Geekbrains

**Разработка проекта «Шифратор» на Java.**

Разработчик:

Веб-разработка на Java

Денисенко С.А.

Краснодар

2024

**«Разработка проекта «Шифратор» на Java»**

**Содержание**

|  |  |
| --- | --- |
| Введение……………………………………………………………………. | 1 |
| Глава 1. Предпроектная стадия…………………………………………… | 2 |
| 1.1. Порядок проведения работ…………………………………………… | 2 |
| 1.2. Определение требований к приложению…………………………… | 3 |
| 1.3. Выбор стека технологий……………………………………………… | 4 |
| Глава 2. Проектная стадия………………………………………………… | 5 |
| 2.1. Проектирование общей архитектуры приложения………………… | 5 |
| Заключение………………………………………………………………… | 11 |
| Список использованной литературы…………………………………….. | 12 |

**Введение**

Графическое приложение "Шифратор" представляет собой приложение, в котором предлагается зашифровать или расшифровать сообщение.

В приложении «Шифратор» реализовано несколько действий:

1. Ввод сообщения

2. Ввод зашифрованного сообщения

3. Ввод ключа шифрования/расшифрования

4. Выбор режима работы

5. Вывод результата работы приложения

«Шифратор» предлагает простой и интуитивно понятный интерфейс для шифрования и расшифровки сообщений.

Основная функциональность приложения включает в себя:

**Возможность выбора режима работы**: Пользователь может выбрать режим работы одним кликом мыши в специализированном поле.

**Ввод зашифрованных сообщений**: в приложении реализована возможность ввода зашифрованных сообщений и их расшифровка используя ключ шифрования.

**Ввод сообщений**: в приложении реализована возможность ввода сообщений и их шифрование используя ключ шифрования.

**Удобный вывод результата работы**: в приложении реализовано отдельное окно вывода результата работы с отображением исходного текста и зашифрованного сообщения.

**Глава 1. Предпроектная стадия**

**1.1. Порядок проведения работ**

В ходе создания проекта были проведены следующие виды работ:

1. **Планирование проекта**:
   * Определение функциональных и нефункциональных требований к приложению.
   * Проектирование архитектуры приложения и выбор используемых технологий.
2. **Настройка окружения разработки**:
   * Установка и настройка Java,
3. **Создание проекта**:
   * Создан новый проект с использованием библиотек Swing и AWT.
4. **Разработка модели данных, сервисов.**
5. **Разработка пользовательского интерфейса**.
6. **Отладка и оптимизация**:
   * Отладка кода для исправления ошибок и улучшения производительности.
   * Оптимизация кода и конфигурации приложения для улучшения скорости работы и эффективности использования ресурсов.
7. **Документирование**:
   * Создание документации по проекту, включая описание архитектуры, инструкции по установке и запуску, и другие справочные материалы.

**1.2. Определение требований к приложению**

Для разработки приложения "Шифратор" были определены следующие требования:

1. Функциональные требования:
   * Пользователи должны иметь право выбора действий простым и удобным способом.
   * Приложение должно удобно и понятно отображать результат своей работы.
   * Приложение должно уметь как шифровать сообщение по заданному ключу, так и производить расшифровку сообщений использую ключ шифрования.
2. Нефункциональные требования:
   * Удобство использования: Интерфейс должен быть интуитивно понятным и простым для пользователя.
   * Производительность: Приложение должно обеспечивать быстрый доступ к данным и отзывчивый интерфейс.
3. Требования к интерфейсу пользователя:
   * Интуитивно понятный интерфейс, который позволяет пользователям легко делать выбор действий.
4. Требования к производительности:
   * Быстрый доступ к данным и отзывчивый интерфейс приложения.

**1.3. Выбор стека технологий**

1. **Язык программирования**: Java для разработки кроссплатформенных приложений так как этот язык хорошо подходит для создания универсальных и надежных программ работающих на любой платформе.
2. **Среда разработки**: IntelliJ IDEA для разработки приложения. Этот стек технологий обеспечивает разработку приложения "Шифратор" с учетом требований к функциональности, производительности и безопасности.

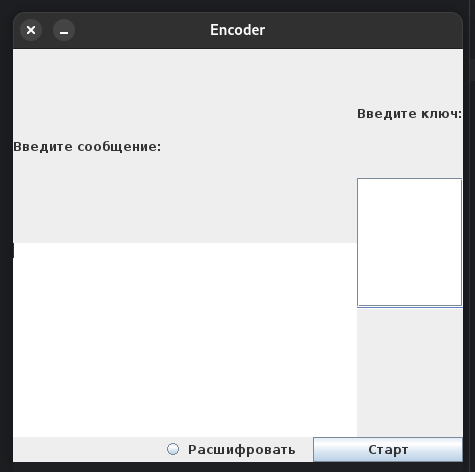
Архитектура приложения в рамках библиотек Swing и AWT основывается на объектно ориентированном подходе, который позволяет разделить приложение на основные компоненты (объекты) и отделить графичесские компоненты от основной логики приложения. Наше приложение состоит из данных составных частей:

1. **Главное окно (MainWindow)**:
   * Класс отвечает за отображение данных пользователю.
2. **Шифратор (Encoder)**:
   * Класс отвечает за алгоритм шифрования данных пользователя, включает два метода: 'encrypt'', ''decrypt''.

