

**МУНИЦИПАЛЬНОЕ ОБРАЗОВАНИЕ**

**ГОРОДСКОЙ ОКРУГ «ГОРОД НИЖНИЙ НОВГОРОД»**

**МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«ШКОЛА № 74 С УГЛУБЛЕННЫМ ИЗУЧЕНИЕМ ОТДЕЛЬНЫХ ПРЕДМЕТОВ»**

Научное общество учащихся «Эврика»

**Разработка приложения «Cryptographer — Шифровальщик» для смартфона под управлением ОС Android**

Выполнил: Малышев Тимур

обучающийся 11а класса

Научный руководитель:

Родина Е.А.,

учитель информатики

Н.Новгород

2021

Оглавление

[Введение 3](#_Toc68637412)

[1. Основы криптографической теории 5](#_Toc68637413)

[1.1. Что такое криптография 5](#_Toc68637414)

[1.2. Стойкая криптография 6](#_Toc68637415)

[1.3. Как действует криптография 6](#_Toc68637416)

[1.4. Традиционная криптография 6](#_Toc68637417)

[1.5. Шифр Цезаря 7](#_Toc68637418)

[1.6. Симметричное шифрование и управление ключами 8](#_Toc68637419)

[1.7. Криптография с открытым ключом 8](#_Toc68637420)

[2. Криптоалгоритмы 10](#_Toc68637421)

[2.1. Алгоритм DES 10](#_Toc68637422)

[2.2. Алгоритм AES 11](#_Toc68637423)

[2.3. Алгоритм BlowFish 12](#_Toc68637424)

[2.4. Алгоритм RSA 13](#_Toc68637425)

[3. Разработка приложения «Cryptographer — Шифровальщик» 16](#_Toc68637426)

[3.1. Алгоритм «eXT»: 17](#_Toc68637427)

[3.2. Графический пользовательский интерфейс (GUI) 19](#_Toc68637428)

[3.3. Экран настроек 20](#_Toc68637429)

[3.4. Элемент Toast 20](#_Toc68637430)

[3.5. Выход из приложения 21](#_Toc68637431)

[3.6. Смена цвета акцента 21](#_Toc68637432)

[3.7. Обмен ключами шифрования 21](#_Toc68637433)

[3.8. Оптимизация приложения 22](#_Toc68637434)

[Заключение 25](#_Toc68637435)

[Список литературы 26](#_Toc68637436)

[Приложения 28](#_Toc68637437)

# Введение

Сегодня все больше людей общаются не лично, а через компьютерную сеть. Честность и безопасность многих протоколов человеческого общения основаны на личном присутствии. И если человек при личной встрече может определить уровень доверия, то компьютеру нужны определенные правила.

Даже в древние времена сохранности информации придавалось большое значение. В наши дни ситуация не изменилась: сегодня ничто не ценится так дорого, как информация, записанная в цифровой форме и передаваемая по сетям. И ничто другое невозможно так легко потерять, как эту информацию.

Для защиты информации применяется ряд мер. К ним относятся:

* организация средств защиты;
* аппаратные средства защиты;
* криптографические средства защиты.

Для обеспечения сохранности информации существуют следующие возможности:

* создание надежного, недоступного для других людей канала связи между абонентами;
* использование общедоступного канала связи но скрыть сам факт передачи информации
* использование общедоступного канала связи, но передавать информацию в таком преобразованном виде, чтобы восстановить ее мог только адресат.

Изучением вопросов шифрования данных занимается наука — криптология, включающая криптографию и криптоанализ.

Криптография изучает методы и алгоритмы шифрования данных (правила построения и использования шифров), направленные на то, чтобы сделать эти данные бесполезными для противника. Методы криптографии также используются для подтверждения подлинности источника данных и контроля целостности данных.

Криптоанализ — это наука о раскрытии исходного текста зашифрованного сообщения без доступа к ключу. Процесс шифрования информации с точки зрения программиста можно представить следующим образом: Открытый текст -> алгоритм шифрования (ключ 1) -> шифрованное сообщение.

А процесс расшифровывания: Шифрованное сообщение -> алгоритм дешифрирования (ключ 2) -> открытый текст.

Целью данной работы является разработка приложения с графическим интерфейсом для смартфона под ОС Android для анонимного общения в сети.

Для достижения поставленной цели предполагается решение ряда задач:

1. Изучение алгоритмов шифрования.
2. Анализ существующих приложений.
3. Проектирование структуры приложения.
4. Программная реализация.
5. Тестирование полученного продукта и устранение недостатков.
6. Организация поддержки программного продукта.

В настоящее время существует большое количество криптоалгоритмов. В ходе работы над проектом автором были изучены следующие алгоритмы: DES, AES, BlowFish, RSA. Так же изучены шифры подстановки. В процессе работы автор разработал свой алгоритм шифрования, основанный на шифре Цезаря, который получил название eXT.

# 1. Основы криптографической теории

Информация, которая может быть прочитана, осмыслена и понята без каких-либо специальных мер, называется открытым текстом (plaintext, clear text). Метод искажения открытого текста таким образом, чтобы скрыть его суть, называется зашифрованием – (encryption или enciphering). Зашифрование открытого текста приводит к его превращению в непонятную абракадабру, именуемую шифртекстом (ciphertext). Шифрование позволяет скрыть информацию от тех, для кого она не предназначается, несмотря на то, что они могут видеть сам шифртекст. Противоположный процесс по обращению шифртекста в его исходный вид называется расшифрованием (decryption или deciphering).



Рис. 1 Цикл шифрования/дешифрования данных

## 1.1. Что такое криптография

Криптография – это наука об использовании математики для зашифрования и расшифрования данных. Криптография позволяет хранить важную информацию или передавать её по ненадёжным каналам связи (таким как Интернет) так, что она не может быть прочитана никем, кроме легитимного получателя.

В то время как криптография – это наука о защите данных, криптоанализ – это наука об анализировании и взломе зашифрованной связи. Классический криптоанализ представляет собой смесь аналитики, математических и статистических расчётов, а также спокойствия, решительности и удачи. Криптоаналитиков также называют взломщиками. Криптология объединяет криптографию и криптоанализ.

## 1.2. Стойкая криптография

Криптография может быть стойкой, а может быть слабой. Криптографическая стойкость измеряется тем, сколько понадобится времени и ресурсов, чтобы из шифртекста восстановить исходный открытый текст. Результатом стойкой криптографии является шифртекст, который исключительно сложно взломать без обладания определёнными инструментами по дешифрованию.

## 1.3. Как действует криптография

Криптографический алгоритм, или шифр, – это математическая формула, описывающая процессы зашифрования и расшифрования. Чтобы зашифровать открытый текст, криптоалгоритм работает в сочетании с ключом – словом, числом или фразой. Одно и то же сообщение одним алгоритмом, но разными ключами будет преобразовываться в разный шифртекст. Защищённость шифртекста целиком зависит от двух вещей: стойкости криптоалгоритма и секретности ключа.

Криптоалгоритм плюс всевозможные ключи и протоколы, приводящие их в действие, составляют криптосистему .

## 1.4. Традиционная криптография

В традиционной криптографии, также называемой шифрованием тайным, или симметричным, ключом, один и тот же ключ используется как для зашифрования, так и для расшифрования данных. Data Encryption Standart (DES) – пример симметричного алгоритма, широко применявшегося на Западе с 70х годов в банковской и коммерческой сферах. В настоящее время его сменяет Advanced Encryption Standard (AES).

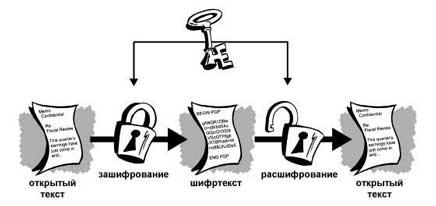


Рис. 2 Шифрование с симметричным ключом

## 1.5. Шифр Цезаря

Крайне простой пример симметричного шифрования – это подстановочный шифр. Подстановочный шифр заменяет каждую часть информации другой информацией. Чаще всего это достигается смещением букв алфавита. Алгоритм состоит в том, чтобы сдвинуть алфавит, а ключ – это число букв, на которое произведено смещение.

Допустим, если мы решим зашифровать слово «SECRET», используя ключ Цезаря, равный 3, то сдвинем латинский алфавит так, чтобы он начинался с третьей буквы (D).

Итак, беря исходный вариант,

и смещая всё на 3, получаем,

где D=A, E=B, F=Cи т. д.

Используя эту схему, открытый текст «SECRET» превращается в «VHFUHW». Чтобы ктото мог восстановить исходный текст, вы сообщаете ему, что ключ – 3.

Данный алгоритм иллюстрирует, как действует симметричное шифрование.

## 1.6. Симметричное шифрование и управление ключами

Симметричное шифрование имеет ряд преимуществ. Первое – скорость криптографических операций. Оно особенно полезно для шифрования данных, которые остаются у вас. Однако, симметричное шифрование, применяемое само по себе как средство защиты пересылаемых ценных данных, может оказаться весьма затратным просто из-за сложности передачи тайного ключа.

Для установления шифрованной связи с помощью симметричного алгоритма, отправителю и получателю нужно предварительно согласовать ключ и держать его в тайне. Если они находятся в географически удалённых местах, то должны прибегнуть к помощи доверенного посредника, например, надёжного курьера, чтобы избежать компрометации ключа в ходе транспортировки. Злоумышленник, перехвативший ключ в пути, сможет позднее читать, изменять и подделывать любую информацию, зашифрованную или заверенную этим ключом. Глобальная проблема симметричных шифров состоит в сложности управления ключами.

## 1.7. Криптография с открытым ключом

Проблема управления ключами была решена криптографией с открытым , или асимметричным, ключом, концепция которой была предложена Уитфилдом Диффи и Мартином Хеллманом в 1975 году.

Криптография с открытым ключом – это асимметричная схема, в которой применяются пары ключей: открытый (public key), который зашифровывает данные, и соответствующий ему закрытый (private key), который их расшифровывает. Вы распространяете свой открытый ключ по всему свету, в то время как закрытый держите в тайне. Любой человек с копией вашего открытого ключа может зашифровать информацию, которую только вы сможете прочитать.

Хотя ключевая пара математически связана, вычисление закрытого ключа из открытого в практическом плане невыполнимо. Каждый, у кого есть ваш открытый ключ, сможет зашифровать данные, но не сможет их расшифровать. Только человек, обладающий соответствующим закрытым ключом может расшифровать информацию.



Рис.3 Шифрование с открытым ключом

Главное достижение асимметричного шифрования в том, что оно позволяет людям, не имеющим существующей договорённости о безопасности, обмениваться секретными сообщениями. Необходимость отправителю и получателю согласовывать тайный ключ по специальному защищённому каналу полностью отпала. Все коммуникации затрагивают только открытые ключи, тогда как закрытые хранятся в безопасности. Примерами криптосистем с открытым ключом являются Elgamal (названная в честь автора, Тахира Эльгамаля), RSA (названная в честь изобретателей: Рона Ривеста, Ади Шамира и Леонарда Адлмана), DiffieHellman (названная, правильно, в честь её создателей) и DSA, Digital Signature Algorithm (изобретённый Дэвидом Кравицом).

# 2. Криптоалгоритмы

Работа основана на использовании следующих алгоритмов шифрования: DES, AES, BlowFish. В процессе работы автор разработал свой алгоритм, основанный на алгоритме Цезаря, который получил название eXT.

## 2.1. Алгоритм DES

DES (Data Encryption Standard) — симметричный алгоритм шифрования, в котором один ключ используется как для шифрования, так и для расшифрования данных. DES разработан фирмой IBM и утвержден правительством США в 1977 году как официальный стандарт (FIPS 46-3). Алгоритм DES широко использовался при хранении и передаче данных между различными вычислительными системами; в почтовых системах, в электронных системах чертежей и при электронном обмене коммерческой информацией. Стандарт DES реализовывался как программно, так и аппаратно. Предприятиями разных стран был налажен массовый выпуск цифровых устройств, использующих DES для шифрования данных. Все устройства проходили обязательную сертификацию на соответствие стандарту.

Несмотря на то, что уже некоторое время эта система не имеет статуса государственного стандарта, она по-прежнему широко применяется и заслуживает внимания при изучении блочных шифров с закрытым ключом. Длина ключа в алгоритме DES составляет 56 бит.

Систему DES вполне можно использовать в небольших и средних приложениях для шифрования данных, имеющих небольшую ценность.

Общая схема DES



## 2.2. Алгоритм AES

В 2001 году после специально объявленного конкурса в США был принят новый стандарт на блочный шифр, названный AES (Advanced Encryption Standard), в основу которого был положен шифр Rijndael, разработанный бельгийскими специалистами. Шифр Rijndael/AES характеризуется размером блока 128 бит, длиной ключа 128, 192 или 256 бит и количеством раундов 10, 12 или 14 в зависимости от длины ключа.

В отличие от шифров, предлагаемых DES основу Rijndael составляют так называемые линейно-подстановочные преобразования. Блок данных, обрабатываемый с использованием Rijndael, делится на массивы байтов, и каждая операция шифрования является байт-ориентированной. Каждый раунд состоит из трех различных обратимых преобразований, называемых слоями. Эти слои следующие.

Нелинейный слой. На этом слое выполняется замена байтов. Слой реализован с помощью S-блоков, имеющих оптимальную нелинейность, и предотвращает возможность использования дифференциального, линейного и других современных методов криптоанализа.

Линейный перемешивающий слой гарантирует высокую степень взаимопроникновения символов блока для маскировки статистических связей. На этом слое в прямоугольном массиве байтов выполняется сдвиг строк массива и перестановка столбцов.

Слой сложения по модулю 2 с подключом выполняет непосредственно шифрование.

Шифр начинается и заканчивается сложением с ключом. Это позволяет закрыть вход первого раунда при атаке по известному тексту и сделать криптографически значимым результат последнего раунда.

В алгоритме широко используются табличные вычисления, причем все необходимые таблицы задаются константно, т.е. не зависят ни от ключа, ни от данных. Функции шифрования и расшифрования различны.

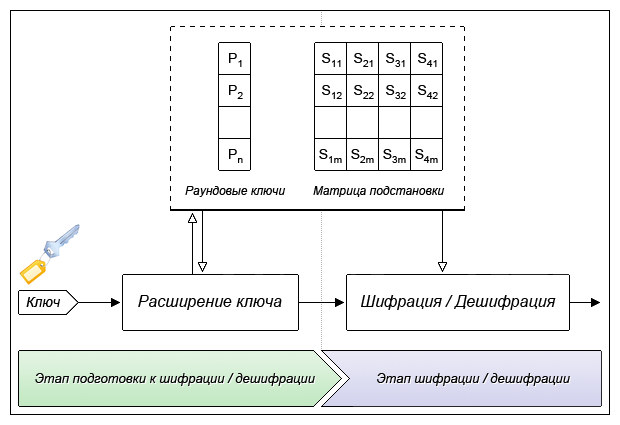
Алгоритм Rijndael хорошо выполняется как в программной, так и в аппаратной реализации. Rijndael имеет небольшие требования к памяти, что делает его пригодным для систем с ограниченными ресурсами. Надежность шифрования алгоритмом Rijndael очень высоко оценивается специалистами.

## 2.3. Алгоритм BlowFish

В 1994 году, на семинаре Fast Software Encryption в Кембридже, а впоследствии и в журнале «Lecture Notes in Computer Science» (#809, 1994), Брюс Шнайер презентовал свой алгоритм блочного шифра, который был назван Blowfish.

Отличительными особенностями этого алгоритма стала более высокая степень криптостойкости, нежели алгоритма DES (в том числе за счет использования переменной длины ключа, до 448 бит), высокая скорость шифрации/дешифрации (за счет генерации таблиц замены) и конечно — возможность его свободного применение для любых целей.

В общем случае алгоритм состоит из двух этапов — расширение ключа и шифрация/дешифрация исходных данных.



На этапе расширения ключа, исходный ключ преобразуется в матрицу раундовых ключей (P) и матрицу подстановки (S, Substitution-box) (или замены), общим объемом в 4168 байт.

## 2.4. Алгоритм RSA

Алгоритм шифрования с открытым ключом RSA был предложен одним из первых в конце 70-х годов ХХ века. Его название составлено из первых букв фамилий авторов: Р.Райвеста (R.Rivest), А.Шамира (A.Shamir) и Л.Адлемана (L.Adleman). Алгоритм RSA является, наверно, наиболее популярным и широко применяемым асимметричным алгоритмом в криптографических системах.

