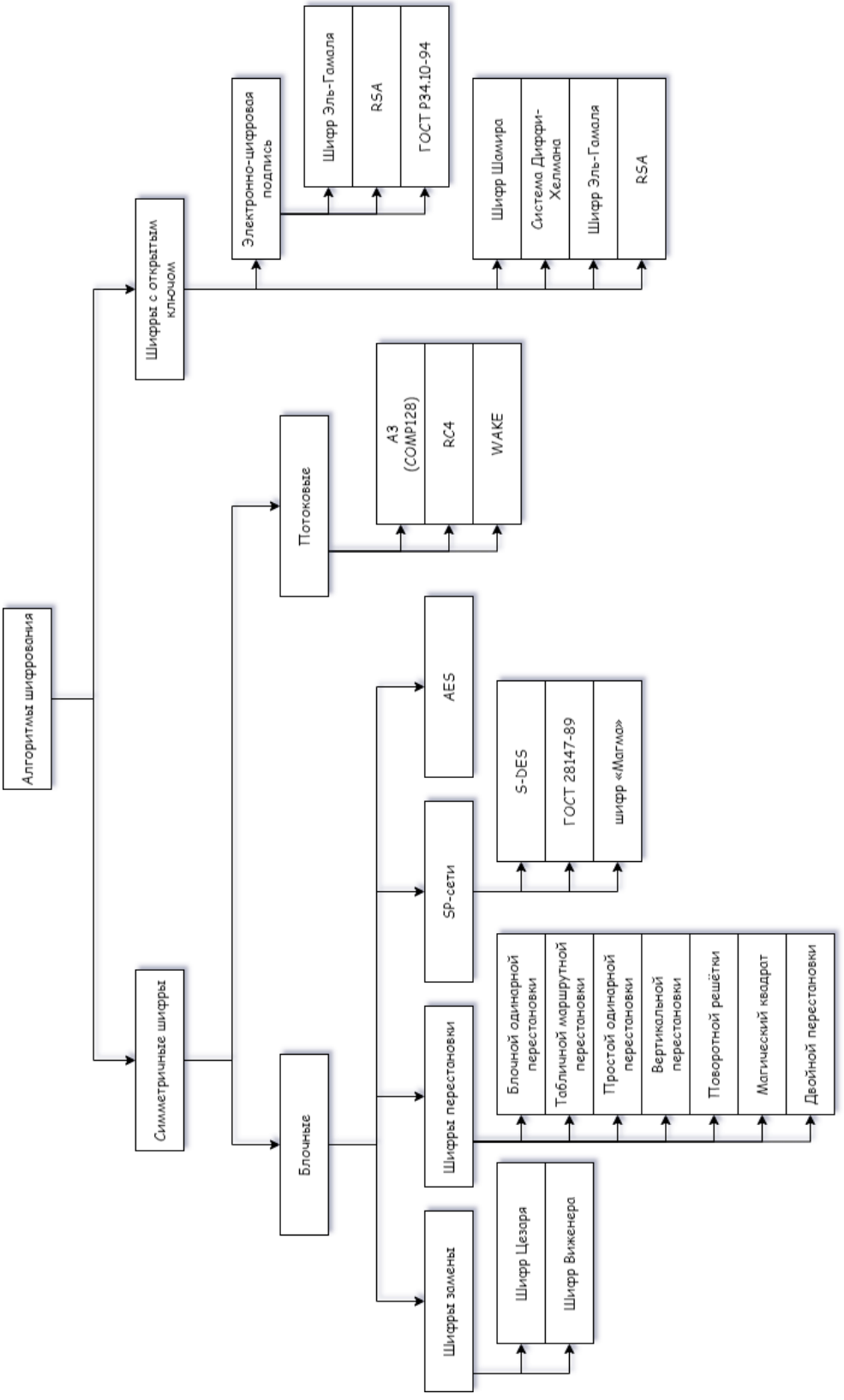


Рисунок 12 – Структура реализованных в программе шифров

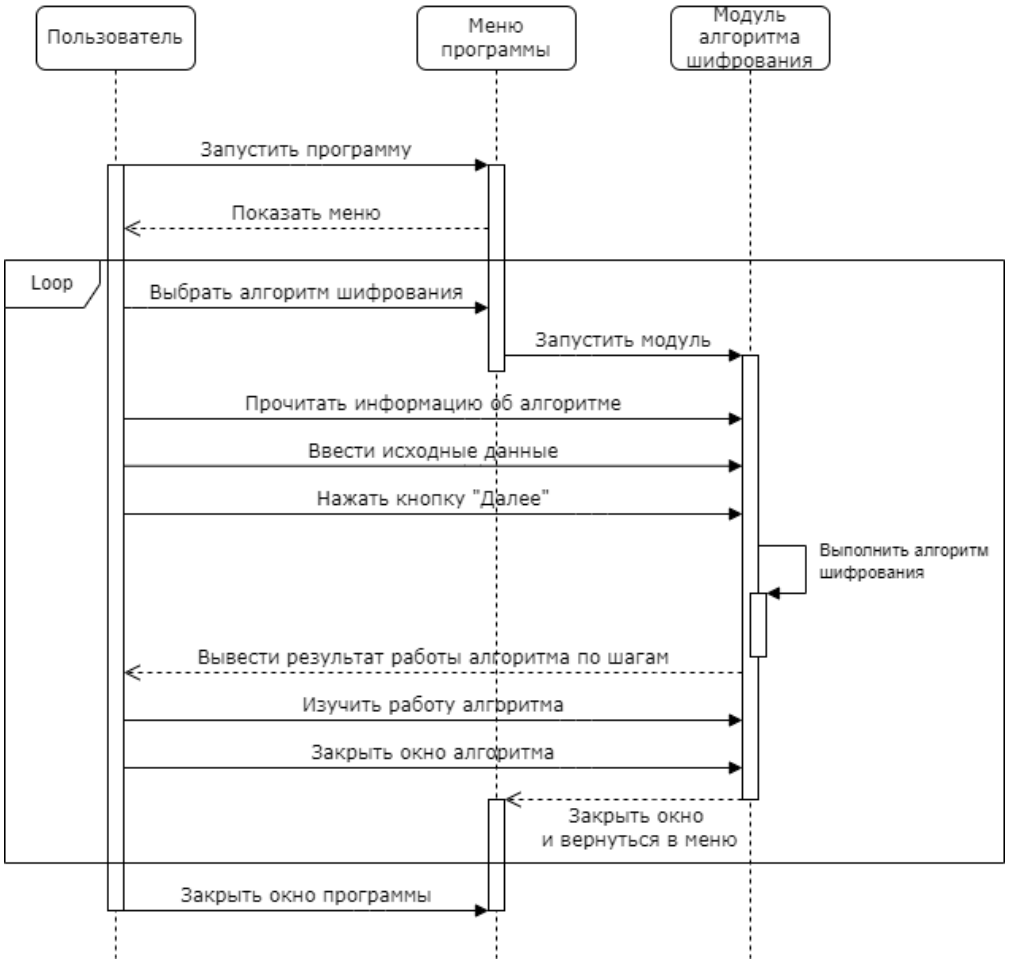


Рисунок 13 – Взаимодействие пользователя с программой.

