

magmacrypt

1. Теоретическая часть

2. Практическая часть

2.1. Общие сведения

Данная реализация на языке Java использует версию языка 11 и JDK версии 11. Для сборки приложения используется фреймворк Maven версии 3.6.3; для написания unit-тестов - JUnit версии 4.11.

Сборка и выполнение по умолчанию осуществляются на виртуальной машине с запущенным дистрибутивом Fedora Linux Server 35 (версия ядра - 5.15.10-200.fc35.x86_64).

Вывод команды `java --version` для среды сборки и выполнения по умолчанию:

```
openjdk 11.0.13 2021-10-19
OpenJDK Runtime Environment 18.9 (build 11.0.13+8)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.13+8, mixed mode, sharing)
```

В качестве основных параметров сборки определены следующие свойства:

1. `groupId = ru.mirea.edu.magmacrypt` - для управления компиляцией исходников, также является основным пакетом приложения и используется в качестве соответствующего идентификатора;
2. `artefactId = magmacrypt` и `version = 1.0` - для генерации jar-архива приложения;
3. `mainClass = ru.mirea.edu.magmacrypt.App` - для определения расположения метода `main` (точки входа в приложение);

Структура файлов исходного кода выглядит следующим образом (вывод команды `tree`):

1. Каталог `src` - содержит файлы исходного кода с расширением `.java`:
 1. Каталог `main` - приложение, далее пакет `ru.mirea.edu.magmacrypt (groupId)`:
 1. Класс `App` - основной класс приложения;
 2. Пакет `auxiliary` - вспомогательные классы для работы с данными;
 3. Пакет `cipher` - классы, относящиеся к реализации алгоритма шифрования.
 2. Каталог `test` - unit-тесты, далее пакет `ru.mirea.edu.magmacrypt (groupId)`, содержащий класс `AppTest` для проверки работы алгоритма шифрования и взаимодействия с различными типами входных данных;
2. Каталог `target` (создается и наполняется при компиляции исходного кода - файлы `.class`, и сборке `jar`-файла):
 1. Каталог `classes`, далее пакет `ru.mirea.edu.magmacrypt (groupId)`
 2. Каталог `test-classes`, далее пакет `ru.mirea.edu.magmacrypt (groupId)`
 3. Файл `magmacrypt-1.0.jar` - собранное приложение;

3. Файл `pom.xml` - конфигурация сборки;
4. Файл `rebuild.sh` - скрипт очистки собранного решения и сборки нового;
5. Каталог `TEST_DATA` - набор данных, используемых для тестирования.

2.2. Описание реализации

2.3. Пример запуска