Алгоритм основан на использовании того факта, что задача разложения большого числа на простые сомножители является трудной. Криптографическая система RSA базируется на следующих двух фактах из теории чисел:

задача проверки числа на простоту является сравнительно легкой;

задача разложения чисел вида n = pq ( р и q — простые числа); на множители является очень трудной, если мы знаем только n, а р и q — большие.

Алгоритм RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и n -1 для некоторого n.

Рассмотрим сам алгоритм. Пусть абонент А хочет передать зашифрованное сообщение абоненту Б. В этом случае абонент Б должен подготовить пару (открытый ключ; закрытый ключ) и отправить свой открытый ключ пользователю А.

Первым этапом является генерация открытого и закрытого ключей. Для этого вначале выбираются два больших простых числа Р и Q. Затем вычисляется произведение N:

N = PQ.

После этого определяется вспомогательное число f:

f = (Р - l)(Q - 1).

Затем случайным образом выбирается число d < f и взаимно простое с f.

Далее необходимо найти число е, такое, что

е d mod f = 1.

Числа d и N будут открытым ключом пользователя, а значение е – закрытым ключом.

Таким образом, на этом этапе у пользователя должна быть следующая информация: открытый ключ N, d закрытый ключ e

Так как пользователь Б хочет получить зашифрованное сообщение от пользователя А, значит пользователь Б должен отправить свой открытый ключ (d, N) пользователю А. Числа Р и Q больше не нужны, однако их нельзя никому сообщать.

Второй этап – шифрование данных. Если абонент А хочет передать некоторые данные абоненту Б, он должен представить свое сообщение в цифровом виде и разбить его на блоки m1, m2, m3, ... , где mi < N. Зашифрованное сообщение будет состоять из блоков сi.

Абонент А шифрует каждый блок своего сообщения по формуле

ci = mid mod N

используя открытые параметры пользователя Б, и пересылает зашифрованное сообщение С=(с1, с2, с3, ...) по открытой линии.

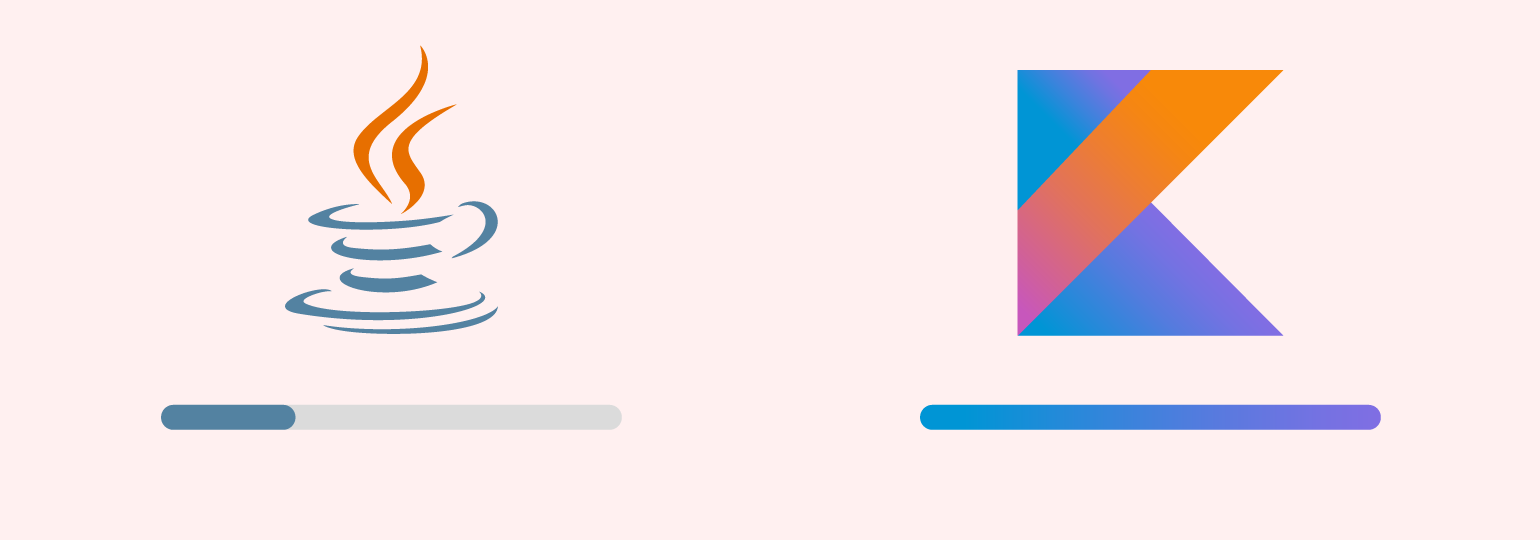
Абонент Б, получивший зашифрованное сообщение, расшифровывает все блоки полученного сообщения по формуле

mi = ce mod N

Все расшифрованные блоки будут точно такими же, как и исходящие от пользователя А.

Злоумышленник, перехватывающий все сообщения и знающий всю открытую информацию, не сможет найти исходное сообщение при больших значениях Р и Q.

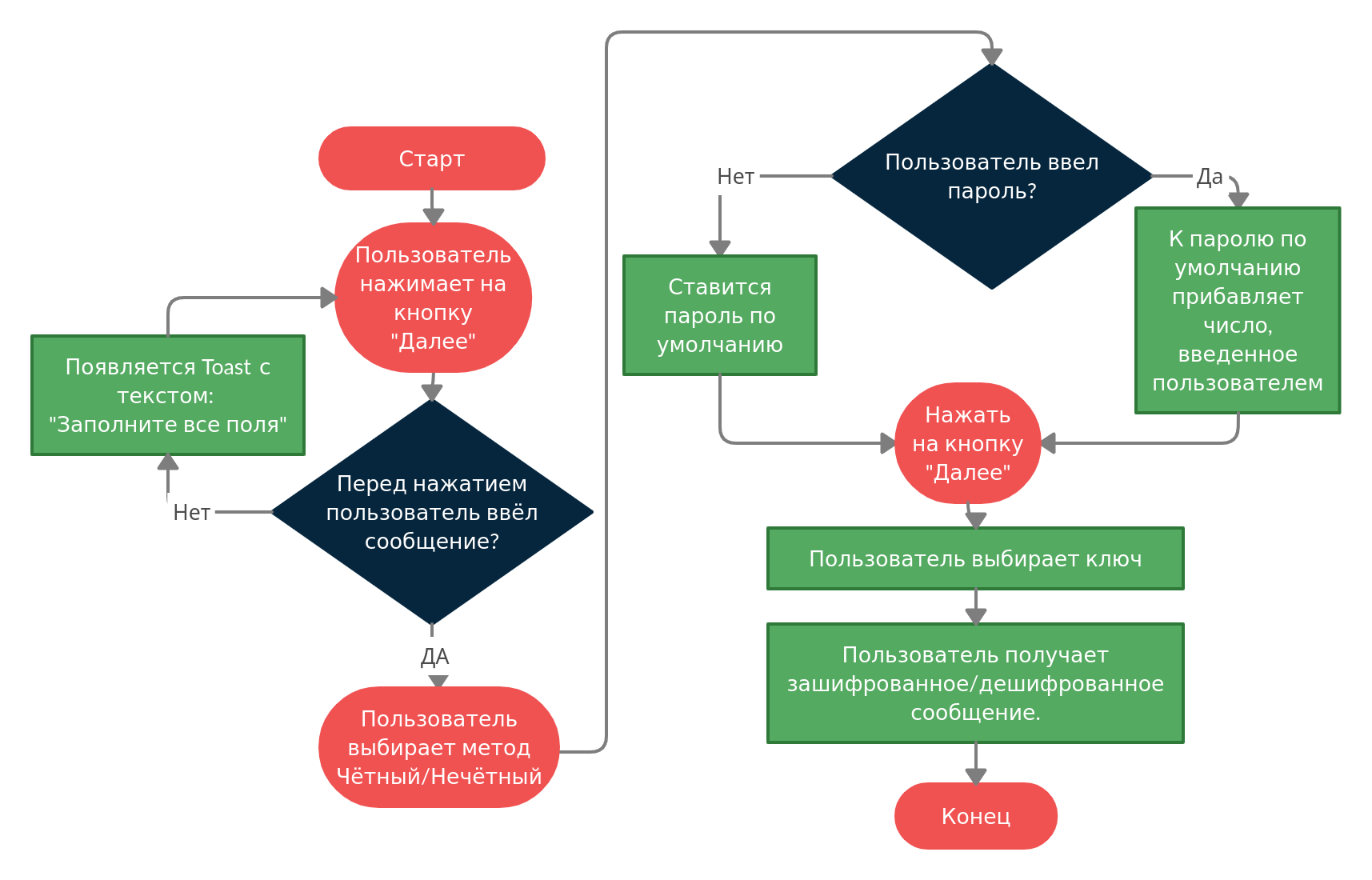
# 3. Разработка приложения «Cryptographer — Шифровальщик»

Проект представляет разработку Android-приложения "MyCryptographer", благодаря которому люди смогут защищать свои данные, а также анонимно общаться в сети.

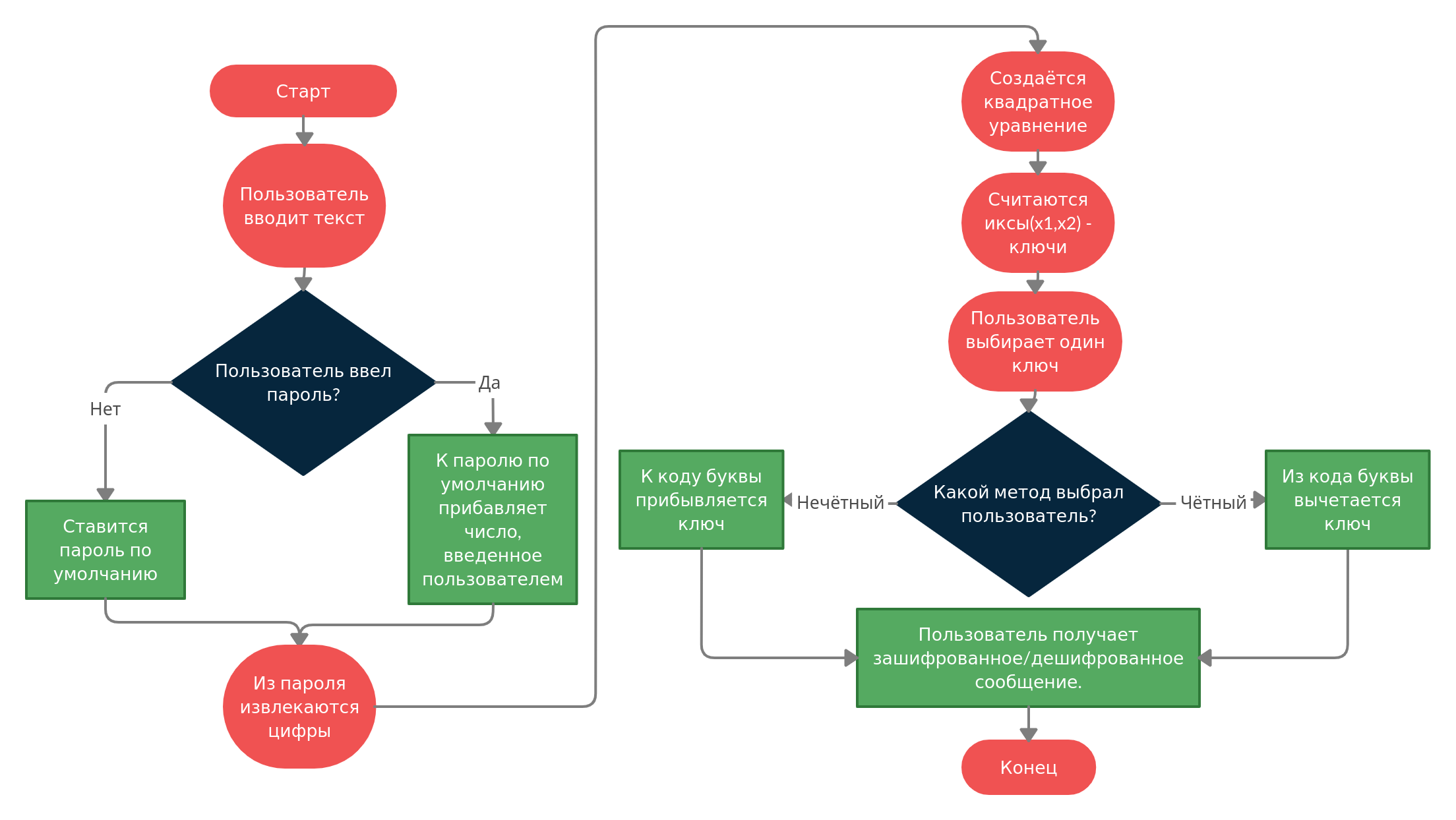
В приложении реализованы алгоритмы DES, AES, BlowFish и шифры Цезаря, заменs символов SwitchChar, eXT.

MyCryptographer изначально был написан на языке программирования Java, но в итоге, приложение полностью переведено на язык программирования Kotlin, что позволило ускорить разработку приложения.

**Схема работы приложения.**



## 3.1. Алгоритм «eXT»:



***Зашифровать/Расшифровать:***   
Пользователь вводит текст/зашифрованное сообщение →

* выбирает метод (чётный/нечётный, нечётный/чётный) →
* вводит пароль →
* выбирает ключ →
* получает зашифрованное/расшифрованное сообщение.

***Пример работы алгоритма:***

Из пароля, который должен быть больше 999, извлекаются цифры. Если пользователь не ввёл пароль, то используется пароль по умолчанию.

Например, пароль=1234, извлекаются 1 2 3 и 4. (a=1, b=2, c=3, e=4)

Далее эти числа представляют квадратное уравнение ax²+bx+c=e.

Затем уравнение преобразуется: ax²+bx+c-e=0 -> ax²+bx-1=0.

Потом вычисляется дискриминант = b \* b - 4\*a\*c.

Потом вычисляются ключи (x):

x1 = (-b + Дискр) / (2\*a)

x2 = (-b - Дискр) / (2\*a)

x1 и x2 - это и есть наши ключи.

Далее каждая буква сообщения меняется.

Если пользователь выбрал нечетный метод, тогда:

Извлекается код буквы, и к нему прибавляется x1.

Например, буква А имеет код=1040, x1=3(к примеру) =>

код=1040+3 а это буква "Г".