2.4. Математическое обеспечение

Криптография является одной из наиболее важных областей применения математики. Стойкость большого числа алгоритмов шифрования обосновывается сложностью решения чисто математических задач: разложения больших чисел на множители, решения показательных сравнений в целых числах и других.

Методы и результаты различных разделов математики (в частности, алгебры, комбинаторики, теории чисел, теории алгоритмов, теории вероятностей и математической статистики) используются как при разработке шифров, так и при их исследованиях, в частности, при поиске методов вскрытия шифров. Шифр можно считать стойким, пока при его исследовании не выявляются особенности, которые потенциально можно использовать для вскрытия шифра. Для пользователей шифра очень важно узнать, что он ненадёжен, раньше, чем этим смогут

воспользоваться злоумышленники.

В частности, криптографические шифры с открытым ключом базируется на результатах классической теории чисел. Одной из важнейших операций в криптографии с открытыми ключами является операция возведения в степень по модулю.

Возведение в степень по модулю вычисляется по формуле:

$$c \equiv a^n \ (mod\ m), \quad (1)$$

где a – натуральное число;

n – степень;

m – натуральное число.

Например, криптостойкость реализованного в программе шифра RSA основывается на вычислительной сложности задачи факторизации больших чисел (разложения на простые множители). Эта система базируется на двух фактах из теории чисел:

1) задача проверки чисел на простоту является сравнительно легкой;

2) задача разложения чисел вида:

$$N = p*q, \quad (2)$$

где p и q — простые числа,

на множители (факторизация) является трудной вычислительной задачей, если известно только N, а p и q — большие числа.

В процессе генерации ключей вычисляется функция Эйлера φ от числа N:

$$\varphi(N) = (p-1)*(q-1), \quad (3)$$

где p и q — простые числа.

Далее находится число d по формуле:

$$d = e^{-1} mod\ \varphi(N), \quad (4)$$

где e – случайное число, взаимно простое со значением φ;

φ(N) – функция Эйлера от числа N.

Поскольку НОД(e, φ(N)) = 1, такое число d существует, причем в одном экземпляре.

Операция возведения в степень по модулю участвует как в процессе шифрования текста:

$$C = M^e mod\ N, \quad (5)$$

где M – передаваемое сообщение;

e – случайное число, взаимно простое со значением φ;

N – число, вычисляемое по формуле (2),

так и в процессе расшифровки:

$$M' = C^d mod\ N, \quad (6)$$

где C – зашифрованное сообщение;

d – число, вычисляемое по формуле (4);

N – число, вычисляемое по формуле (2).

3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ПРОДУКТА

При запуске программы пользователя встречает главное меню (рис. 14).

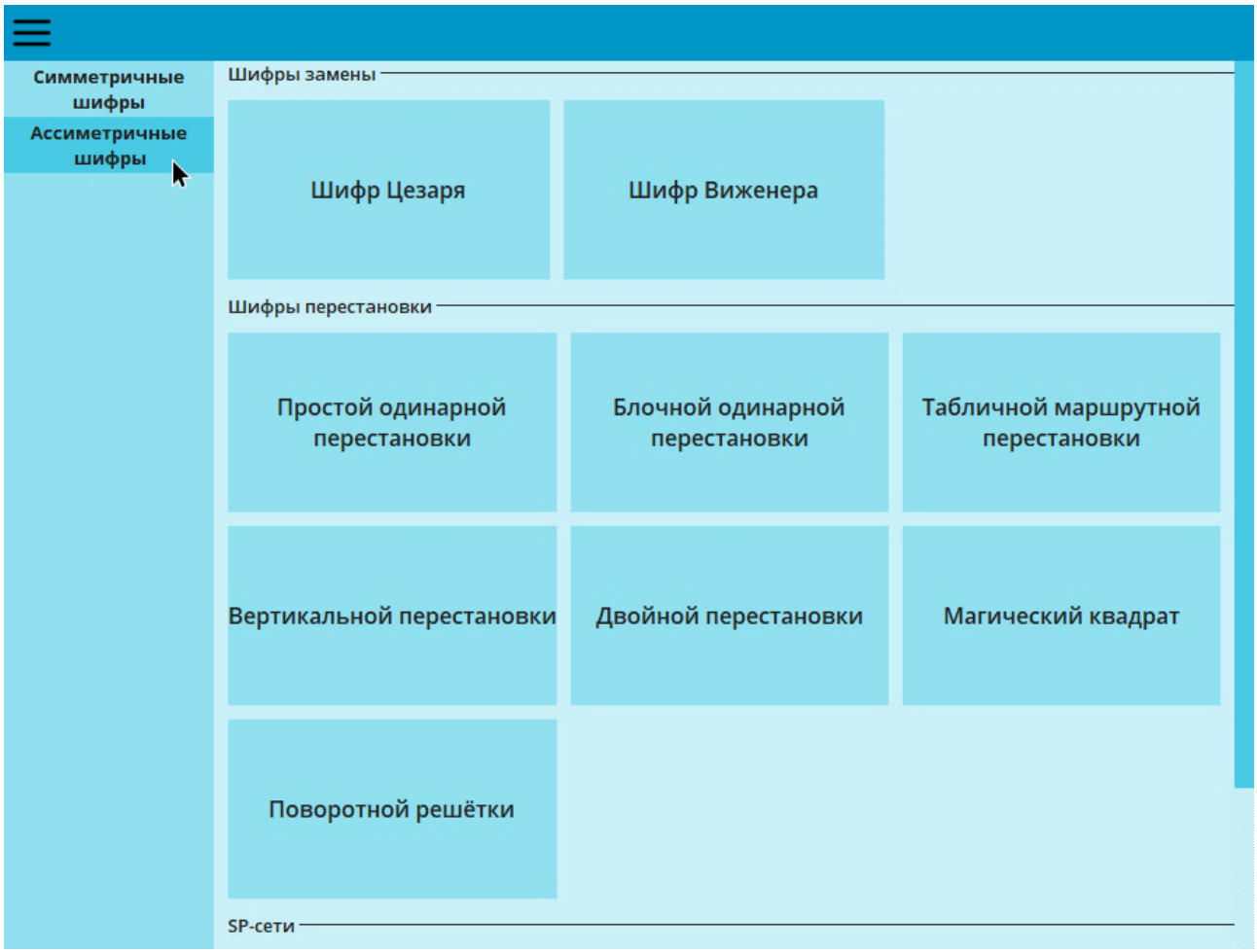


Рисунок 14 – Окно главного меню программы

Боковое меню позволяет переключаться между страницами с симметричными и асимметричными шифрами (рисунок 15 а). При необходимости данное меню можно свернуть, нажав на соответствующую кнопку (рисунок 15 б).