Каждый из этих компонентов играет важную роль в построении архитектуры приложения на Java, обеспечивая его модульность, гибкость и масштабируемость. Они позволяют разделить различные аспекты приложения и обеспечить их независимую разработку и тестирование.

|  |
| --- |
| **Глава 2. Проектная стадия** |
| **2.1. Проектирование общей архитектуры приложения** |

Первое, с чего начиналась работа — это разработка графической части приложения.

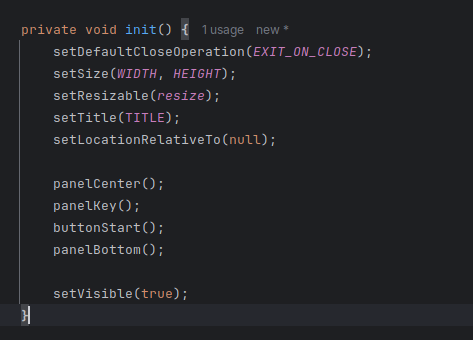


Класс MainWindow является компонентом приложения, отвечающим за взаимодействие с пользователем и обработку событий, связанных с управлением перечня действий и шифратором. Класс MainWindow предоставляет различные методы для обработки собтий ввода пользователей и взаимодействия с соответствующими сервисами.

Метод main используется для запуска приложения

Методы класса позволяют пользователю выбирать режим работы программы, отрисовать графическую часть программы и обрабатывать события пользователя:

Метод init() производит инициализацию графических состовляющих окна путем запуска соответствующих им методов и определения параметров окна (ширина, высота, заголовок).



Методы panelCenter и panelKey отрисовывают поле ввода сообщения и поле ввода ключа шифрования.



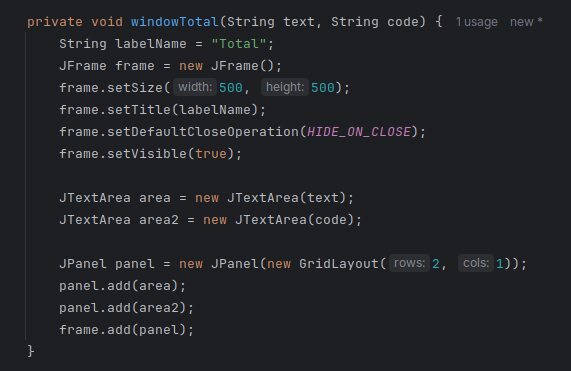
Метод buttonStart инициализирует клавишу начала процесса шифрования/расшифрования, определяет обработчик события и взаимодействует с шифратором.



Метод panelButton отрисовывает выбор редима работы шифратора.

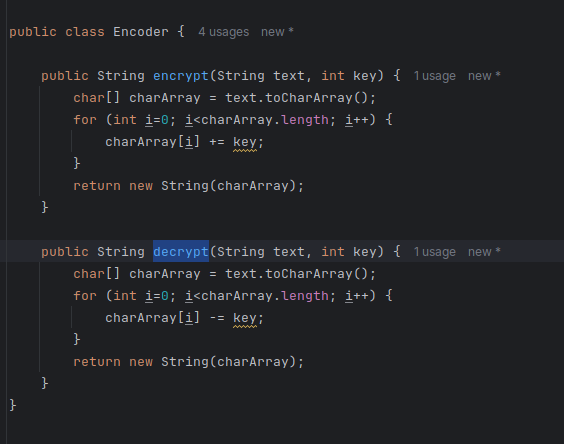


Метод windowsTotal отрисовывает итоговое окно.



Начало формы

Класс Encoder определяет основную логику приложения. Содержит методы encrypt и decrypt позволяющие шифровать и расшифровывать сообщения использую ключ шифрования



**Заключение**

В рамках данного проекта было разработано приложение «Щифратор», предназначенное для шифрования сообщения пользователя используя ключ шифрования. Проект был реализован с использованием современных технологий разработки, таких как Java и др.

В процессе разработки были достигнуты следующие цели:

* Создание удобного и интуитивно понятного интерфейса для пользователей приложения.
* Реализация функционала в выборе режима работы одним кликом мыши.

Дальнейшее развитие приложения может включать в себя добавление новых методов шифрования, мультиязычность, запись и чтение из файла, синхронизация с сервером и веб-версия.

В целом проект "Шифратор" позволил не только показать навыки разработки графических приложений с использованием современных технологий, но и продемонстрировать понимание принципов удобства использования для конечного пользователя.

**Список литературы**

1. Материал Гикбрейнс.

2. Статьи и видео-материалы из сети интернет.