Если метод четный, тогда из кода буквы вычитается x1=3 =>

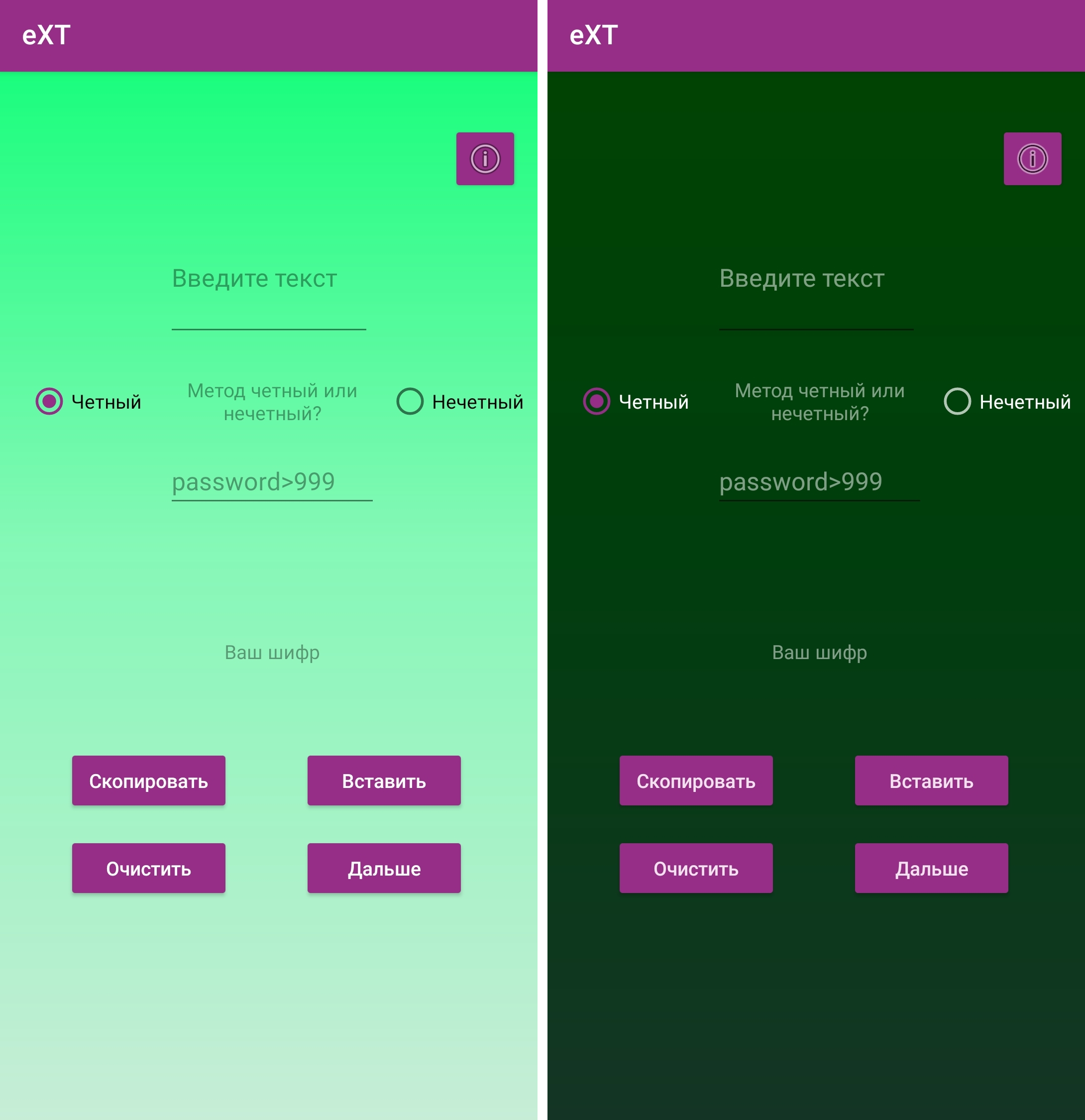
"А" превратиться в "Э".

Таким образом шифруется/дешифруется всё сообщение.

Особенность алгоритма в том, что ключи получаются из пароля, с помощью «переделанного Дискриминанта», который всегда будет возвращать именно два ключа.

Алгоритм, на данный момент, работает только с русскими словами.

## 3.2. Графический пользовательский интерфейс (GUI)

Тема приложения меняется в зависимости от глобальной темы в Android. Если у пользователя стоит светлая тема, то приложение будет со светлой темой. Если стоит темная, то приложение будет в темном оформлении.

Но пользователь может менять цвет акцента кнопок и тулбара.

## 3.3. Экран настроек

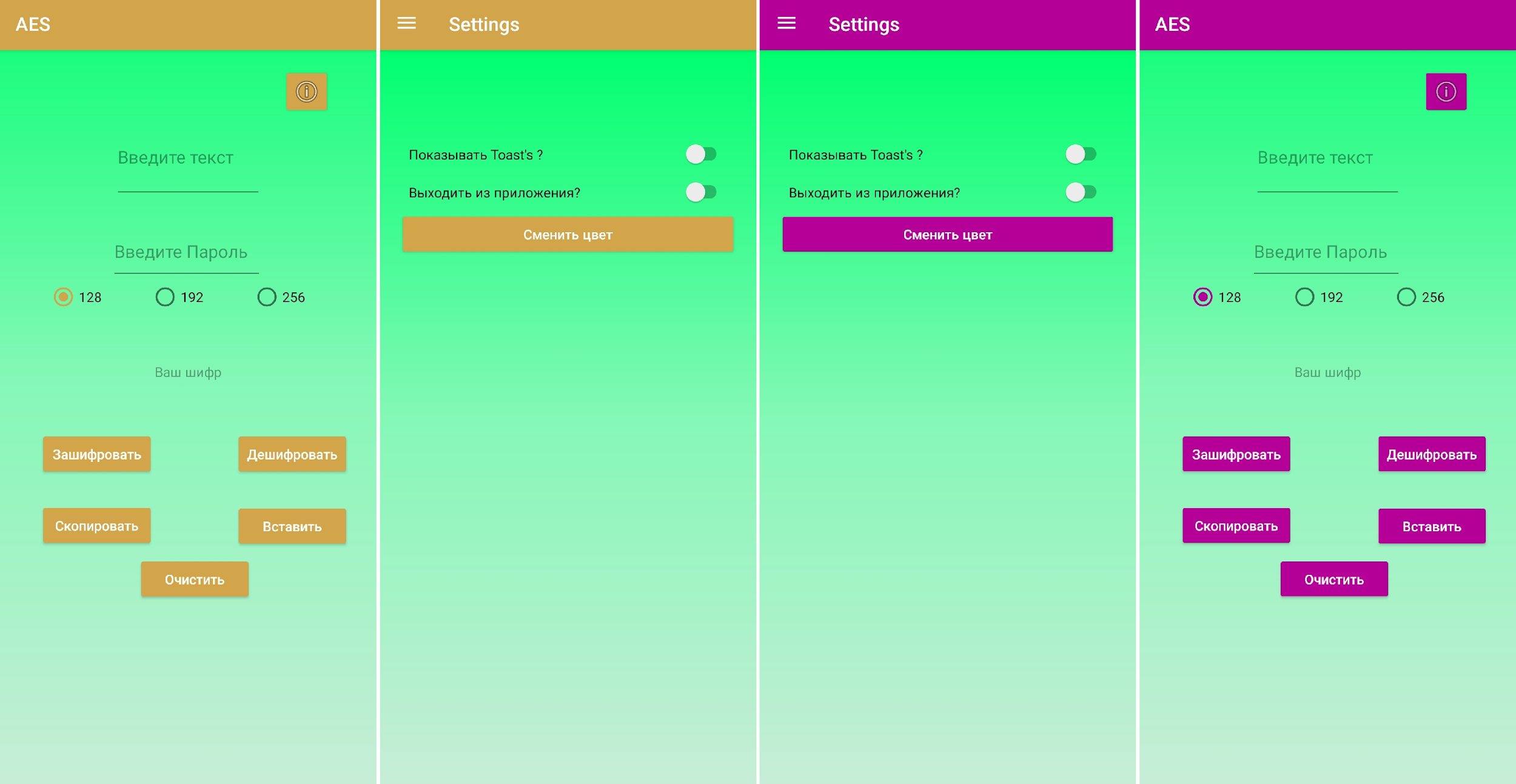
На данный момент у приложения всего пару настроек.

1) Показывать/Скрывать Toast.

2) Выходить из приложения или нет.

3) Смена цвета акцента.

## 3.4. Элемент Toast



Элемент включен по умолчанию. Toast нужен для того, чтобы новый пользователь быстрее привык к интерфейсу программы. Если пользователь знает, что он делает, то он может выключить Toast. Например, если пользователь выбирает алгоритм/шифр, то появляется Toast.

## 3.5. Выход из приложения

У пользователей Android-устройств часто бывают такие моменты, когда они случайно нажимают кнопку «Обратно». Особенно такое часто происходит на маленьких экранах. Если пользователь встречается с данной проблемой достаточно часто, то он может обезопасить себя, выключив пункт «Выходить из приложения?». Чтобы выйти из приложения кнопку «Обратно» нажимать бесполезно, необходимо использовать кнопки «Домой» или «Задачи». С жестами тоже самое.

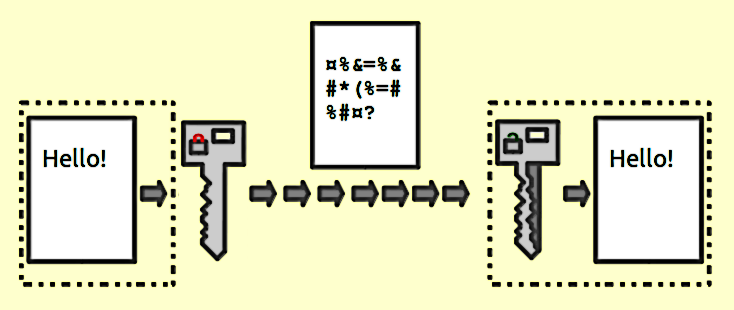
## 3.6. Смена цвета акцента

По умолчанию для кнопок стоит красный цвет, а для тулбара зелено-синий. Если пользователю это не нравится, то он может изменить цвет кнопок и тулбара. После нажатия на кнопку «Сменить цвет» создаётся единый цвет. Этот цвет заменяется как для кнопок, так и для тулбара. Единый цвет записывается в файл настроек как шестнадцатеричная система счисления (RGB).

Обобщающим элементом настроек является запись состояния каждого пункта в файл. Если перезапустить приложение, то все настройки сохранятся. Таким образом интерфейс подстраивается под пользователя, а благодаря настройкам экономит время так, как пользователю не нужно при каждом запуске заново создавать идеальную цветовую гамму.

## 3.7. Обмен ключами шифрования

Чтобы общение по сети было анонимным, пользователям нужно встретиться и договориться о ключе шифрования. На данным момент это единственный способ создания приватного ключа.

В следующей версии программы будут добавлены ассиметричные методы шифрования, например «GPG».

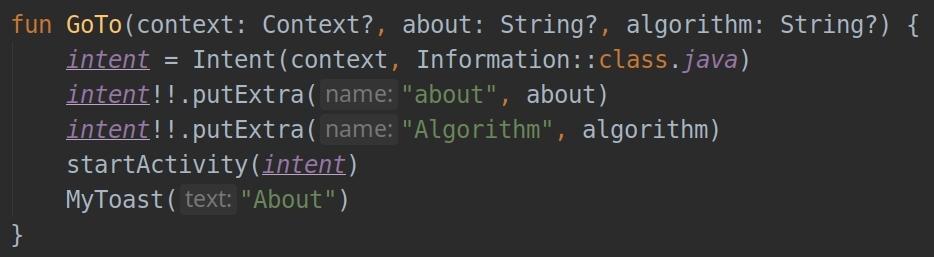
Этот метод позволит пользователям не встречаться. Им не придется договариваться. Так как «GPG» используется 2 ключа. Публичный и приватный. Публичный ключ используется для шифрования, его можно не скрывать от глаз окружающий. А приватный ключ используется для расшифровки, его необходимо хранить в безопасности, никому не показывать.

Благодаря такой схеме расшифровать сообщение может только владелец приватного ключа. Даже тот, кто зашифровывал сообщение, не может произвести обратную операцию.

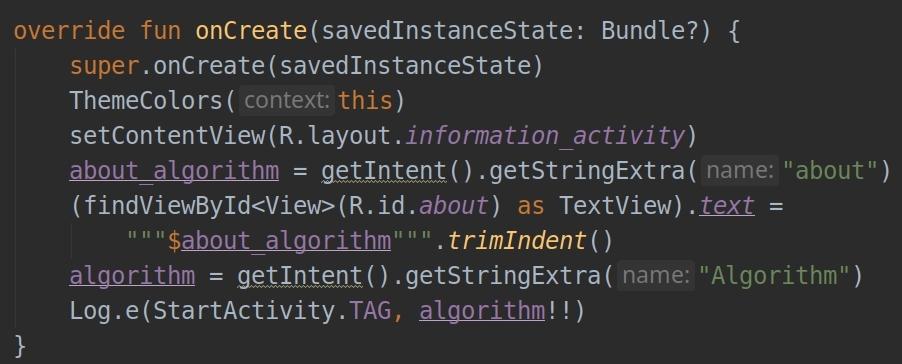
## 3.8. Оптимизация приложения

Из главного класса через методы переходим к разным активностям. (Например, DES, AES)

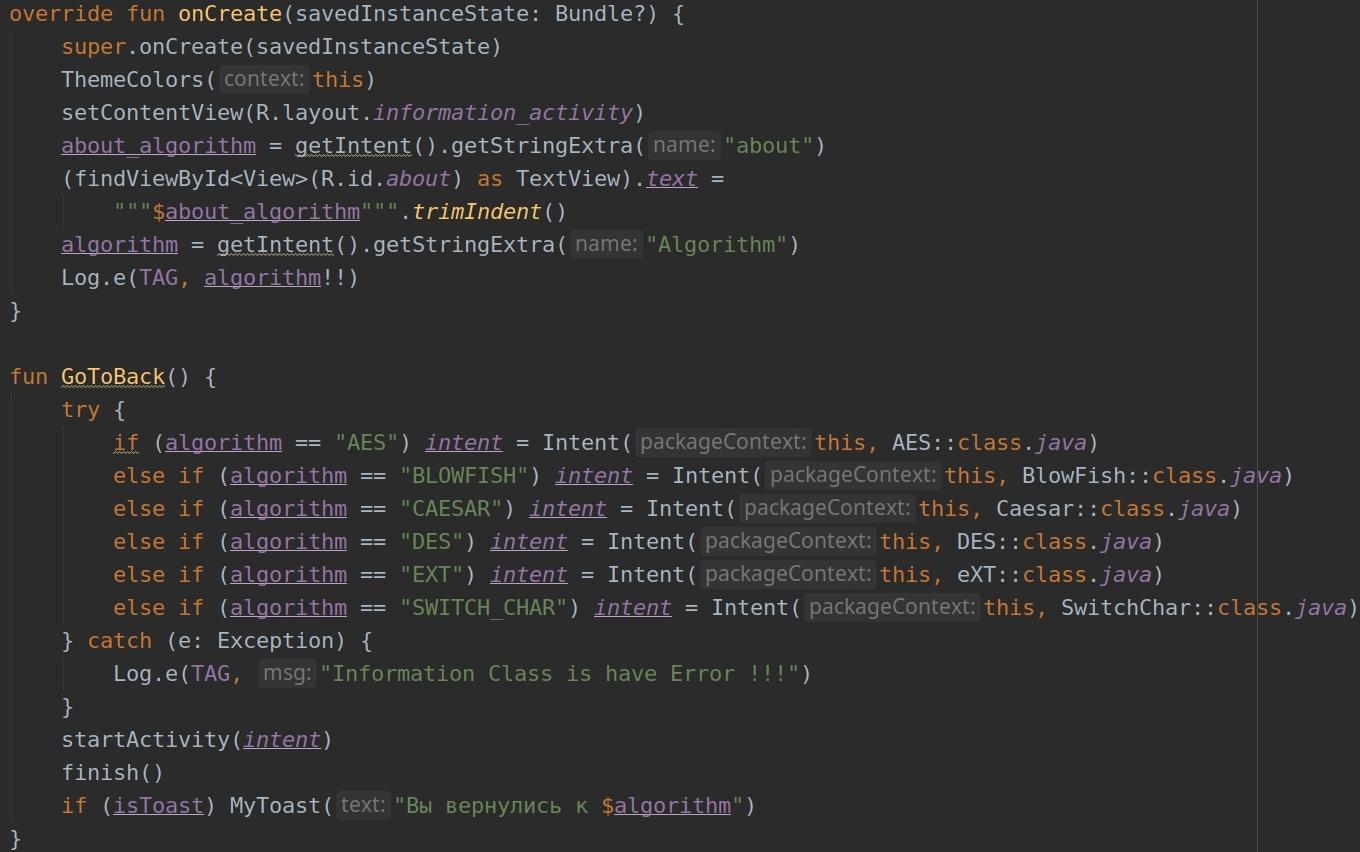
Из DES/AES с помощью «intent» в информационный класс передаем R.string.DES\_About и метод шифрования.

Когда мы открываем информационный класс ставится разметка того алгоритма, который мы передаём.



Из информационного класса, чтобы вернуться назад, мы проверяем какой алгоритм мы сейчас читаем, и через «intent» возвращаемся обратно. Благодаря Intent получилось оптимизировать программу. Всего 6 классов, значит должно быть 6 разметок под каждый класс. Но получилось так, что классов может быть много, а разметка всего одна.



# Заключение

В ходе выполнения работы было создано приложение для смартфона под управлением ОС Android «Cryptographer — Шифровальщик», а также:

* были изучены различные алгоритмы шифрования (AES, DES, BlowFish, RSA);
* изучены шифры подстановки;
* приобретены навыки программирования на языках Java и Kotlin;
* изучен расширяемый язык разметки XML.

Реализованное приложение прошло тестирование среди учащихся школы и получило хорошие отзывы. В будущем планируется его доработка с последующей загрузкой в Google Play и организация поддержки программного продукта.