а) б)

Рисунок 15 – Боковое меню и кнопка сворачивания меню

Выбрав необходимый модуль в меню, пользователь видит окно алгоритма

шифрования (рисунок 16), где пользователь может ввести входные данные.

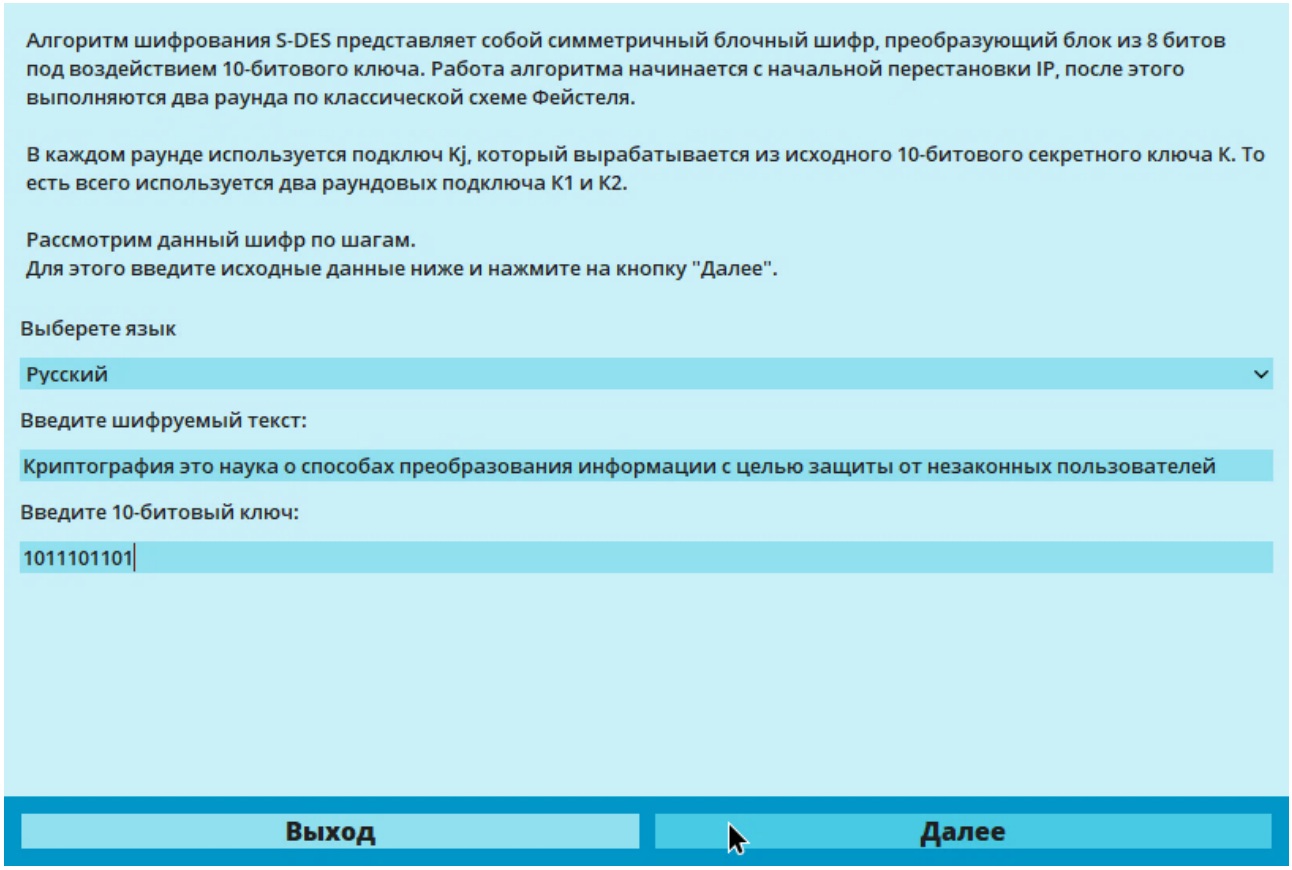


Рисунок 16 – Интерфейс алгоритма шифрования S-DES

В нижней части у каждого модуля шифрования располагается панель навигации с двумя кнопками (рисунок 17). Кнопка «Выход» закрывает окно алгоритма. Кнопка «Далее» запускает работу алгоритма и переносит пользователя на страницу с результатами. Кнопка «Далее» становится неактивной и цвет текста меняется на серый в случае, если введённых данных недостаточно, либо они могут привести к ошибке. На странице результатов кнопка «Далее» меняется на кнопку «Назад».

Рисунок 17 – Панель навигации, кнопка «Далее» неактивна.

На рисунке 18 представлена страница с результатами работы шифра. Каждый этап алгоритма расписан в подробностях.