# Список литературы

* Алгоритм AES URL: <https://intuit.ru/studies/courses/4089/1285/lecture/24208> (дата обращения 10.12.2020)
* Алгоритм DES URL <https://fb.ru/article/390656/algoritm-des-opisanie-i-primer> (дата обращения 25.11.2020)
* Алгоритмы симметричного шифрования Алгоритм Blowfish URL: <https://intuit.ru/studies/courses/28/28/lecture/20414> (дата обращения 12.12.2020)
* Басалова Г. В. Основы криптографии: учебное пособие, М., ИНТУИТ, 2016 Алгоритм DES Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: <https://www.litmir.me/br/?b=228667&p=1> (дата обращения 19.11.2020)
* Введение в криптографию и шифрование URL: <https://habr.com/ru/company/yandex/blog/324866/> (дата обращения 18.11.2020)
* Как устроен AES URL: <https://habr.com/ru/post/112733/> (дата обращения 9.12.2020)
* Криптографические алгоритмы с открытым ключом и их использование Национальный открытый университет Интуит URL: <https://intuit.ru/studies/courses/691/547/lecture/12391?page=1> (дата обращения 25.03.2021)
* Стандарт шифрования данных Data Encryption Standard URL: <https://protect.htmlweb.ru/des.htm> (дата обращения 25.11.2020)
* Стандарт шифрования данных Data Encryption Standard URL: <https://protect.htmlweb.ru/des.htm> (дата обращения 25.11.2020)
* Шифр Цезаря Материал из Национальной библиотеки им. Н. Э. Баумана URL: <https://ru.bmstu.wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F> (дата обращения 30.11.2020)
* Циммерманн Филипп. Введение в криптографию. Перевод В. Миллера, 2004г. URL: <https://www.litmir.me/br/?b=228667&p=1> (дата обращения 17.11.2020)

# Приложения

**AboutProgramFragment**

**Отображает информацию о программе.**

package com.extrime.my\_criptographer

import android.os.Build

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.TextView

import androidx.fragment.app.Fragment

class AboutProgramFragment : Fragment() {

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?):

View? {

val view = inflater.inflate(R.layout.aboutprogram\_activity, container, false)

val textView = view.findViewById<TextView>(R.id.tvAboutProgram)

if (Build.VERSION.SDK\_INT < 26) {

textView.append(getString(R.string.about\_my\_programm))

} else textView.append(getString(R.string.about\_my\_programm\_for\_Android8\_And\_Up))

textView.append("""API Android-устройства: ${Build.VERSION.SDK\_INT}""")

return view

}

**Information -**

**Загружает информацию про конкретный алгоритм.**

package com.extrime.my\_criptographer

import androidx.appcompat.app.AppCompatActivity

import android.content.Intent

import android.os.Bundle

import android.util.Log

import android.view.View

import android.widget.TextView

import android.widget.Toast

import com.extrime.my\_criptographer.ui.Algorithms.Caesar

import com.extrime.my\_criptographer.ui.Algorithms.SwitchChar

import com.extrime.my\_criptographer.ui.Algorithms.eXT

import com.extrime.my\_criptographer.ui.Ciphers.AES

import com.extrime.my\_criptographer.ui.Ciphers.BlowFish

import com.extrime.my\_criptographer.ui.Ciphers.DES

import com.extrime.my\_criptographer.ui.Settings.ThemeColors

import java.lang.Exception

class Information : AppCompatActivity() {

var about\_algorithm: String? = null

var algorithm: String? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

ThemeColors(this)

setContentView(R.layout.information\_activity)

about\_algorithm = getIntent().getStringExtra("about")

(findViewById<View>(R.id.about) as TextView).text =

"""$about\_algorithm""".trimIndent()

algorithm = getIntent().getStringExtra("Algorithm")

Log.e(StartActivity.TAG, algorithm!!)

}

override fun onBackPressed() {

super.onBackPressed()

GoToBack()

}

fun GoToBack() {

try {

if (algorithm == "AES") intent = Intent(this, AES::class.java)

else if (algorithm == "BLOWFISH") intent = Intent(this, BlowFish::class.java)

else if (algorithm == "CAESAR") intent = Intent(this, Caesar::class.java)

else if (algorithm == "DES") intent = Intent(this, DES::class.java)

else if (algorithm == "EXT") intent = Intent(this, eXT::class.java)

else if (algorithm == "SWITCH\_CHAR") intent = Intent(this, SwitchChar::class.java)

} catch (e: Exception) {

Log.e(StartActivity.TAG, "Information Class is have Error !!!")

}

startActivity(intent)

finish()

if (StartActivity.isToast) Toast.makeText(this, "Вы вернулись к $algorithm", Toast.LENGTH\_SHORT).show()

}

}

**MainActivity**

**Главный класс интерфейса, именно он создаёт интерфейс.**

package com.extrime.my\_criptographer

import android.content.Context

import android.content.Intent

import android.graphics.Color

import android.net.Uri

import android.os.Bundle

import android.util.Log

import android.view.View

import android.widget.Switch

import android.widget.Toast

import com.google.android.material.navigation.NavigationView

import androidx.navigation.findNavController

import androidx.navigation.ui.AppBarConfiguration

import androidx.navigation.ui.navigateUp

import androidx.navigation.ui.setupActionBarWithNavController

import androidx.navigation.ui.setupWithNavController

import androidx.drawerlayout.widget.DrawerLayout

import androidx.appcompat.app.AppCompatActivity

import androidx.appcompat.widget.Toolbar

import androidx.cardview.widget.CardView

import com.extrime.my\_criptographer.ui.Algorithms.Caesar

import com.extrime.my\_criptographer.ui.Algorithms.SwitchChar

import com.extrime.my\_criptographer.ui.Algorithms.eXT

import com.extrime.my\_criptographer.ui.Ciphers.AES

import com.extrime.my\_criptographer.ui.Ciphers.BlowFish

import com.extrime.my\_criptographer.ui.Ciphers.DES

import com.extrime.my\_criptographer.ui.Settings.ThemeColors

import java.lang.Exception

import java.util.\*

open class MainActivity : StartActivity() {

private lateinit var appBarConfiguration: AppBarConfiguration

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.start\_activity)

onToolBar()

}

fun onToolBar() {

try {

val toolbar: Toolbar = findViewById(R.id.toolbar)

setSupportActionBar(toolbar)

}catch (e: Exception){

Log.e(TAG, " ToolBar Error !!!")

}finally {

val drawerLayout: DrawerLayout = findViewById(R.id.drawer\_layout)

val navView: NavigationView = findViewById(R.id.nav\_view)

val navController = findNavController(R.id.nav\_host\_fragment)

// Передаем каждый идентификатор меню, как набор идентификаторов,

// потому что каждое меню следует рассматривать как пункты назначения верхнего уровня.

appBarConfiguration = AppBarConfiguration(setOf(

R.id.nav\_algorithms,

R.id.nav\_ciphers,

R.id.nav\_settings,

R.id.nav\_about\_program

), drawerLayout)

setupActionBarWithNavController(navController, appBarConfiguration)

navView.setupWithNavController(navController)

}

}

override fun onPause() {

super.onPause()

}

override fun onSupportNavigateUp(): Boolean {

val navController = findNavController(R.id.nav\_host\_fragment)

return navController.navigateUp(appBarConfiguration) || super.onSupportNavigateUp()

}

override fun onBackPressed() {

if (!isExit) {

val intent = Intent(this, MainActivity::class.java)

startActivity(intent)

}

finish()

}

}

**StartActivity**

**Главный класс, именно он управляет остальными классами.**

StartActivity

package com.extrime.my\_criptographer

import android.content.ClipData

import android.content.ClipboardManager

import android.content.Context

import android.content.Intent

import android.net.Uri

import android.os.Build

import android.os.Bundle

import android.util.Log

import android.view.View

import android.widget.\*

import androidx.appcompat.app.AppCompatActivity

import com.extrime.my\_criptographer.ui.Algorithms.Caesar

import com.extrime.my\_criptographer.ui.Algorithms.SwitchChar

import com.extrime.my\_criptographer.ui.Algorithms.eXT

import com.extrime.my\_criptographer.ui.Ciphers.AES

import com.extrime.my\_criptographer.ui.Ciphers.BlowFish

import com.extrime.my\_criptographer.ui.Ciphers.DES

import com.extrime.my\_criptographer.StartActivity.Companion.NAME\_SETTINGS

import com.extrime.my\_criptographer.ui.Settings.SettingsFragment

import com.extrime.my\_criptographer.ui.Settings.ThemeColors

import kotlinx.android.synthetic.main.nav\_header\_main.view.\*

import java.util.\*

open class StartActivity : AppCompatActivity() {

companion object {

val NAME\_SETTINGS = "settings"

val TAG = "eXTrimeTAG"

var isToast = true

var isExit = false

var copyText = ""

var btn\_Copy: Button? = null

var btn\_Paste: Button? = null

var btn\_Clear: Button? = null

var btn\_info: ImageButton? = null

var btn\_Encrypt: Button? = null

var btn\_Decrypt: Button? = null

}

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

ThemeColors(this)

checkSettings()

}

open fun MyToast(text: String?) {

if (isToast) Toast.makeText(this, text, Toast.LENGTH\_SHORT).show()

}

fun GoTo(text: String?) {

try {

startActivity(intent)

MyToast(text)

finish()

} catch (e: Exception) {

MyToast("MANIFEST!!!")

}

}

fun GoTo(context: Context?, about: String?, algorithm: String?) {

intent = Intent(context, Information::class.java)

intent!!.putExtra("about", about)

intent!!.putExtra("Algorithm", algorithm)

startActivity(intent)

MyToast("About")

}

fun changeTheme(view: View) {

val red: Int = Random().nextInt(255)

val green: Int = Random().nextInt(255)

val blue: Int = Random().nextInt(255)

ThemeColors.setNewThemeColor(this, red, green, blue)

MyToast("Тема сменилась")

}

fun ClearText() {

(findViewById<View>(R.id.input\_text) as EditText).setText("")

(findViewById<View>(R.id.password) as EditText).setText("")

(findViewById<View>(R.id.secret\_text) as TextView).text = ""

MyToast("Очищено")

}

fun CopySecretText() {

try {

copyText = (findViewById<View>(R.id.secret\_text) as TextView).text.toString()

val clipboard = getSystemService(CLIPBOARD\_SERVICE) as ClipboardManager

val clip = ClipData.newPlainText("", copyText)

clipboard.setPrimaryClip(clip)

if (copyText.equals("")) MyToast("Нечего копировать!")

else MyToast("Скопировано в буфер обмена")

} catch (e: java.lang.Exception) {

MyToast("Нечего копировать!")

}

}

fun PasteText() {

(findViewById<View>(R.id.input\_text) as EditText).setText(copyText)

MyToast("Текст вставлен")

}

open fun SetOnClick(onClickListener: View.OnClickListener?) {

// 1

btn\_info = findViewById(R.id.BtnInfo)

btn\_Encrypt = findViewById(R.id.BtnEncrypt)

btn\_Decrypt = findViewById(R.id.BtnDecrypt)

btn\_info!!.setOnClickListener(onClickListener)

btn\_Encrypt!!.setOnClickListener(onClickListener)

btn\_Decrypt!!.setOnClickListener(onClickListener)

}

fun MainClick(view: View) {

when (view.id) {

R.id.BtnCopyText -> CopySecretText()

R.id.BtnPasteText -> PasteText()

R.id.BtnClearText -> ClearText()

R.id.go\_vk1, R.id.go\_vk2, R.id.go\_vk3 -> {

val browserIntent = Intent(Intent.ACTION\_VIEW, Uri.parse(getString(R.string.vk\_link)))

startActivity(browserIntent)

}

R.id.btnAES -> {

intent = Intent(this, AES::class.java)

GoTo("AES")

}

R.id.btnBlowFish -> {

intent = Intent(this, BlowFish::class.java)

GoTo("BlowFish")

}

R.id.btnCaesar -> {

intent = Intent(this, Caesar::class.java)

GoTo("Caesar")

}

R.id.btnDES -> {

intent = Intent(this, DES::class.java)

GoTo("DES")

}

R.id.btnEXT -> {

intent = Intent(this, eXT::class.java)

GoTo("eXT")

}

R.id.btnSwitchChar -> {

intent = Intent(this, SwitchChar::class.java)

GoTo("Switch Char")

}

}

}

fun checkSettings(){

val prefs = this.getSharedPreferences(NAME\_SETTINGS, MODE\_PRIVATE)

isToast = prefs.getBoolean("Toast", true)

isExit = prefs.getBoolean("Exit", false)

}

override fun onPause() {

super.onPause()

checkSettings()

}

override fun onBackPressed() {

super.onBackPressed()

val intent = Intent(this, MainActivity::class.java)

startActivity(intent)

finish()

}

}

AlgorithmsFragment

**Часть интерфейса, отвечающая за разметку с алгоритмами.**

package com.extrime.my\_criptographer.ui.Algorithms

import android.os.Bundle

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import androidx.fragment.app.Fragment

import com.extrime.my\_criptographer.R

class AlgorithmsFragment : Fragment() {

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?):

View? {

return inflater.inflate(R.layout.algorithms\_activity, container, false)

}

}

**Caesar**

**Класс, отвечающий за алгоритм Цезаря.**

package com.extrime.my\_criptographer.ui.Algorithms

import android.os.Bundle

import android.view.View

import android.widget.EditText

import android.widget.TextView

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import java.lang.Exception

import java.lang.StringBuilder

class Caesar : StartActivity() {

private var c = 0

private var y = 0

private var maxStep: Byte = 16

private var conditionStep = "Шаг должен быть < " + (maxStep + 1)

private var algorithm = "CAESAR"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.caesar\_activity)

SetOnClick(caesarListener)

}

private val step: Byte get() = (findViewById<View>(R.id.password) as EditText).text.toString().toByte()

private val caesarListener = View.OnClickListener { view: View ->

when (view.id) {

R.id.BtnInfo -> GoTo(this@Caesar, getString(R.string.Caesar\_About), algorithm)

R.id.BtnEncrypt -> ClickLeft(true)

R.id.BtnDecrypt -> ClickLeft(false)

}

}

private fun ClickLeft(isLeft: Boolean) {

try {

// получаем текст из поля input\_text

val text = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

val step = step

if (step > maxStep) MyToast(conditionStep)

else {

val new\_str = StringBuilder()

for (i in text.indices) {

c = if (isLeft) text[i].toByte() - step // влево

else text[i].toByte() + step // вправо

y =

(c - 1040) % 32 + 1040 // 1040 - это код буквы А, 32 - это кол-во букв 1072 - это код буквы а, 1103 - код буквы я

new\_str.append(y.toChar()) // получаем букву по коду

}

(findViewById<View>(R.id.secret\_text) as TextView).text = new\_str.toString()

}

} catch (e: Exception) {

MyToast("Заполните все поля!")

}

}

}

**eXT**

**Класс, который отвечает за алгоритм eXT.**

package com.extrime.my\_criptographer.ui.Algorithms

import android.widget.RadioButton

import android.widget.TextView

import android.os.Bundle

import android.util.Log

import android.view.View

import android.widget.EditText

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import java.lang.Exception

import java.lang.StringBuilder

class eXT : StartActivity() {

private var input: String? = null

private var output: String? = null

private var algorithm = "EXT"

private var password: Int = 0

private var first\_root: RadioButton? = null

private var second\_root: RadioButton? = null

private var rb\_chetnyu: RadioButton? = null

private var rb\_nechetnyu: RadioButton? = null

private var keyTextView: TextView? = null

private var isEncoding = true

private var x1 = 0

private var x2 = 0

private var z = 0

private var y = 0

private var key = 0

private var count = 0

private var a = 0

private var b = 0

private var c = 0

private var e = 0

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.ext\_activity)

keyTextView = findViewById(R.id.key)

first\_root = findViewById(R.id.radio\_1)

second\_root = findViewById(R.id.radio\_2)

SetShowElements(false)

rb\_chetnyu = findViewById(R.id.radio\_3)

rb\_nechetnyu = findViewById(R.id.radio\_4)

count = 0

SetOnClick(eXT\_Listener)

}

override fun SetOnClick(onClickListener: View.OnClickListener?) {

Companion.btn\_info = findViewById(R.id.BtnInfo)

Companion.btn\_Encrypt = findViewById(R.id.BtnEncrypt)

Companion.btn\_info!!.setOnClickListener(onClickListener)

Companion.btn\_Encrypt!!.setOnClickListener(onClickListener)

btn\_Clear = findViewById(R.id.BtnClearText)

btn\_Clear!!.setOnClickListener(onClickListener)

}

private val eXT\_Listener = View.OnClickListener { view: View ->

when (view.id) {

R.id.BtnInfo -> GoTo(this@eXT, getString(R.string.eXT\_About), algorithm)

R.id.BtnEncrypt -> eXT\_process()

R.id.BtnClearText -> {

ClearText()

SetShowElements(false)

}

}

}

fun SetShowElements(Show: Boolean) {

if (Show) {

keyTextView!!.visibility = View.VISIBLE

first\_root!!.visibility = View.VISIBLE

second\_root!!.visibility = View.VISIBLE

} else {

keyTextView!!.visibility = View.GONE

first\_root!!.visibility = View.GONE

second\_root!!.visibility = View.GONE

}

}

fun getInput(): String {

return (findViewById<View>(R.id.input\_text) as EditText).text.toString()

}

fun getPassword(): Int {

var pass = (findViewById<View>(R.id.password) as EditText).text.toString()

Log.e(TAG, "pass = $pass")

if (pass == "" || pass.toInt() < 999) return 1283

else return pass.toInt() + 1283

}

fun setKeyText(str: String?) {

(findViewById<View>(R.id.key) as TextView).text = str

}

fun setSecretText(str: String?) {

(findViewById<View>(R.id.secret\_text) as TextView).text = str

}

fun eXT\_process() {

try {

if (count == 0) {

First\_Step()

if (input == "" || password == 0) MyToast("Заполните все поля!")