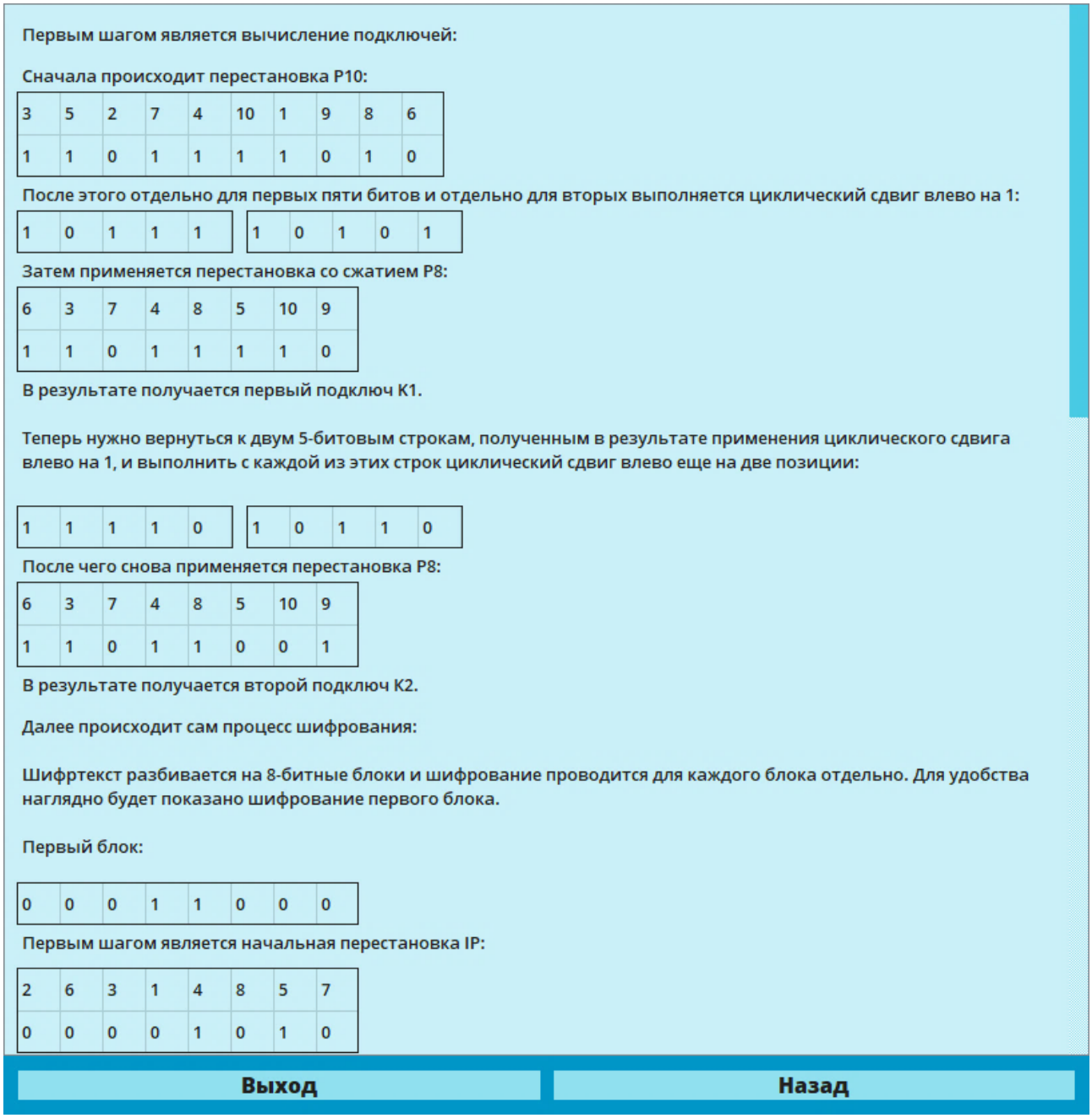


Рисунок 18 – Страница результатов

В самом низу страницы представлен конечный результат работы шифра, а также кнопка, для сохранения отчёта (рисунок 19).

После нажатия на кнопку отчёт сохраняется в виде текстового файла в папку, куда установлена программа (рисунки 20 и 21).

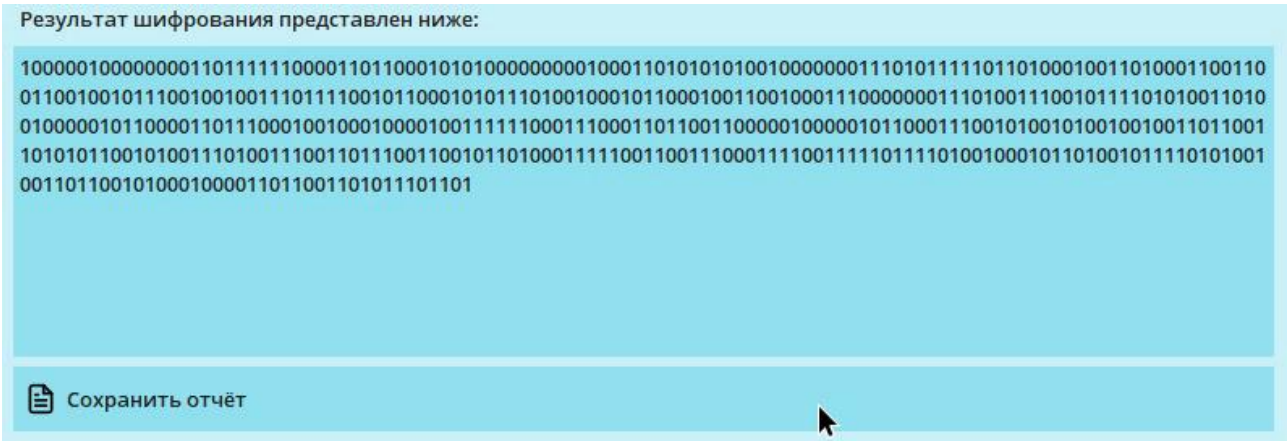


Рисунок 19 – Конечный результат и кнопка «Сохранить отчёт»

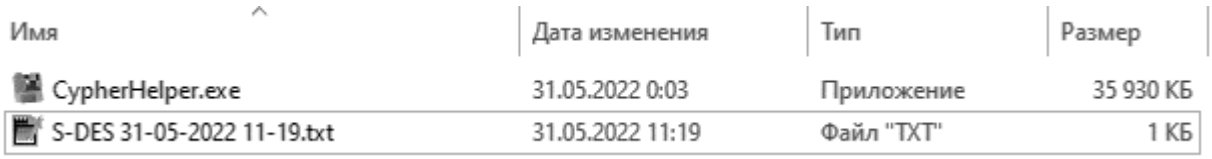


Рисунок 20 – Расположение файла с отчётом

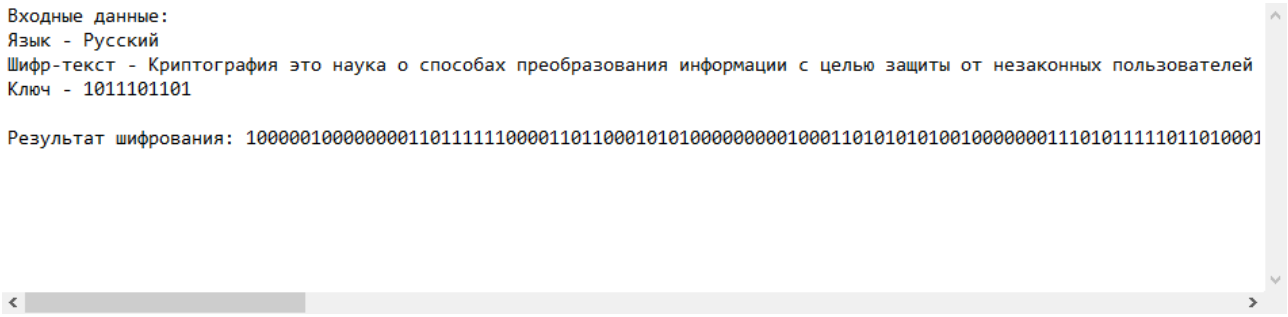


Рисунок 21 – Содержимое файла с отчётом

Подобным образом происходит взаимодействие с другими модулями программы.

4. БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1. Безопасность

4.1.1. Анализ графического интерфейса с точки зрения эргономичности

Для наиболее эффективного взаимодействия пользователя с программой необходимо соблюдать некоторые требования эргономичности, в частности при разработке графического пользовательского интерфейса программы.

Программа рассчитана на пользователей операционной системы Windows 10. Стиль и дизайн интерфейса приближен к приложениям, разработанных для данной операционной системы.

Цветовая палитра программы состоит из цветов преимущественно светло-голубого оттенка, которая не даёт напряжения на глаза пользователя. Интерактивные элементы имеют достаточный контраст с фоном, чтобы их различить.

Шрифт Open Sans был выбран в программе в качестве основного, так как он был спроектирован с прямым штрихом, открытыми формами и нейтральной, но дружественной внешностью, а также оптимизирован для удобочитаемости при печати, в веб- и мобильных интерфейсах. Размер шрифта варьируется от 10 до 12pt в зависимости от приоритета текста.

Для акцентирования внимания пользователя используется крупный текст и выделение дополнительным цветом. Например, при наведении курсора на нажимаемый объект или кнопку она меняет цвет (рисунок 22).

Рисунок 22 – Выделение кнопки цветом

Текстовые поля выделены дополнительным цветом, а также отображают

специальный курсор при наведении (рисунок 23).

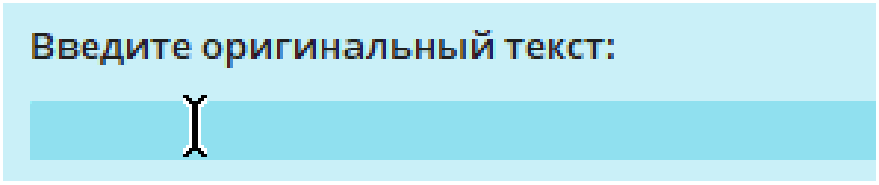


Рисунок 23 – Текстовое поле программы

Интерфейс каждого модуля программы согласован и состоит из панели навигации с двумя кнопками и основной панели с информацией.

Для того чтобы пользователю было проще понять почему кнопка «Далее», необходимая для получения результатов шифрование может быть неактивна при наведении на неё отображается подсказка (рисунок 24).

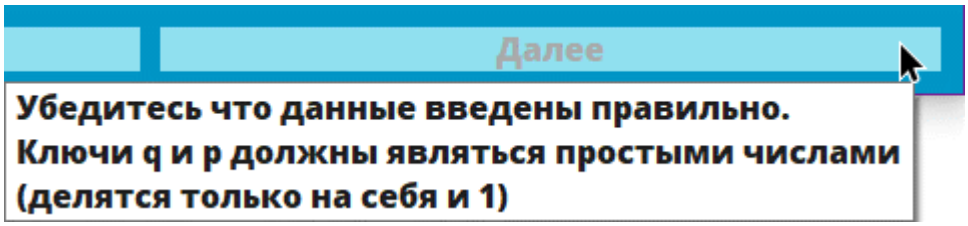


Рисунок 24 – Подсказка для пользователя

В целом, интерфейс программы соответствует рекомендациям к эргономичности взаимодействия с пользователем.

4.1.2. Эргономические требования к рабочему месту пользователя программного продукта.

Рабочий стол должен иметь высоту 720 мм, а его оптимальные размеры - 1600 х 900 мм. Под столешницей рабочего стола должно быть свободное пространство для ног. Размеры: как минимум 600 мм в высоту и 500 мм в ширину.

Рабочий стул: 400-500 мм в высоту, 400 мм в ширину, 380 мм в толщину. Поверхность сидения и спинки должна быть полумягкой, с нескользящим, неэлектризующим и воздухонепроницаемым покрытием.

На рабочем месте необходимо предусматривать подставку для ног. Ее размеры: длина - 400 мм, ширина - 350 мм, высота - 150 мм. Угол наклона подставки

- в пределах 0-20 градусов. Она должна иметь рифленое покрытие и бортик высотой 10 мм по нижнему краю.

4.1.3. Режимы труда и отдыха.

Для программного продукта применимы требования группы «В» (творческая работа в режиме диалога с ЭВМ). В зависимости от длительности работы подразделяются на три категории: первая группа (длительность работы до 2-х часов) при 8-ой рабочей смене суммарное время регламентированных перерывов не менее 30 минут и не менее 70 минут при 12-часовой; вторая группа (продолжительность работы до 4-х часов) при 8-ой рабочей смене время регламентированных перерывов не менее 50 минут и не менее 90 минут при 12-часовой; третья группа - длительность работы до 6-х часов, при 8-ой рабочей смене суммарное время регламентированных перерывов не менее 70 минут и не менее 120 минут при 12-часовой смене.

Продолжительность непрерывной работы с ВДТ без регламентированного перерыва не должна превышать 2 часов.

Профессиональные пользователи ВДТ и ПЭВМ должны проходить обязательные предварительные и периодические медицинские осмотры не менее 1 раза в год.

4.2. Экологичность

Активная разработка новых технологий неизбежно влечет за собой замену не только пришедшего в негодность, но и устаревшего офисного и компьютерного оборудования. Важной составляющей этого процесса является утилизация офисного и компьютерного оборудования. Эта процедура строго регулируется действующим законодательством Российской Федерации, в связи с чем невозможно просто выбросить старый компьютер, принтер или системный блок в мусорный контейнер.

Электроника, как и многие современные бытовые приборы, содержит ценные металлы, а также вредные компоненты (ртуть, свинец, мышьяк), отнесенные

к классу высокой опасности. Попав на свалку, такое оборудование причинит серьезный вред окружающей среде и здоровью людей.

В перечень законов, регламентирующих порядок утилизации офисной и домашней электронной техники, входят:

– ФЗ № 89 «Об отходах производства и потребления» от 24.06.1998;

– ФЗ № 41 «О драгоценных металлах и драгоценных камнях» от 26.03.1998.

Важно знать, что законодательство полностью применимо как к юридическим, так и к физическим лицам. В соответствии с постановлением правительства Российской Федерации № 524 от 26 августа 2006 года деятельность, связанная с утилизацией офисного оборудования и электроники, требует наличия у организации лицензии на работу с опасными отходами различных классов.

Законы регулируют необходимость утилизации компьютерной и оргтехники. Несоблюдение этих законов может повлечь за собой строгие санкции в отношении предприятия по инициативе надзорных органов. Если руководство предприятия или ответственные за этот процесс лица не будут соблюдать правила утилизации или выполнять их должным образом, предприятие будет оштрафовано на достаточно серьезную сумму. Это касается и несанкционированного вывоза техники, которая была отработана или вышла из строя.

На рынке имеется значительное число компаний, оказывающих услуги по утилизации отходов. Многие из них действуют в строгом соответствии с законом, предоставляя качественные услуги, а некоторые прибегают к фиктивным методам утилизации отходов. Обычно под эту категорию попадают легальные фирмы, имеющие разрешение только на вывоз ТБО, желающие получить дополнительную прибыль. Обратившись в такую организацию, ответственность за несоблюдение КОаП 19.14 лежит на заказчике.

Надежного партнера в оказании утилизационных услуг можно распознать

по наличию у него соответствующих документов, а именно свидетельства о постановке на учет в приборной палате и лицензии на осуществление операций по сбору, применению, обезвреживанию, перевозке и распределению опасных отходов, полученную от Ростехнадзора.

По требованию любая фирма обязуется предъявлять карту о постановке на учет, где перечисляются разрешенные процедуры с драгметаллами, а к лицензии должен прилагаться список категорий отходов, разрешенных к утилизации данной компанией.

4.3. Чрезвычайные ситуации

Наиболее распространенными источниками возникновения чрезвычайных ситуаций техногенного характера являются пожары и взрывы, которые происходят:

- на промышленных объектах;
- на объектах добычи, хранения и переработки легковоспламеняющихся, горючих и взрывчатых веществ;
- на транспорте;
- в шахтах, горных выработках, метрополитенах;
- в зданиях и сооружениях жилого, социально-бытового и культурного назначения.

ПОЖАР – это вышедший из-под контроля процесс горения, уничтожающий материальные ценности и создающий угрозу жизни и здоровью людей.

Основными причинами пожара являются: неисправность электросетей, несоблюдение технических и противопожарных мер (курение, открытый огонь, использование неисправного оборудования и т.д.).

Основными опасными факторами возгорания являются термическое излучение, высокая температура, токсичное воздействие дыма (продукты горения:

окись углерода и т.д.) и снижение видимости при задымлении. В случае долговременного воздействия этих факторов опасности, критическими значениями для человека являются:

температура – 70° С;

плотность теплового излучения – 1,26 кВт/м2;

концентрация окиси углерода – 0,1% объема;

видимость в зоне задымления – 6-12 м.

ВЗРЫВ – это горение, сопровождающееся освобождением большого количества энергии в ограниченном объеме за короткий промежуток времени.

Результатом взрыва является формирование и распространение со сверхзвуковой скоростью ударной волны (при избыточном давлении более 5 кПа), оказывающей механическое воздействие на окружающие объекты.

Основными последствиями взрыва являются ударная волна и осколочные поля, вызванные обломками от различных объектов, технического оборудования и взрывных устройств.

В число предупредительных мероприятий могут быть включены мероприятия, направленные на устранение причин, которые могут вызвать пожар (взрыв), на ограничение (локализацию) распространения пожаров, создание условий для эвакуации людей и имущества при пожаре, своевременное обнаружение пожара и оповещение о нем, тушение пожара, поддержание сил ликвидации пожаров в постоянной готовности.

Соблюдение технологических режимов производства и содержание оборудования, особенно энергосетей, в хорошем рабочем состоянии позволяют в большинстве случаев исключить причину пожара.

Своевременное обнаружение пожара может быть обеспечено за счет оснащения помещений системами автоматической пожарной сигнализации или, в некоторых случаях, за счет организационных мер.

Первоначальное пожаротушение успешно осуществляется в помещениях,

оборудованных автоматическими установками пожаротушения.

В случае обнаружения пожара необходимо незамедлительно реагировать на него, используя все имеющиеся методы тушения пожара (песок, вода, огнетушители и т.д.). Если потушить огонь в кратчайшие сроки невозможно, нужно вызвать пожарную охрану предприятия или города (по телефону 01).

При эвакуации горящие помещения и задымленные места проходятся быстро, задержав дыхание, защитив нос и рот влажной плотной тканью. В сильно задымленном помещении нужно передвигаться ползком или пригнувшись – в прилегающем к полу пространстве чистый воздух сохраняется дольше.

Если на человеке загорелась одежда, нужно сбросить ее либо набросить на горящего любое покрывало и плотно прижать. Если доступ воздуха ограничен, горение быстро прекратится. Нельзя давать человеку с горящей одеждой бежать.

Не стоит подходить к взрывоопасным предметам и трогать их. При угрозе взрыва лучше лечь на живот, защищая голову руками, дальше от окон, застекленных дверей, проходов, лестниц. Если произошел взрыв, необходимо принять меры к недопущению пожара и паники, оказать первую медицинскую помощь пострадавшим.

ЗАКЛЮЧЕНИЕ

В результате работы было разработано информационно-обучающее программное обеспечение по дисциплине «Криптографические методы защиты информации», а также были выполнены основные задачи:

- исследована предметная область;
- определены основные требования к программному продукту;
- произведён выбор программного обеспечения и инструментов для разработки;
- разработан дизайн графического интерфейса;
- написана программа.

Данная программа, несомненно, должна помочь в преподавании дисциплины как для студентов направления среднего профессионального образования, так и для студентов бакалавриата.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1 Алистер, Коберн Современные методы описания функциональных требований к системам: моногр. / Коберн Алистер. - М.: ЛОРИ, 2019. - 610 с.

2 Алферов Основы криптографии: Учебное пособие / Алферов, А.П. и. - М.: Гелиос АРВ; Издание 2-е, испр. и доп., 2015. - 480 с.

3 Бабаш, А. В. История криптографии. Часть I / А.В. Бабаш, Г.П. Шанкин. - М.: Гелиос АРВ, 2020. - 240 с.

4 Бахаров, Л. Е. Информационная безопасность и защита информации : разделы криптография и стеганография : практикум / Л. Е. Бахаров. - Москва : Изд. Дом НИТУ «МИСиС», 2019. - 59 с.