Discriminant(a, b, c, e)

Second\_Step()

SetShowElements(true)

CheckSelectKey()

CheckIsEncode()

count++

setSecretText("")

Log.e(TAG, "count if = " + count)

} else if (count == 1) {

CheckSelectKey()

CheckIsEncode()

output = SwitchSymbol(input, key, isEncoding)

setSecretText(output)

Log.e(TAG, "count else if1 = " + count)

count = 0

Log.e(TAG, "count else if2 = " + count)

}

} catch (e: Exception) {

Log.e(TAG, "EXCEPTION\n$e")

if (input == "" || password == 0) MyToast("Заполните все поля!")

}

}

fun First\_Step() {

input = getInput()

password = getPassword()

a = password / 1000

b = password / 100 % 10

c = password / 10 % 10

e = password % 10

}

fun Second\_Step() {

setKeyText("""Какой ключ выбрать? $x1 или $x2""")

first\_root!!.text = x1.toString()

second\_root!!.text = x2.toString()

}

fun CheckSelectKey() {

if (first\_root!!.isChecked) key = x1

else if (second\_root!!.isChecked) key = x2

}

fun CheckIsEncode() {

if (rb\_chetnyu!!.isChecked) isEncoding = false

else if (rb\_nechetnyu!!.isChecked) isEncoding = true

}

fun SwitchSymbol(str: String?, step: Int, isEncode: Boolean): String {

var zero = 1

val new\_str = StringBuilder()

if (isEncode) {

zero = 0

}

for (i in 0 until str!!.length) {

// нечетный

// получаем букву по коду

// 1040 - это код буквы А, 32 - это кол-во букв

if (i % 2 == zero) {

z = str[i].toInt() + step

} else {

z = str[i].toInt() - step

}

y = (z - 1040) % 32 + 1040 // 1040 - это код буквы А, 32 - это кол-во букв

new\_str.append(y.toChar()) // получаем букву по коду

}

Log.e(TAG, "ответ = $new\_str\nЧетность = $isEncode")

return new\_str.toString()

}

fun Discriminant(a: Int, b: Int, c: Int, e: Int) {

try {

this@eXT.c -= e

val Discr = b \* b - 4 \* a \* c

x1 = (-b + Discr) / (2 \* a)

x2 = (-b - Discr) / (2 \* a)

Log.e("INFO EXT", "Discr = $Discr x1 = $x1 x2 = $x2")

} catch (exception: Exception) {

Log.e("ERROR", "Введите число больше 999")

}

}

}

**SwitchChar**

**Этот класс отвечает за алгоритм смены символов.**

package com.extrime.my\_criptographer.ui.Algorithms

import android.widget.EditText

import android.os.Bundle

import android.view.View

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

class SwitchChar : StartActivity() {

private var edit\_textET: EditText? = null

private var from\_charET: EditText? = null

private var to\_charET: EditText? = null

private var from\_char: String? = null

private var to\_char: String? = null

private var edit\_text: String? = null

var algorithm = "SWITCH\_CHAR"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.switch\_char\_activity)

from\_charET = findViewById(R.id.password)

to\_charET = findViewById(R.id.secret\_text)

edit\_textET = findViewById(R.id.input\_text)

}

fun onClick(context: View?) {

GoTo(this, getString(R.string.SwitchChar\_About), algorithm)

MyToast("About")

}

fun Clear\_Text(context: View?) {

from\_charET!!.setText("")

to\_charET!!.setText("")

edit\_textET!!.setText("")

MyToast("Очищено")

}

fun replace(context: View?) {

from\_char = from\_charET!!.text.toString()

to\_char = to\_charET!!.text.toString()

edit\_text = edit\_textET!!.text.toString()

edit\_text = edit\_text!!.replace(from\_char!!.toRegex(), to\_char!!)

edit\_textET!!.setText(edit\_text)

MyToast("Изменено")

}

}

**Ciphers**

**DES**

**Класс, который работает с шифром DES.**

package com.extrime.my\_criptographer.ui.Ciphers

import android.os.Bundle

import kotlin.Throws

import android.content.ClipData

import android.content.ClipboardManager

import android.content.Intent

import android.util.Base64

import android.util.Log

import android.view.View

import android.widget.\*

import com.extrime.my\_criptographer.MainActivity

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import java.io.ByteArrayInputStream

import java.io.ByteArrayOutputStream

import java.io.IOException

import java.lang.Exception

import java.lang.StringBuilder

import java.nio.charset.StandardCharsets

import java.security.InvalidKeyException

import java.security.NoSuchAlgorithmException

import java.security.spec.InvalidKeySpecException

import javax.crypto.\*

import javax.crypto.spec.DESKeySpec

open class DES : StartActivity() {

private var cipher: Cipher? = null

private var copyText: String? = null

private val algorithm = "DES"

private var output: ByteArrayOutputStream? = null

var btn\_info: ImageButton? = null

var btn\_Encrypt: Button? = null

var btn\_Decrypt: Button? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.blowfish\_and\_des\_activity)

SetOnClick(desListener)

create\_cipher()

}

private fun create\_cipher() {

try {

val key = KeyGenerator.getInstance(algorithm).generateKey()

cipher = Cipher.getInstance(algorithm)

cipher!!.init(Cipher.DECRYPT\_MODE, key)

} catch (e: NoSuchAlgorithmException) {

e.printStackTrace()

} catch (e: NoSuchPaddingException) {

e.printStackTrace()

} catch (e: InvalidKeyException) {

e.printStackTrace()

}

}

private val desListener = View.OnClickListener { view: View ->

when (view.id) {

R.id.BtnInfo -> GoTo(this, getString(R.string.DES\_About), algorithm)

R.id.BtnEncrypt -> DES\_Encode()

R.id.BtnDecrypt -> DES\_Decode()

}

}

@Throws(IOException::class, InvalidKeyException::class,

NoSuchAlgorithmException::class, InvalidKeySpecException::class)

private fun code(input: String, key: String): String {

DES\_CreateKeys(input, key, true)

return Base64.encodeToString(output!!.toByteArray(), Base64.DEFAULT)

}

@Throws(IOException::class, InvalidKeyException::class,

NoSuchAlgorithmException::class, InvalidKeySpecException::class)

private fun DES\_CreateKeys(input: String, key: String, Encode: Boolean) {

val KS = DESKeySpec(key.toByteArray(StandardCharsets.UTF\_8))

val skf = SecretKeyFactory.getInstance(algorithm)

val desKey = skf.generateSecret(KS)

val cis: CipherInputStream =

if (Encode) {

cipher!!.init(Cipher.ENCRYPT\_MODE, desKey)

CipherInputStream(ByteArrayInputStream(input.toByteArray()), cipher)

} else {

cipher!!.init(Cipher.DECRYPT\_MODE, desKey)

CipherInputStream(ByteArrayInputStream(Base64.decode(input, Base64.DEFAULT)), cipher)

}

output = ByteArrayOutputStream()

val buffer = ByteArray(64)

var numBytes: Int

while (cis.read(buffer).also { numBytes = it } != -1) {

output!!.write(buffer, 0, numBytes)

}

}

@Throws(IOException::class, InvalidKeyException::class,

NoSuchAlgorithmException::class, InvalidKeySpecException::class)

private fun decode(input: String, key: String): String {

DES\_CreateKeys(input, key, false)

return output.toString()

}

private fun CheckLengthPassword(password: String): String {

val startStr = password.length

val new\_password = StringBuilder()

for (i in 0 until startStr) {

new\_password.append(password[i])

}

for (i in startStr..7) {

new\_password.append("#")

}

Log.e("passwd", new\_password.toString())

return new\_password.toString()

}

private fun DES\_Encode() {

val input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

var password = (findViewById<View>(R.id.password) as EditText).text.toString()

val output = findViewById<TextView>(R.id.secret\_text)

try {

password = CheckLengthPassword(password)

val secretText = code(input, password)

output.text = secretText

if (input != decode(secretText, password)) {

// Расшифровка текста приводит к другому результату.

MyToast("Внутренняя ошибка шифрования")

} else MyToast("Зашифровано")

} catch (e: Exception) {

MyToast("Заполните все поля!")

}

}

private fun DES\_Decode() {

val input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

var password = (findViewById<View>(R.id.password) as EditText).text.toString()

try {

password = CheckLengthPassword(password)

val output = decode(input, password)

(findViewById<View>(R.id.secret\_text) as TextView).text = output

MyToast("Дешифровано")

} catch (e: Exception) {

MyToast("Заполните все поля!")

}

}

override fun onBackPressed() {

super.onBackPressed()

val intent = Intent(this, MainActivity::class.java)

startActivity(intent)

MyToast("Шифры")

finish()

}

}

**AES**

**Класс, который работает с шифром AES.**

package com.extrime.my\_criptographer.ui.Ciphers

import android.widget.RadioButton

import android.os.Bundle

import android.view.View

import android.widget.EditText

import android.widget.TextView

import com.extrime.my\_criptographer.Base64

import com.extrime.my\_criptographer.R

import java.lang.Exception

import java.nio.charset.StandardCharsets

import java.security.MessageDigest

import java.security.NoSuchAlgorithmException

import java.util.Arrays

import javax.crypto.Cipher

import javax.crypto.SecretKey

import javax.crypto.spec.SecretKeySpec

class AES : DES() {

private var secretKey: SecretKey? = null

private var cipher: Cipher? = null

var algorithm = "AES" // AES/ECB/PKCS5Padding

var input: String? = null

var secretText: String? = null

private var bits\_key = 0

private var KeySecret: String? = null

private var rb\_128: RadioButton? = null

private var rb\_192: RadioButton? = null

private var rb\_256: RadioButton? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.aes\_activity)

input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

rb\_128 = findViewById(R.id.radio\_128)

rb\_192 = findViewById(R.id.radio\_192)

rb\_256 = findViewById(R.id.radio\_256)

SetOnClick(aesListener)

}

private val aesListener = View.OnClickListener { view: View ->

when (view.id) {

R.id.BtnInfo -> GoTo(this, getString(R.string.AES\_About), algorithm)

R.id.BtnEncrypt -> AES\_Encode()

R.id.BtnDecrypt -> AES\_Decode()

}

}

fun setKey(myKey: String) {

try {

var key: ByteArray? = myKey.toByteArray(StandardCharsets.UTF\_8)

val sha = MessageDigest.getInstance("SHA-1")

key = sha.digest(key)

key = Arrays.copyOf(key, bits\_key / 8)

secretKey = SecretKeySpec(key, algorithm)

} catch (e: NoSuchAlgorithmException) {

e.printStackTrace()

}

}

fun encode(strToEncrypt: String, secret: String): String? {

try {

setKey(secret)

cipher = Cipher.getInstance(algorithm)

cipher!!.init(Cipher.ENCRYPT\_MODE, secretKey)

val arr\_encode = cipher!!.doFinal(strToEncrypt.toByteArray(StandardCharsets.UTF\_8))

return Base64.getEncoder().encodeToString(arr\_encode)

} catch (e: Exception) {

println("Error while encrypting: $e")

}

return null

}

fun decode(strToDecrypt: String?, secret: String): String? {

try {

setKey(secret)

cipher = Cipher.getInstance(algorithm)

cipher!!.init(Cipher.DECRYPT\_MODE, secretKey)

return String(cipher!!.doFinal(Base64.getDecoder().decode(strToDecrypt)))

} catch (e: Exception) {

println("Error while decrypting: $e")

}

return null

}

fun CheckBits() {

if (rb\_128!!.isChecked) bits\_key = 128

if (rb\_192!!.isChecked) bits\_key = 192

if (rb\_256!!.isChecked) bits\_key = 256

if (bits\_key != 128 && bits\_key != 192 && bits\_key != 256) {

MyToast("Измените битность пароля")

}

}

fun AES\_Encode() {

try {

input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

KeySecret = (findViewById<View>(R.id.password) as EditText).text.toString()

CheckBits()

secretText = encode(input!!, KeySecret!!)

(findViewById<View>(R.id.secret\_text) as TextView).text = secretText

MyToast("Зашифровано")

} catch (e: Exception) {

if (bits\_key == 0) {

MyToast("Заполните все поля!")

} else MyToast("Ошибка!")

}

}

fun AES\_Decode() {

try {

input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

KeySecret = (findViewById<View>(R.id.password) as EditText).text.toString()

CheckBits()

secretText = decode(input, KeySecret!!)

if (secretText == null) {

MyToast("Ошибка дешифрования")

(findViewById<View>(R.id.secret\_text) as TextView).text = null

} else {

(findViewById<View>(R.id.secret\_text) as TextView).text = secretText

MyToast("Дешифровано")

}

} catch (e: Exception) {

MyToast("Ошибка!")

}

}

}

**BlowFish**

**Класс, который работает с шифром BlowFish.**

package com.extrime.my\_criptographer.ui.Ciphers

import android.os.Bundle

import android.view.View

import android.widget.EditText

import android.widget.TextView

import com.extrime.my\_criptographer.Base64

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import java.lang.Exception

import java.nio.charset.StandardCharsets

import java.security.NoSuchAlgorithmException

import javax.crypto.Cipher

import javax.crypto.NoSuchPaddingException

import javax.crypto.spec.SecretKeySpec

class BlowFish : StartActivity() {

var KeyData : ByteArray? = null

private var cipher: Cipher? = null

var KS: SecretKeySpec? = null

var key: String? = null

var input: String? = null

var algorithm = "BLOWFISH"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.blowfish\_and\_des\_activity)

SetOnClick(blowfishListener)

}

private val blowfishListener = View.OnClickListener { view: View ->

when (view.id) {

R.id.BtnInfo -> GoTo(this@BlowFish, getString(R.string.BlowFish\_About), algorithm)

R.id.BtnEncrypt -> BlowFish\_Encode()

R.id.BtnDecrypt -> BlowFish\_Decode()

}

}

private fun createCipherAndKey(key: String) {

try {

cipher = Cipher.getInstance(algorithm)

KeyData = key.toByteArray()

KS = SecretKeySpec(KeyData, algorithm)

} catch (e: NoSuchAlgorithmException) {

e.printStackTrace()

} catch (e: NoSuchPaddingException) {

e.printStackTrace()

}

}

private fun BlowFish\_Encode() {

try {

input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

key = (findViewById<View>(R.id.password) as EditText).text.toString()

createCipherAndKey(key!!)

cipher!!.init(Cipher.ENCRYPT\_MODE, KS)

val encrypted = cipher!!.doFinal(input!!.toByteArray(StandardCharsets.UTF\_8))

val encode = Base64.getEncoder().encodeToString(encrypted)

(findViewById<View>(R.id.secret\_text) as TextView).text = encode

MyToast("Зашифровано")

} catch (e: Exception) {

MyToast("Заполните все поля!")