5 Информационный мир XXI века. Криптография — основа информационной безопасности : методическое руководство / под ред. Э. А. Болелова ; Московский государственный технический университет гражданской авиации. - 4-е изд. — Москва : Издательско-торговая корпорация «Дашков и К°», 2020. — 126 с.

6 Булгаков А.Б., Безопасность жизнедеятельности: учебное пособие /А.Б. Булгаков. – Благовещенск: Изд-во АмГУ, 2013. – 627 с.

7 Бушманов А.В., Галаган Т.А., Самохвалова С.Г. Методические указания к выполнению и защите выпускной квалификационной работы бакалавра / [ред. С.Г. Самохвалова]. - Благовещенск: АмГУ, 2017.-50 с.

8 Васильева, И. Н. Криптографические методы защиты информации : учебник и практикум для вузов / И. Н. Васильева. — Москва : Издательство Юрайт, 2022. — 349 с.

9 Васильев, А. Н. Python на примерах. Практический курс по программированию / А.Н. Васильев. - М.: Наука и техника, 2016. - 432 с.

10 Герман, О. Н. Теоретико-числовые методы в криптографии / О.Н. Герман, Ю.В. Нестеренко. - М.: Академия, 2020. - 272 с.

11 Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2022. — 235 с.

12 ГОСТ Р ИСО 14915-1-2016. Эргономика мультимедийных пользовательских интерфейсов. Часть 1. Принципы проектирования и структура. – М. : Стандартиформ, 2019. с изм. и доп. – 14 с.

13 Грекул, В. И. Проектирование информационных систем: учебное пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. – 3-е изд. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 299 с.

14 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 343 с.

15 Дронов, Владимир Python 3 и PyQt 5. Разработка приложений / Владимир Дронов. - М.: БХВ-Петербург, 2016. - 991 с.

16 Златопольский, Д.М. Основы программирования на языке Python / Д.М. Златопольский. - Москва : ДМК Пресс, 2017. - 284 с.

17 Иванов, М. А. Криптографические методы защиты информации в компьютерных системах и сетях : учебное пособие / М. А. Иванов, И. В. Чугунков. — Москва : НИЯУ МИФИ, 2012. — 400 с.

18 Каширская, Е. Н. Криптографические системы : учебное пособие / Е. Н. Каширская, А. П. Кушнир. — Москва : РТУ МИРЭА, 2021. — 66 с.

19 Лось, А. Б. Криптографические методы защиты информации для изучающих компьютерную безопасность : учебник для вузов / А. Б. Лось, А. Ю. Нестеренко, М. И. Рожков. — 2-е изд., испр. — Москва : Издательство Юрайт, 2022. — 473 с.

20 Полупанов, Д. В. Программирование в Python 3 : учебное пособие / Д. В. Полупанов, С. Р. Абдюшева, А. М. Ефимов. — Уфа : БашГУ, 2020. — 164 с.

21 Прохоренок, Н. Python 3 и PyQt 5. Разработка приложений / Прохоренок Н., Дронов В. – СПб: БХВ-Петербург, 2016. – 832 с.

22 Саммерфилд, Марк Программирование на Python 3. Подробное руководство / Марк Саммерфилд. - М.: Символ-плюс, 2017. - 386 с.

23 Саммерфильд, Марк Python на практике / Марк Саммерфильд. - М.: ДМК Пресс, 2020. - 338 с.

24 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2022. — 210 с.

25 Фомичев, В. М. Криптография — наука о тайнописи : учебное пособие / В. М. Фомичев. - Москва : Прометей, 2020. - 66 с.

26 Фомичёв, В. М. Криптографические методы защиты информации в 2 ч. Часть 1. Математические аспекты : учебник для вузов / В. М. Фомичёв, Д. А. Мельников ; под редакцией В. М. Фомичёва. — Москва : Издательство Юрайт, 2022. — 209 с.

27 Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с.

28 Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-наДону ; Таганрог : Издательство Южного федерального университета, 2017. - 107 с.

29 doc.qt.io [Электронный ресурс] : офиц. Сайт – Режим доступа: https://doc.qt.io/qtforpython/index.html. – 25.05.2022

ПРИЛОЖЕНИЕ А

Техническое задание

1 ВВЕДЕНИЕ

1.1 Наименование программы

Полное наименование разрабатываемой системы: Информационно-обучающая программа «Cypher Helper»

1.2 Наименование предприятия разработчика и заказчика системы:

Разработчик: студент факультета математики и информатики ФГБОУ ВО «АмГУ» Капитонов Сергей Олегович, группа 855-об, отделение очного обучения.

Заказчик: Амурский государственных университет.

Форма собственности: государственная.

Адрес: 675000, Россия, Амурская область, г. Благовещенск, ул. Игнатьевское шоссе, д.21.

1.3 Перечень правил документов, на основании которых создания создается система

– ГОСТ 34.602-89 – техническое задание параметры на проектирование автоматизированной также системы управления;

Система создается на основании технического задания (ТЗ). ТЗ на ИС является основным документом, определяющим требования и порядок создания автоматизированной размещаться системы качестве, в соответствии с которым проводится разработка ИС и ее приемка при вводе в действие.

– требование к системе;

– первичные надежность документы прохождения;

– отчет по практической практике.

1.4 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ: 20.05.2021

Продолжение ПРИЛОЖЕНИЯ А

Плановый срок окончания работ: 20.06.2021

2 НАЗНАЧЕНИЕ И ЦЕЛИ СОЗДАНИЯ СИСТЕМЫ

2.1 Назначение системы

Разрабатываемая система предназначена для помощи в преподавании дисциплины «Криптографические методы защиты информации».

2.1.1 Функциональное и эксплуатационное назначение

- реализовать работу криптографических шифров;
- разработать графический интерфейс;
- реализовать взаимодействие между интерфейсом и алгоритмами шифрования.

В данный момент существует очень малое количество подобного программного обеспечения. Разрабатываемый программный продукт позволит сэкономить деньги и время на разработку подобной программы с нуля.

2.2 Цель создания системы

Целью создания системы является модернизация учебного процесса и упрощение преподавания дисциплины с помощью интерактивной информационно-обучающей программы.

3 ХАРАКТЕРИСТИКА ОБЪЕКТА АВТОМАТИЗАЦИИ

3.1 Краткие сведения об объекте автоматизации

Объектом автоматизации является образовательный процесс ФГБОУ ВО «Амурский государственный университет».

Улучшению принадлежит дисциплина «Криптографические методы защиты информации»

4 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ПРОДУКТУ

4.1 Требования к приложению

4.1.1 Требования к структуре и функционированию

Продолжение ПРИЛОЖЕНИЯ А

Разрабатываемый программный продукт должен реализовывать следующие функции:

- Программа должна подробно описывать работу всех алгоритмов шифрования, использующихся в процессе обучения дисциплине «Криптографические методы защиты информации»
- В программе должна быть реализована возможность изменения входных данных и повторный запуск алгоритма;
- Работа алгоритмов шифрования должна быть расписана по шагам;
- В описании работы алгоритма должны использоваться входные данные, вводимые пользователем с клавиатуры;
- В программе должна быть реализована возможность сохранения результатов в файл отчёта;
- При необходимости в ограничении алфавита вводимых данных должна быть реализована возможность выбрать русский или английский язык ввода;

4.1.2 Требования к квалификации и численности персонала, режиму его работы

Преподавателю, ведущему курс, в котором будет использоваться программа, необходимо ознакомиться с руководством пользователя, чтобы консультировать учеников во время работы с программой.

4.1.3 Требования к интерфейсу пользователя

К графическому интерфейсу предъявляются следующие требования:

- Графический интерфейс должен быть простым и интуитивно понятным для пользователя;
- Графический интерфейс должен быть построен так, чтобы препятствовать ошибочным действиям пользователя;
- Цветовая палитра интерфейса должна быть мягкой и чрезмерно не

напрягающей глаза.

4.1.5 Перспективы развития, модернизация системы

К возможным перспективам развития можно отнести:

- улучшение взаимодействия пользователя и программы, увеличение участия пользователя в работе программы;
- добавление новых шифров;
- добавление анимации;
- улучшение системы сохранения отчётов.

4.1.6 Требования к защите информации от несанкционированного доступа

В приложении необходимо реализовать защиту от несанкционированного доступа, с помощью методов шифрования исходного кода.

4.2 Требования к видам обеспечения

4.2.1 Требования к лингвистическому обеспечению

Для разработки и поддержки данного программного обеспечения необходимы знания языка Python, библиотеки PySide6, фреймворка Qt6 и умение пользоваться программой Qt designer.

4.2.2 Требования к программному обеспечению

Для разработки программного продукта необходимо иметь следующее программное обеспечение:

- 64-разрядная операционная система Windows 10;
- Программа редактирования пользовательского интерфейса Qt Designer;
- Язык программирования Python;
- Библиотека PySide6, установленная в Python.

Для эксплуатации программы необходимо иметь устройство с операционной системой Windows 10.

4.2.3 Требования к техническому обеспечению

Для разработки и эксплуатации необходимо иметь персональный компьютер или ноутбук, к которым предъявляются следующие минимальные требования:

- процессор с архитектурой x64
- 10 Гб свободного места на жестком диске или SSD;
- 4 Гб ОЗУ.

4.2.4 Требования к условиям эксплуатации, и характеристика окружающей среды

Помещения, в которых предполагается использование программы, должны соответствовать согласованным показателям температуры, влажности и освещённости.

Условия эксплуатации должны соответствовать нормальным климатическим условиям, определённым в ГОСТ 27201-87 и иметь следующие значения:

- искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой равномерного освещения;
- температура воздуха в помещении от 15 °С до 25 °С;
- относительная влажность воздуха в помещении от 40% до 60% при температуре 25 °С;
- атмосферное давление от 630 мм. Рт. Ст. до 800 мм Рт. Ст.

4.2.5 Требования к организационному обеспечению

Для корректной эксплуатации системы следует разработать руководство пользователей и провести инструктаж.

5 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМЫ

5.1 Перечень стадий и этапов работ по созданию системы

Этапы создания приложения, которые необходимо выполнить:

1 этап – разработка технического задания, определение требований к при-

ложению, стадий, этапов и сроков разработки программы, согласование и утверждение технического задания;

2 этап – анализ процессов деятельности организации.

3 этап – анализ предметной области и средств разработки.

4 этап – разработка программного продукта.

5 этап – тестирование программного продукта.

6 этап – доработка программного продукта;

7 этап – согласование созданного приложения с требованиями заказчика;

8 этап – внедрение и сопровождение.

5.2 Состав организации исполнителя работ

Все работы выполняются студентом факультета математики и информатики Амурского государственного университета Капитоновым Сергеем Олеговичем.

6 ТРЕБОВАНИЯ К ПРИЕМКЕ-СДАЧЕ ПРОЕКТА

6.1 Виды, состав, объем и методы испытаний программы

Должны быть проведены следующие виды испытаний:

- предварительные испытания;
- опытная эксплуатация;
- приёмочные испытания.

На этапе предварительных испытаний проводится тестирование программы, проверка её работоспособности при запуске алгоритмов шифрования.

На этапе опытной эксплуатации проверяется работоспособность приложения на реальных занятиях, на лабораторных работах и практиках. В ходе этого этапа устраняются выявленные недостатки системы.

Приемочные испытания проводят для определения соответствия системы

техническому заданию, оценки качества опытной эксплуатации и решения вопроса о возможности приемки системы в постоянную эксплуатацию.

6.2 Общие требования к приемке работ по стадиям

Предварительные испытания и эксплуатация проводятся на аппаратных средствах Исполнителя.

По результатам испытаний возможны доработки и исправления.

Выявленные в ПО и документации недостатки Исполнитель исправляет в специально оговоренные после проведения испытаний сроки.

Сдача-приёмка работ производится поэтапно, в соответствии с календарным планом.

Все создаваемые в рамках настоящей работы программные изделия передаются Заказчику, как в виде готового файла для установки приложения, так и в виде исходных кодов, представляемых в электронной форме на стандартном машинном носителе.

7 ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОРГАНИЗАЦИИ ЗАКАЗЧИКА К ВВОДУ ПРИЛОЖЕНИЯ В ДЕЙСТВИЕ

Перед вводом в эксплуатацию готового продукта разработчик должен договорится с руководителем организации о временном промежутке, в течение которого он обязан внедрить разработанный программный продукт.

Под внедрением понимается комплекс мероприятий, включающий обучение пользователей, установку программы для дальнейшего использования, предоставление необходимой документации к программе.

7.1 Организационные мероприятия

Под организационными мероприятиями понимаются ознакомление пользователей с «Руководством пользователя», а также предоставление инструкций по установке программы на компьютер.

Продолжение ПРИЛОЖЕНИЯ А

8 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

При вводе программы в эксплуатацию пакет сопроводительных документов должен включать:

- техническое задание;
- описание программного продукта;
- руководство пользователя.