}

}

private fun BlowFish\_Decode() {

try {

input = (findViewById<View>(R.id.input\_text) as EditText).text.toString()

key = (findViewById<View>(R.id.password) as EditText).text.toString()

createCipherAndKey(key!!)

cipher!!.init(Cipher.DECRYPT\_MODE, KS)

val decode = String(cipher!!.doFinal(Base64.getDecoder().decode(input)))

println("decode = $decode")

(findViewById<View>(R.id.secret\_text) as TextView).text = decode

MyToast("Дешифровано")

} catch (e: Exception) {

if (input == null || key == null) MyToast("Заполните все поля!")

else MyToast("Ошибка дешифрования")

(findViewById<View>(R.id.secret\_text) as TextView).text = ""

}

}

}

**Класс, который работает с шифром RSA.**

package com.extrime.my\_criptographer.ui.Ciphers

import android.content.ClipData

import android.content.ClipboardManager

import android.content.Intent

import android.os.Bundle

import android.view.View

import android.widget.Button

import android.widget.EditText

import android.widget.TextView

import com.extrime.my\_criptographer.Base64

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import java.lang.Exception

import java.security.\*

import java.security.spec.InvalidKeySpecException

import java.security.spec.PKCS8EncodedKeySpec

import java.security.spec.X509EncodedKeySpec

import javax.crypto.Cipher

class RSA : StartActivity() {

private var privateKey: PrivateKey

private var publicKey: PublicKey

val algorithm = "RSA"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.rsa\_activity)

}

fun getStringPublicKey() : String { return Base64.getEncoder().encodeToString(this.publicKey.encoded); }

fun getStringPrivateKey() : String { return Base64.getEncoder().encodeToString(this.privateKey.encoded); }

fun RSA\_OnClick(view: View){

when(view.id){

R.id.BtnEncrypt -> {

try {

var text = findViewById<EditText>(R.id.input\_text).text.toString()

var key = findViewById<EditText>(R.id.password).text.toString()

var secret\_text = findViewById<TextView>(R.id.secret\_text)

if (key.equals("")) secret\_text.text = encrypt(text, getStringPublicKey())

else secret\_text.text = encrypt(text, key)

MyToast("Зашифровано")

}catch (e: Exception){

MyToast("Вы уже зашифровали сообщение!")

}

}

R.id.BtnDecrypt -> {

try {

var text = findViewById<EditText>(R.id.input\_text).text.toString()

var key = findViewById<EditText>(R.id.password).text.toString()

var secret\_text = findViewById<TextView>(R.id.secret\_text)

if (key.equals("")) secret\_text.text = decrypt(text, getStringPrivateKey())

else secret\_text.text = decrypt(text, key)

MyToast("Дешифровано")

} catch (e: Exception){

MyToast("Введите приватный ключ")

var secret\_text = findViewById<TextView>(R.id.secret\_text)

secret\_text.text = ""

}

}

R.id.BtnInfo -> GoTo(this, getString(R.string.RSA\_About), algorithm)

R.id.BtnPublicKey -> copyKeys(R.id.BtnPublicKey)

R.id.BtnPrivateKey -> copyKeys(R.id.BtnPrivateKey)

}

}

fun copyKeys(button : Int){

try {

var copyKey = ""

var PublicOrPrivate = ""

if(button == R.id.BtnPublicKey) {

copyKey = getStringPublicKey()

PublicOrPrivate = "Публичный"

}

else if (button == R.id.BtnPrivateKey) {

copyKey = getStringPrivateKey()

PublicOrPrivate = "Приватный"

}

val clipboard = getSystemService(CLIPBOARD\_SERVICE) as ClipboardManager

val clip = ClipData.newPlainText("", copyKey)

clipboard.setPrimaryClip(clip)

if (copyKey.equals("")) MyToast("Нечего копировать!")

else MyToast("$PublicOrPrivate ключ скопирован в буфер обмена")

} catch (e: java.lang.Exception) { MyToast("Нечего копировать!")}

}

init {

val keyGen = KeyPairGenerator.getInstance(algorithm)

keyGen.initialize(1024)

val pair = keyGen.generateKeyPair()

privateKey = pair.private

publicKey = pair.public

}

private fun getPublicKey(base64PublicKey: String): PublicKey? {

var publicKey: PublicKey? = null

try {

val keySpec = X509EncodedKeySpec(Base64.getDecoder().decode(base64PublicKey.toByteArray()))

val keyFactory = KeyFactory.getInstance(algorithm)

publicKey = keyFactory.generatePublic(keySpec)

}

catch (e: Exception) { e.printStackTrace() }

return publicKey

}

private fun getPrivateKey(base64PrivateKey: String): PrivateKey? {

var privateKey: PrivateKey? = null

try {

val keySpec = PKCS8EncodedKeySpec(Base64.getDecoder().decode(base64PrivateKey.toByteArray()))

val keyFactory = KeyFactory.getInstance(algorithm)

privateKey = keyFactory!!.generatePrivate(keySpec)

}

catch (e: InvalidKeySpecException) { e.printStackTrace() }

return privateKey

}

@Throws(Exception::class)

private fun encrypt(data: String, publicKey: String): String {

val cipher = Cipher.getInstance("$algorithm/ECB/PKCS1Padding")

cipher.init(Cipher.ENCRYPT\_MODE, getPublicKey(publicKey))

return Base64.getEncoder().encodeToString(cipher.doFinal(data.toByteArray()))

}

@Throws(Exception::class)

private fun decrypt(string\_data: String, string\_PrivateKey: String): String {

val byte\_data = Base64.getDecoder().decode(string\_data.toByteArray())

val privateKey = getPrivateKey(string\_PrivateKey)

val cipher = Cipher.getInstance("$algorithm/ECB/PKCS1Padding")

cipher.init(Cipher.DECRYPT\_MODE, privateKey)

return String(cipher.doFinal(byte\_data))

}

// companion object {

// @Throws(Exception::class)

// @JvmStatic

// fun main(args: Array<String>) {

// try {

// val kotlinRsaUtil = Kotlin\_RSA\_Util()

// val encryptedString = kotlinRsaUtil.encrypt("Приветик", kotlinRsaUtil.getStringPublicKey())

//// val encryptedString = "fW0JJwycXsm+cX7M5v0dPQZsB8yCt1y5qexJZo5+gVmEhCKQY88ysFgOT5IjQAkuUgoRU4hnhUCWwHM3ipKX0/geWyEY9s7UoKg9matl70C6ALhHmqnDuxN2WxsTLCbLT0B0kjdcShvu55IUNbuWhzPRkvZ3a3NIAkkPFu045gY="

//// val encryptedString = "RKq0/gifvb46e2jVr71I1H97Pj+n0TLVj940eeFwo6yqcUr2XWtl/IJVfTixbtKggLy4tl8mx8W5L0QsDAmvYJ96hmL7lxrCJ+ygJqAHIJeCZHTyQL7PMq4H2rK+5gaLAF9C7ruEHkndpzzLJlIwygEK1aOYpwmOf+tCHz8qm+g="

// val decryptedString = kotlinRsaUtil.decrypt(encryptedString, kotlinRsaUtil.getStringPrivateKey())

// println("\nПубличный ключ: ${kotlinRsaUtil.publicKey} " +

// "\nПриватный ключ: ${kotlinRsaUtil.privateKey} " +

// "\n\nПубличный ключ-строка: ${kotlinRsaUtil.getStringPublicKey()} " +

// "\nПриватный ключ-строка: ${kotlinRsaUtil.getStringPrivateKey()} " +

// "\n\nЗашифрованная строка: $encryptedString " +

// "\nРасшифрованная строка: $decryptedString")

// } catch (e: NoSuchAlgorithmException) {

// System.err.println(e.message)

// }

// }

// }

}

**CiphersFragment**

**Часть интерфейса, отвечающая за разметку с шифрами.**

package com.extrime.my\_criptographer.ui.Ciphers

import android.os.Build

import android.os.Bundle

import android.util.Log

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.Button

import androidx.fragment.app.Fragment

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

class CiphersFragment : Fragment() {

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?):

View? {

val view = inflater.inflate(R.layout.ciphers\_activity, container, false)

// checkVersion(view)

return view

}

private fun checkVersion(view: View) {

val version = Build.VERSION.SDK\_INT

Log.e(StartActivity.TAG, "version: $version")

// if (Build.VERSION.SDK\_INT > 26) {

val button\_aes = view.findViewById<Button>(R.id.btnAES)

val button\_blowfish = view.findViewById<Button>(R.id.btnBlowFish)

button\_aes.visibility = View.GONE // скрываем кнопки

button\_blowfish.visibility = View.GONE

// }

}

}

**Settings**

**SettingsFragment -**

**Часть интерфейса, отвечающая за разметку с настройками.**

package com.extrime.my\_criptographer.ui.Settings

import android.content.Context.MODE\_PRIVATE

import android.content.SharedPreferences

import android.os.Bundle

import android.util.Log

import android.view.LayoutInflater

import android.view.View

import android.view.ViewGroup

import android.widget.Switch

import androidx.fragment.app.Fragment

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity.Companion.TAG

import com.extrime.my\_criptographer.StartActivity.Companion.isToast

import com.extrime.my\_criptographer.StartActivity.Companion.NAME\_SETTINGS

import com.extrime.my\_criptographer.StartActivity.Companion.isExit

class SettingsFragment : Fragment() {

companion object {

private var savedToast = true

private var savedExit = false

}

private lateinit var switchToast: Switch

private lateinit var switchExit: Switch

private lateinit var prefs : SharedPreferences

override fun onCreateView(inflater: LayoutInflater, container: ViewGroup?, savedInstanceState: Bundle?):

View? {

val view = inflater.inflate(R.layout.settings\_activity, container, false)

prefs = this.activity!!.getSharedPreferences(NAME\_SETTINGS, MODE\_PRIVATE)

initToast(view)

initExit(view)

return view

}

fun initExit(view: View){

switchExit = view.findViewById(R.id.setExit)

savedExit = prefs.getBoolean("Exit", false)

switchExit.setChecked(savedExit)

checkExit()

switchExit.setOnClickListener{ checkExit() }

}

fun checkExit(){

if (switchExit.isChecked) {

isExit = true

Log.e(TAG, "EXIT ON")

} else {

Log.e(TAG, "EXIT OFF")

isExit = false

switchExit.isChecked = false

}

}

fun initToast(view: View) {

switchToast = view.findViewById(R.id.setToast)

savedToast = prefs.getBoolean("Toast", true)

switchToast.setChecked(savedToast)

checkToast()

switchToast.setOnClickListener{ checkToast() }

}

fun checkToast(){

if (switchToast.isChecked) {

isToast = true

Log.e(TAG, "TOAST ON")

} else {

Log.e(TAG, "TOAST OFF")

isToast = false

switchToast.isChecked = false

}

}

override fun onPause() {

super.onPause()

// пишем нужное в SharedPreferences

val ed = this.activity!!.getSharedPreferences(NAME\_SETTINGS, MODE\_PRIVATE).edit()

ed.putBoolean("Toast", switchToast.isChecked())

ed.putBoolean("Exit", switchExit.isChecked())

ed.commit()

}

}

**ThemeColors**

**Класс, который отвечает за цветовую гамму.**

**package com.extrime.my\_criptographer.ui.Settings**

import android.content.Context

import android.os.Build

import android.content.Intent

import android.graphics.Color

import android.util.Log

import com.extrime.my\_criptographer.R

import com.extrime.my\_criptographer.StartActivity

import com.extrime.my\_criptographer.StartActivity.Companion.TAG

import com.extrime.my\_criptographer.StartActivity.Companion.NAME\_SETTINGS

class ThemeColors(context: Context) {

var color: Int

// Checking if title text color will be black

private val isLightActionBar: Boolean

get() { // Checking if title text color will be black

val rgb = (Color.red(color) + Color.green(color) + Color.blue(color)) / 3

return rgb > 210

}

companion object {

fun setNewThemeColor(activity: StartActivity, red: Int, green: Int, blue: Int) {

var red = red

var green = green

var blue = blue

val colorStep = 15

red = Math.round((red / colorStep).toFloat()) \* colorStep

green = Math.round((green / colorStep).toFloat()) \* colorStep

blue = Math.round((blue / colorStep).toFloat()) \* colorStep

val stringColor = Integer.toHexString(Color.rgb(red, green, blue)).substring(2)

val editor = activity.getSharedPreferences(NAME\_SETTINGS, Context.MODE\_PRIVATE).edit()

editor.putString("color", stringColor)

editor.apply()

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.HONEYCOMB) activity.recreate()

else {

val i = activity.packageManager.getLaunchIntentForPackage(activity.packageName)

i!!.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP)

activity.startActivity(i)

}

}

}

init {

val sharedPreferences = context.getSharedPreferences(NAME\_SETTINGS, Context.MODE\_PRIVATE)

val stringColor = sharedPreferences.getString("color", "B64D4D")

color = Color.parseColor("#$stringColor")

if (isLightActionBar) context.setTheme(R.style.AppTheme)

context.setTheme(

context.resources.getIdentifier

("T\_$stringColor", "style", context.packageName)

)

}

}

**Layout**

**aboutprogram\_activity –**

**Разметка для отображения информации о приложении.**

<?xml version="1.0" encoding="utf-8"?>

<ScrollView

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:tools="http://schemas.android.com/tools"

android:background="@drawable/background"

android:padding="25dp"

tools:context=".AboutProgramFragment">

<TextView

android:text="@string/about\_program"

android:textSize="20sp"

android:layout\_gravity="top"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/tvAboutProgram"/>

</ScrollView>

**aes\_activity**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Ciphers.AES">

<EditText

android:layout\_width="155dp"

android:layout\_height="89dp"

android:text=""

android:hint="@string/Введите\_текст"

android:layout\_y="40dp"

android:layout\_x="122dp"

android:id="@+id/input\_text"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.2"

app:layout\_constraintHorizontal\_bias="0.5"/>

<EditText

android:layout\_width="159dp"

android:layout\_height="62dp"

android:text=""

android:hint="@string/Введите\_Пароль"

android:layout\_y="144dp"

android:layout\_x="122dp"

android:id="@+id/password"

app:layout\_constraintEnd\_toEndOf="@+id/input\_text"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"

android:layout\_marginTop="23dp"

app:layout\_constraintTop\_toBottomOf="@+id/input\_text"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintVertical\_bias="0.0"/>

<RadioGroup

android:id="@+id/radioGroup"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_y="220dp"

android:layout\_x="0dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/password"

app:layout\_constraintHorizontal\_bias="0.0"

>

<RadioButton

android:id="@+id/radio\_128"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="128"

android:checked="true"

android:layout\_marginLeft="50dp"

/>

<RadioButton

android:id="@+id/radio\_192"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="192"

android:layout\_marginLeft="50dp"

/>

<RadioButton

android:id="@+id/radio\_256"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="256"

android:layout\_marginLeft="50dp"

/>

</RadioGroup>

<TextView

android:layout\_width="151dp"

android:layout\_height="92dp"

android:text=""

android:gravity="center"

android:hint="@string/Ваш\_шифр"

android:layout\_y="285dp"

android:layout\_x="123dp"

android:id="@+id/secret\_text"

app:layout\_constraintStart\_toStartOf="@+id/radioGroup"

app:layout\_constraintEnd\_toEndOf="@+id/radioGroup"

app:layout\_constraintTop\_toBottomOf="@+id/radioGroup"

android:layout\_marginTop="16dp"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="@string/encrypt"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="390dp"

android:layout\_x="50dp"

android:id="@+id/BtnEncrypt"

app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"

android:layout\_marginTop="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.15"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="@string/decrypt"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="390dp"

android:layout\_x="230dp"

android:id="@+id/BtnDecrypt"

app:layout\_constraintStart\_toEndOf="@+id/BtnEncrypt"

android:layout\_marginStart="84dp"

app:layout\_constraintTop\_toTopOf="@+id/BtnEncrypt"

app:layout\_constraintBottom\_toBottomOf="@+id/BtnEncrypt"

app:layout\_constraintEnd\_toEndOf="@+id/radioGroup"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Скопировать"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="460dp"

android:layout\_x="50dp"

android:onClick="MainClick"

android:id="@+id/BtnCopyText"

tools:ignore="OnClick"

app:layout\_constraintStart\_toStartOf="@+id/BtnEncrypt"

app:layout\_constraintEnd\_toEndOf="@+id/BtnEncrypt"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/BtnEncrypt"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.097"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Очистить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="530dp"

android:layout\_x="140dp"

android:onClick="MainClick"

android:id="@+id/BtnClearText"

tools:ignore="OnClick"

app:layout\_constraintEnd\_toEndOf="@+id/BtnPasteText"

app:layout\_constraintStart\_toStartOf="@+id/BtnCopyText"

app:layout\_constraintTop\_toTopOf="@+id/BtnCopyText"

android:layout\_marginTop="56dp"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.0"/>

<ImageButton

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="50dp"

android:layout\_height="50dp"

app:srcCompat="@android:drawable/ic\_menu\_info\_details"

android:layout\_y="40dp"

android:layout\_x="330dp"

android:id="@+id/BtnInfo"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"

android:layout\_marginStart="176dp"

app:layout\_constraintBottom\_toBottomOf="@+id/input\_text"

android:layout\_marginBottom="88dp"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Вставить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="460dp"

android:layout\_x="230dp"

android:onClick="MainClick"

android:id="@+id/BtnPasteText"

tools:ignore="OnClick"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.1"

app:layout\_constraintEnd\_toEndOf="@+id/BtnDecrypt"

app:layout\_constraintStart\_toStartOf="@+id/BtnDecrypt"

app:layout\_constraintTop\_toBottomOf="@+id/BtnDecrypt"

app:layout\_constraintBottom\_toBottomOf="parent"

/>

</androidx.constraintlayout.widget.ConstraintLayout>

**algorithms\_activity**

**Разметка для отображения алгоритмов.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Algorithms.AlgorithmsFragment">

<GridLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="50dp"

android:rowCount="3"

android:columnCount="1"

android:layout\_alignParentStart="true"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="0"

android:layout\_column="0"

android:layout\_gravity="fill"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnCaesar"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/red"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="Caesar"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="1"

android:layout\_column="0"

android:layout\_gravity="fill"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnEXT"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/red"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="EXT"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="2"

android:layout\_column="0"

android:layout\_gravity="fill"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnSwitchChar"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/red"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="SwitchChar"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

</GridLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**app\_bar\_main**

**Разметка для отображения панели приложения.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.coordinatorlayout.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<com.google.android.material.appbar.AppBarLayout

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:theme="@style/AppTheme.AppBarOverlay">

<androidx.appcompat.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary"

app:popupTheme="@style/AppTheme.PopupOverlay"/>

</com.google.android.material.appbar.AppBarLayout>

<include layout="@layout/content\_main"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

**blowfish\_and\_des\_activity**

**Разметка для отображения шифров BlowFish и DES.**

?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background">

<ImageButton

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="50dp"

android:layout\_height="50dp"

app:srcCompat="@android:drawable/ic\_menu\_info\_details"

android:id="@+id/BtnInfo"

app:layout\_constraintBottom\_toBottomOf="@+id/input\_text"

android:layout\_marginBottom="96dp"

app:layout\_constraintStart\_toEndOf="@+id/input\_text"

android:layout\_marginStart="20dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintHorizontal\_bias="0.768"

app:layout\_constraintVertical\_bias="1.0"/>

<EditText

android:layout\_width="160dp"

android:layout\_height="89dp"

android:hint="@string/Введите\_текст"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.446"

android:id="@+id/input\_text"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.2"/>

<EditText

android:layout\_width="160dp"

android:layout\_height="99dp"

android:hint="@string/Введите\_Пароль"

app:layout\_constraintTop\_toBottomOf="@+id/input\_text"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"

android:layout\_marginTop="15dp"

app:layout\_constraintEnd\_toEndOf="@+id/input\_text"

app:layout\_constraintHorizontal\_bias="0.0"

android:id="@+id/password"/>

<TextView

android:layout\_width="160dp"

android:layout\_height="95dp"

android:gravity="center"

android:hint="@string/Ваш\_шифр"

app:layout\_constraintTop\_toBottomOf="@+id/password"

app:layout\_constraintStart\_toStartOf="@+id/password"

app:layout\_constraintEnd\_toEndOf="@+id/password"

android:layout\_marginTop="15dp"

app:layout\_constraintHorizontal\_bias="0.0"

android:id="@+id/secret\_text"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="@string/encrypt"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="40dp"

android:id="@+id/BtnEncrypt"

app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"

android:layout\_marginTop="20dp"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginLeft="64dp"

android:layout\_marginStart="64dp"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="@string/decrypt"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="40dp"

android:id="@+id/BtnDecrypt"

app:layout\_constraintEnd\_toEndOf="parent"

android:layout\_marginEnd="64dp"

android:layout\_marginRight="64dp"

app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"

android:layout\_marginTop="20dp"

app:layout\_constraintStart\_toEndOf="@+id/BtnEncrypt"

app:layout\_constraintHorizontal\_bias="0.93"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Скопировать"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="40dp"

android:onClick="MainClick"

android:id="@+id/BtnCopyText"

app:layout\_constraintStart\_toStartOf="@+id/BtnEncrypt"

app:layout\_constraintEnd\_toEndOf="@+id/BtnEncrypt"

app:layout\_constraintTop\_toBottomOf="@+id/BtnEncrypt"

android:layout\_marginTop="20dp"

app:layout\_constraintHorizontal\_bias="0.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Вставить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="40dp"

android:onClick="MainClick"

android:id="@+id/BtnPasteText"

app:layout\_constraintEnd\_toEndOf="@+id/BtnDecrypt"

app:layout\_constraintStart\_toStartOf="@+id/BtnDecrypt"

app:layout\_constraintBottom\_toBottomOf="@+id/BtnCopyText"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Очистить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="40dp"

android:onClick="MainClick"

android:id="@+id/BtnClearText"

app:layout\_constraintStart\_toStartOf="@+id/BtnCopyText"

app:layout\_constraintEnd\_toEndOf="@+id/BtnPasteText"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/BtnCopyText"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintVertical\_bias="0.1"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**caesar\_activity**

**Разметка для отображения алгоритма Цезаря.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Algorithms.Caesar">

<EditText

android:layout\_width="250dp"

android:layout\_height="150dp"

android:inputType="text"

android:hint="@string/Введите\_текст"

android:id="@+id/input\_text"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintVertical\_bias="0.25"/>

<EditText

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:id="@+id/password"

android:hint="Введите длину шага(не >16)"

app:layout\_constraintTop\_toBottomOf="@+id/input\_text"

android:layout\_marginTop="16dp"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"/>

<TextView

android:layout\_width="261dp"

android:layout\_height="67dp"

android:layout\_y="339dp"

android:layout\_x="74dp"

android:id="@+id/secret\_text"

android:gravity="center"

app:layout\_constraintTop\_toBottomOf="@+id/password"

android:layout\_marginTop="8dp"

app:layout\_constraintStart\_toStartOf="@+id/password"

app:layout\_constraintBottom\_toBottomOf="@+id/BtnDecrypt"

app:layout\_constraintVertical\_bias="0.152"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="В\nл\nе\nв\nо"

android:layout\_width="45dp"

android:layout\_height="370dp"

android:id="@+id/BtnEncrypt"

app:layout\_constraintTop\_toTopOf="@+id/input\_text"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginLeft="16dp"

android:layout\_marginStart="16dp"

app:layout\_constraintEnd\_toStartOf="@+id/input\_text"

android:layout\_marginTop="4dp"

app:layout\_constraintHorizontal\_bias="0.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="В\nп\nр\nа\nв\nо"

android:layout\_width="45dp"

android:layout\_height="370dp"

android:id="@+id/BtnDecrypt"

app:layout\_constraintTop\_toTopOf="@+id/input\_text"

app:layout\_constraintStart\_toEndOf="@+id/input\_text"

android:layout\_marginStart="16dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

android:layout\_marginTop="4dp"

android:layout\_marginLeft="16dp"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Очистить"

android:textAllCaps="false"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:onClick="MainClick"

android:id="@+id/BtnClearText"

android:layout\_y="613dp"

android:layout\_x="155dp"

app:layout\_constraintStart\_toStartOf="@+id/BtnCopyText"

app:layout\_constraintTop\_toBottomOf="@+id/BtnCopyText"

android:layout\_marginStart="72dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.179"

app:layout\_constraintHorizontal\_bias="0.038"

tools:ignore="OnClick"

app:layout\_constraintEnd\_toEndOf="@+id/BtnPasteText"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Копировать"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:onClick="MainClick"

android:id="@+id/BtnCopyText"

app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"

android:layout\_marginTop="20dp"

app:layout\_constraintStart\_toStartOf="parent"

android:layout\_marginLeft="80dp"

android:layout\_marginStart="80dp"

tools:ignore="OnClick"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Вставить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:onClick="MainClick"

android:id="@+id/BtnPasteText"

app:layout\_constraintTop\_toTopOf="@+id/BtnCopyText"

app:layout\_constraintEnd\_toStartOf="@+id/BtnDecrypt"

android:layout\_marginEnd="16dp"

android:layout\_marginRight="16dp"

tools:ignore="OnClick"/>

<ImageButton

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="50dp"

android:layout\_height="50dp"

app:srcCompat="@android:drawable/ic\_menu\_info\_details"

app:layout\_constraintBottom\_toTopOf="@+id/input\_text"

android:layout\_marginBottom="20dp"

app:layout\_constraintStart\_toEndOf="@+id/input\_text"

android:layout\_marginLeft="8dp"

android:layout\_marginStart="8dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="1.0"

android:id="@+id/BtnInfo"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**Разметка для отображения алгоритма RSA.**

**<?xml version="1.0" encoding="utf-8"?>**

**<androidx.constraintlayout.widget.ConstraintLayout**

**xmlns:android="http://schemas.android.com/apk/res/android"**

**xmlns:app="http://schemas.android.com/apk/res-auto"**

**xmlns:tools="http://schemas.android.com/tools"**

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**android:background="@drawable/background"**

**tools:context=".ui.Ciphers.RSA">**

**<ImageButton**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:layout\_width="50dp"**

**android:layout\_height="50dp"**

**app:srcCompat="@android:drawable/ic\_menu\_info\_details"**

**android:onClick="RSA\_OnClick"**

**android:id="@+id/BtnInfo"**

**app:layout\_constraintBottom\_toBottomOf="@+id/input\_text"**

**android:layout\_marginBottom="96dp"**

**app:layout\_constraintStart\_toEndOf="@+id/input\_text"**

**android:layout\_marginStart="20dp"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.768"**

**app:layout\_constraintVertical\_bias="1.0"/>**

**<EditText**

**android:layout\_width="160dp"**

**android:layout\_height="89dp"**

**android:hint="@string/Введите\_текст"**

**app:layout\_constraintTop\_toTopOf="parent"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.446"**

**android:id="@+id/input\_text"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintVertical\_bias="0.2"/>**

**<EditText**

**android:layout\_width="160dp"**

**android:layout\_height="99dp"**

**android:hint="@string/Введите\_Пароль"**

**app:layout\_constraintTop\_toBottomOf="@+id/input\_text"**

**app:layout\_constraintStart\_toStartOf="@+id/input\_text"**

**android:layout\_marginTop="15dp"**

**app:layout\_constraintEnd\_toEndOf="@+id/input\_text"**

**app:layout\_constraintHorizontal\_bias="0.0"**

**android:id="@+id/password"/>**

**<TextView**

**android:layout\_width="160dp"**

**android:layout\_height="95dp"**

**android:gravity="center"**

**android:hint="@string/Ваш\_шифр"**

**app:layout\_constraintTop\_toBottomOf="@+id/password"**

**app:layout\_constraintStart\_toStartOf="@+id/password"**

**app:layout\_constraintEnd\_toEndOf="@+id/password"**

**android:layout\_marginTop="15dp"**

**app:layout\_constraintHorizontal\_bias="0.0"**

**android:id="@+id/secret\_text"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="@string/encrypt"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="RSA\_OnClick"**

**android:id="@+id/BtnEncrypt"**

**app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"**

**android:layout\_marginTop="20dp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**android:layout\_marginLeft="64dp"**

**android:layout\_marginStart="64dp"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="@string/decrypt"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="RSA\_OnClick"**

**android:id="@+id/BtnDecrypt"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**android:layout\_marginEnd="64dp"**

**android:layout\_marginRight="64dp"**

**app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"**

**android:layout\_marginTop="20dp"**

**app:layout\_constraintStart\_toEndOf="@+id/BtnEncrypt"**

**app:layout\_constraintHorizontal\_bias="0.93"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="Скопировать"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="MainClick"**

**tools:ignore="OnClick"**

**android:id="@+id/BtnCopyText"**

**app:layout\_constraintStart\_toStartOf="@+id/BtnEncrypt"**

**app:layout\_constraintEnd\_toEndOf="@+id/BtnEncrypt"**

**app:layout\_constraintTop\_toBottomOf="@+id/BtnEncrypt"**

**android:layout\_marginTop="10dp"**

**app:layout\_constraintHorizontal\_bias="0.0"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="Вставить"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="MainClick"**

**tools:ignore="OnClick"**

**android:id="@+id/BtnPasteText"**

**app:layout\_constraintEnd\_toEndOf="@+id/BtnDecrypt"**

**app:layout\_constraintStart\_toStartOf="@+id/BtnDecrypt"**

**app:layout\_constraintBottom\_toBottomOf="@+id/BtnCopyText"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="Публичный"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="RSA\_OnClick"**

**android:id="@+id/BtnPublicKey"**

**app:layout\_constraintStart\_toStartOf="@+id/BtnCopyText"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**android:layout\_marginTop="10dp"**

**app:layout\_constraintTop\_toBottomOf="@+id/BtnCopyText"**

**app:layout\_constraintEnd\_toEndOf="@+id/BtnCopyText"**

**app:layout\_constraintVertical\_bias="0.0"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="Приватный"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="RSA\_OnClick"**

**android:id="@+id/BtnPrivateKey"**

**app:layout\_constraintStart\_toStartOf="@+id/BtnPasteText"**

**app:layout\_constraintEnd\_toEndOf="@+id/BtnPasteText"**

**app:layout\_constraintBottom\_toBottomOf="@+id/BtnPublicKey"**

**app:layout\_constraintTop\_toTopOf="@+id/BtnPublicKey"/>**

**<Button**

**style="@style/Widget.AppCompat.Button.Colored"**

**android:text="Очистить"**

**android:textAllCaps="false"**

**android:layout\_width="120dp"**

**android:layout\_height="40dp"**

**android:onClick="MainClick"**

**tools:ignore="OnClick"**

**android:id="@+id/BtnClearText"**

**app:layout\_constraintTop\_toBottomOf="@+id/BtnPrivateKey"**

**android:layout\_marginTop="10dp"**

**app:layout\_constraintEnd\_toEndOf="@+id/BtnPrivateKey"**

**app:layout\_constraintStart\_toStartOf="@+id/BtnPublicKey"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.5"**

**app:layout\_constraintVertical\_bias="0.0"/>**

**</androidx.constraintlayout.widget.ConstraintLayout>**

**ciphers\_activity**

**Разметка для отображения шифров.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Ciphers.CiphersFragment">

<GridLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_margin="50dp"

android:rowCount="3"

android:columnCount="1"

android:layout\_alignParentStart="true"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent">

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="0"

android:layout\_column="0"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnAES"

android:layout\_gravity="fill"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/green"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="AES"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="1"

android:layout\_column="0"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnBlowFish"

android:layout\_gravity="fill"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/green"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="BlowFish"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_row="2"

android:layout\_column="0"

android:layout\_gravity="fill"

android:onClick="MainClick"

tools:ignore="OnClick"

android:id="@+id/btnDES"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_margin="8dp"

app:cardCornerRadius="8dp"

app:cardElevation="8dp">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:layout\_gravity="center\_vertical|center\_horizontal"

android:gravity="center">

<ImageView

android:src="@drawable/green"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAlignment="center"

android:textStyle="bold"

android:text="DES"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

</GridLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**content\_main**

**Связывающая разметка всех фрагментов. Также устанавливает начальный фрагмент.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<fragment

android:id="@+id/nav\_host\_fragment"

android:name="androidx.navigation.fragment.NavHostFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:defaultNavHost="true"

app:navGraph="@navigation/mobile\_navigation"

/>

</androidx.constraintlayout.widget.ConstraintLayout>

**ext\_activity**

**Разметка для отображения алгоритма eXT.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Algorithms.eXT">

<ImageButton

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="50dp"

android:layout\_height="50dp"

app:srcCompat="@android:drawable/ic\_menu\_info\_details"

android:id="@+id/BtnInfo"

app:layout\_constraintBottom\_toBottomOf="@+id/input\_text"

android:layout\_marginBottom="108dp"

app:layout\_constraintStart\_toEndOf="@+id/input\_text"

android:layout\_marginStart="32dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintHorizontal\_bias="0.666"

app:layout\_constraintVertical\_bias="1.0"/>

<EditText

android:layout\_width="150dp"

android:layout\_height="92dp"

android:hint="@string/Введите\_текст"

android:layout\_y="25dp"

android:layout\_x="120dp"

android:id="@+id/input\_text"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.25"

app:layout\_constraintHorizontal\_bias="0.5"/>

<EditText

android:layout\_width="155dp"

android:layout\_height="wrap\_content"

android:text=""

android:inputType="number"

android:hint="password>999"

android:layout\_y="240dp"

android:layout\_x="120dp"

android:id="@+id/password"

app:layout\_constraintStart\_toStartOf="@+id/question"

app:layout\_constraintEnd\_toEndOf="@+id/question"

app:layout\_constraintTop\_toBottomOf="@+id/question"

android:layout\_marginTop="8dp"

app:layout\_constraintHorizontal\_bias="0.0"/>

<TextView

android:layout\_width="155dp"

android:layout\_height="56dp"

android:text=""

android:hint="Метод четный или нечетный?"

android:gravity="center"

android:layout\_y="170dp"

android:layout\_x="120dp"

android:id="@+id/question"

app:layout\_constraintTop\_toBottomOf="@+id/input\_text"

android:layout\_marginTop="16dp"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"

app:layout\_constraintEnd\_toEndOf="@+id/input\_text"

app:layout\_constraintHorizontal\_bias="0.0"/>

<RadioGroup

android:id="@+id/radioGroup1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_y="180dp"

android:layout\_x="0dp"

app:layout\_constraintTop\_toTopOf="@+id/question"

app:layout\_constraintBottom\_toBottomOf="@+id/question"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent">

<RadioButton

android:id="@+id/radio\_3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Четный"

android:checked="true"

android:layout\_marginLeft="20dp"/>

<RadioButton

android:id="@+id/radio\_4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Нечетный"

android:layout\_marginLeft="180dp"

/>

</RadioGroup>

<TextView

android:layout\_width="155dp"

android:layout\_height="56dp"

android:hint="key"

android:gravity="center"

android:layout\_y="300dp"

android:layout\_x="120dp"

android:id="@+id/key"

app:layout\_constraintTop\_toBottomOf="@+id/password"

android:layout\_marginTop="24dp"

app:layout\_constraintEnd\_toEndOf="@+id/password"

app:layout\_constraintStart\_toStartOf="@+id/password"

app:layout\_constraintHorizontal\_bias="0.0"/>

<RadioGroup

android:id="@+id/rg\_roots"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

android:layout\_y="310dp"

android:layout\_x="0dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintTop\_toTopOf="@+id/key"

app:layout\_constraintBottom\_toBottomOf="@+id/key" app:layout\_constraintVertical\_bias="0.0">

<RadioButton

android:id="@+id/radio\_1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="1 "

android:checked="true"

android:layout\_marginLeft="20dp"

/>

<RadioButton

android:id="@+id/radio\_2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="2"

android:layout\_marginLeft="220dp"

/>

</RadioGroup>

<ScrollView

android:layout\_width="155dp"

android:layout\_height="70dp"

android:layout\_y="350dp"

android:layout\_x="120dp"

android:id="@+id/scrollView3"

app:layout\_constraintTop\_toBottomOf="@+id/password"

android:layout\_marginTop="92dp"

app:layout\_constraintEnd\_toEndOf="@+id/password"

app:layout\_constraintStart\_toStartOf="@+id/password"

app:layout\_constraintHorizontal\_bias="0.0">

<TextView

android:layout\_width="155dp"

android:layout\_height="70dp"

android:gravity="center"

android:hint="@string/Ваш\_шифр"

android:layout\_y="350dp"

android:layout\_x="120dp"

android:id="@+id/secret\_text"/>

</ScrollView>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Дальше"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="550dp"

android:layout\_x="225dp"

android:id="@+id/BtnEncrypt"

app:layout\_constraintBottom\_toBottomOf="parent"

android:layout\_marginBottom="16dp"

app:layout\_constraintEnd\_toEndOf="@+id/BtnPasteText"

app:layout\_constraintStart\_toStartOf="@+id/BtnPasteText"

app:layout\_constraintTop\_toTopOf="@+id/BtnClearText"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Скопировать"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="480dp"

android:layout\_x="30dp"

android:onClick="MainClick"

android:id="@+id/BtnCopyText"

app:layout\_constraintTop\_toTopOf="@+id/scrollView3"

android:layout\_marginTop="80dp"

app:layout\_constraintEnd\_toStartOf="@+id/BtnPasteText"

android:layout\_marginEnd="52dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Вставить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="480dp"

android:layout\_x="225dp"

android:onClick="MainClick"

android:id="@+id/BtnPasteText"

app:layout\_constraintEnd\_toEndOf="parent"

android:layout\_marginEnd="52dp"

app:layout\_constraintBottom\_toBottomOf="@+id/BtnCopyText"

app:layout\_constraintTop\_toTopOf="@+id/scrollView3"

app:layout\_constraintVertical\_bias="1.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:text="Очистить"

android:textAllCaps="false"

android:layout\_width="120dp"

android:layout\_height="wrap\_content"

android:layout\_y="550dp"

android:layout\_x="30dp"

android:id="@+id/BtnClearText"

app:layout\_constraintBottom\_toBottomOf="parent"

android:layout\_marginBottom="16dp"

app:layout\_constraintStart\_toStartOf="@+id/BtnCopyText"

app:layout\_constraintEnd\_toEndOf="@+id/BtnCopyText"

app:layout\_constraintTop\_toBottomOf="@+id/BtnCopyText"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.1"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**information\_activity**

**Разметка для отображения информации алгоритмов и шифров.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingTop="100dp"

android:paddingLeft="20dp"

android:paddingRight="20dp"

android:paddingBottom="75dp"

android:background="@drawable/background"

tools:context=".Information">

<TextView

android:text=""

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/about"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintVertical\_bias="0.3"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**nav\_header\_main**

**Разметка в меню. Показывает ссылку на группу VK.**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="@dimen/nav\_header\_height"

android:background="@drawable/side\_nav\_bar"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:theme="@style/ThemeOverlay.AppCompat.Dark"

android:orientation="vertical"

android:gravity="bottom">

<ImageView

android:layout\_width="150dp"

android:layout\_height="150dp"

android:paddingTop="0dp"

android:onClick="MainClick"

app:srcCompat="@drawable/vk\_ava\_"

android:contentDescription="@string/nav\_header\_desc"

android:id="@+id/go\_vk1"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="MainClick"

android:paddingTop="@dimen/nav\_header\_vertical\_spacing"

android:text="@string/vk\_name\_group"

android:textAppearance="@style/TextAppearance.AppCompat.Body1"

android:id="@+id/go\_vk2"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="MainClick"

android:text="@string/vk\_link"

android:id="@+id/go\_vk3"/>

</LinearLayout>

**settings\_activity**

**Разметка для отображения настроек.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

android:background="@drawable/background"

android:padding="20dp"

tools:context=".ui.Settings.SettingsFragment">

<Switch

android:id="@+id/setToast"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:text="Показывать Toast's ?"

android:padding="10dp"

tools:ignore="UseSwitchCompatOrMaterialXml"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintVertical\_bias="0.1"

android:checked="true"/>

<Switch

android:id="@+id/setExit"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:text="Выходить из приложения?"

android:padding="10dp"

tools:ignore="UseSwitchCompatOrMaterialXml"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

android:checked="false"

app:layout\_constraintTop\_toBottomOf="@+id/setToast"

app:layout\_constraintVertical\_bias="0"/>

<Button

android:id="@+id/setTheme"

android:onClick="changeTheme"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:text="Сменить цвет"

android:textAllCaps="false"

android:scrollbarStyle="insideOverlay"

style="@style/Widget.AppCompat.Button.Colored"

tools:ignore="OnClick"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/setExit"

app:layout\_constraintVertical\_bias="0"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**start\_activity**

**Разметка, отражающая всё приложение.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.drawerlayout.widget.DrawerLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fitsSystemWindows="true"

tools:openDrawer="start">

<include

layout="@layout/app\_bar\_main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

<com.google.android.material.navigation.NavigationView

android:id="@+id/nav\_view"

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:background="@drawable/background"

android:fitsSystemWindows="true"

app:headerLayout="@layout/nav\_header\_main"

app:menu="@menu/activity\_main\_drawer"/>

</androidx.drawerlayout.widget.DrawerLayout>

**switch\_char\_activity**

**Разметка для отображения алгоритма смены символов.**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/background"

tools:context=".ui.Algorithms.SwitchChar">

<EditText

android:id="@+id/input\_text"

android:layout\_width="match\_parent"

android:layout\_height="100dp"

android:lines="10"

android:hint="@string/Введите\_текст"

android:ems="10"

android:layout\_y="100dp"

android:layout\_x="0dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintVertical\_bias="0.29">

<requestFocus/>

</EditText>

<EditText

android:id="@+id/password"

android:layout\_width="250dp"

android:layout\_height="46dp"

android:layout\_alignParentStart="true"

android:layout\_below="@+id/input\_text"

android:hint="какой символ заменить"

android:ems="10"

android:width="40dp"

android:layout\_y="208dp"

android:layout\_x="15dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/input\_text"

app:layout\_constraintVertical\_bias="0.05"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.503"/>

<EditText

android:id="@+id/secret\_text"

android:layout\_width="250dp"

android:layout\_height="51dp"

android:layout\_alignParentStart="true"

android:layout\_alignTop="@+id/password"

android:layout\_toEndOf="@+id/password"

android:width="40dp"

android:hint="на какой символ заменить"

android:ems="10"

android:layout\_y="264dp"

android:layout\_x="15dp"

app:layout\_constraintStart\_toStartOf="@+id/password"

app:layout\_constraintTop\_toBottomOf="@+id/password"

android:layout\_marginTop="12dp"

app:layout\_constraintEnd\_toEndOf="@+id/password"

app:layout\_constraintHorizontal\_bias="0.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/password"

android:layout\_toEndOf="@+id/secret\_text"

android:onClick="replace"

android:text="Заменить"

android:textAllCaps="false"

android:layout\_y="335dp"

android:layout\_x="20dp"

app:layout\_constraintEnd\_toEndOf="@+id/secret\_text"

app:layout\_constraintTop\_toTopOf="@+id/button"

app:layout\_constraintBottom\_toBottomOf="@+id/button"

app:layout\_constraintStart\_toEndOf="@+id/button"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintVertical\_bias="1.0"/>

<Button

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignTop="@+id/password"

android:layout\_toEndOf="@+id/secret\_text"

android:text="Очистить"

android:onClick="Clear\_Text"

android:textAllCaps="false"

android:layout\_y="335dp"

android:layout\_x="130dp"

app:layout\_constraintStart\_toStartOf="@+id/secret\_text"

app:layout\_constraintTop\_toBottomOf="@+id/secret\_text"

android:layout\_marginTop="28dp"

android:id="@+id/button"/>

<ImageButton

style="@style/Widget.AppCompat.Button.Colored"

android:layout\_width="50dp"

android:layout\_height="50dp"

app:srcCompat="@android:drawable/ic\_menu\_info\_details"

android:layout\_y="40dp"

android:layout\_x="330dp"

android:onClick="onClick"

app:layout\_constraintStart\_toStartOf="@+id/input\_text"

app:layout\_constraintTop\_toTopOf="parent"

android:layout\_marginTop="16dp"

app:layout\_constraintEnd\_toEndOf="@+id/input\_text"

app:layout\_constraintBottom\_toTopOf="@+id/input\_text"

app:layout\_constraintHorizontal\_bias="0.932"

app:layout\_constraintVertical\_bias="0.7"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**Класс, который кодирует и декодирует строки.**

package com.extrime.my\_criptographer;

import android.os.Build;

import androidx.annotation.RequiresApi;

// TODO: HotSpotIntrinsicCandidate

//import jdk.internal.HotSpotIntrinsicCandidate;

import java.io.FilterOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.nio.ByteBuffer;

import java.nio.charset.StandardCharsets;

import java.util.Arrays;

import java.util.Objects;

/\*\*

\* This class consists exclusively of static methods for obtaining

\* encoders and decoders for the Base64 encoding scheme. The

\* implementation of this class supports the following types of Base64

\* as specified in

\* <a href="http://www.ietf.org/rfc/rfc4648.txt">RFC 4648</a> and

\* <a href="http://www.ietf.org/rfc/rfc2045.txt">RFC 2045</a>.

\*

\* <ul>

\* <li><a id="basic"><b>Basic</b></a>

\* <p> Uses "The Base64 Alphabet" as specified in Table 1 of

\* RFC 4648 and RFC 2045 for encoding and decoding operation.

\* The encoder does not add any line feed (line separator)

\* character. The decoder rejects data that contains characters

\* outside the base64 alphabet.</p></li>

\*

\* <li><a id="url"><b>URL and Filename safe</b></a>

\* <p> Uses the "URL and Filename safe Base64 Alphabet" as specified

\* in Table 2 of RFC 4648 for encoding and decoding. The

\* encoder does not add any line feed (line separator) character.

\* The decoder rejects data that contains characters outside the

\* base64 alphabet.</p></li>

\*

\* <li><a id="mime"><b>MIME</b></a>

\* <p> Uses "The Base64 Alphabet" as specified in Table 1 of

\* RFC 2045 for encoding and decoding operation. The encoded output

\* must be represented in lines of no more than 76 characters each

\* and uses a carriage return {@code '\r'} followed immediately by

\* a linefeed {@code '\n'} as the line separator. No line separator

\* is added to the end of the encoded output. All line separators

\* or other characters not found in the base64 alphabet table are

\* ignored in decoding operation.</p></li>

\* </ul>

\*

\* <p> Unless otherwise noted, passing a {@code null} argument to a

\* method of this class will cause a {@link NullPointerException

\* NullPointerException} to be thrown.

\*

\* @author Xueming Shen

\* @since 1.8

\*/

public class Base64 {

private Base64() {}

/\*\*

\* Returns a {@link Encoder} that encodes using the

\* <a href="#basic">Basic</a> type base64 encoding scheme.

\*

\* @return A Base64 encoder.

\*/

public static Encoder getEncoder() {

return Encoder.RFC4648;

}

/\*\*

\* Returns a {@link Encoder} that encodes using the

\* <a href="#url">URL and Filename safe</a> type base64

\* encoding scheme.

\*

\* @return A Base64 encoder.

\*/

public static Encoder getUrlEncoder() {

return Encoder.RFC4648\_URLSAFE;

}

/\*\*

\* Returns a {@link Encoder} that encodes using the

\* <a href="#mime">MIME</a> type base64 encoding scheme.

\*

\* @return A Base64 encoder.

\*/

public static Encoder getMimeEncoder() {

return Encoder.RFC2045;

}

/\*\*

\* Returns a {@link Encoder} that encodes using the

\* <a href="#mime">MIME</a> type base64 encoding scheme

\* with specified line length and line separators.

\*

\* @param lineLength

\* the length of each output line (rounded down to nearest multiple

\* of 4). If the rounded down line length is not a positive value,

\* the output will not be separated in lines

\* @param lineSeparator

\* the line separator for each output line

\*

\* @return A Base64 encoder.

\*

\* @throws IllegalArgumentException if {@code lineSeparator} includes any

\* character of "The Base64 Alphabet" as specified in Table 1 of

\* RFC 2045.

\*/

public static Encoder getMimeEncoder(int lineLength, byte[] lineSeparator) {

Objects.requireNonNull(lineSeparator);

int[] base64 = Decoder.fromBase64;

for (byte b : lineSeparator) {

if (base64[b & 0xff] != -1)

throw new IllegalArgumentException(

"Illegal base64 line separator character 0x" + Integer.toString(b, 16));

}

// round down to nearest multiple of 4

lineLength &= ~0b11;

if (lineLength <= 0) {

return Encoder.RFC4648;

}

return new Encoder(false, lineSeparator, lineLength, true);

}

/\*\*

\* Returns a {@link Decoder} that decodes using the

\* <a href="#basic">Basic</a> type base64 encoding scheme.

\*

\* @return A Base64 decoder.

\*/

public static Decoder getDecoder() {

return Decoder.RFC4648;

}

/\*\*

\* Returns a {@link Decoder} that decodes using the

\* <a href="#url">URL and Filename safe</a> type base64

\* encoding scheme.

\*

\* @return A Base64 decoder.

\*/

public static Decoder getUrlDecoder() {

return Decoder.RFC4648\_URLSAFE;

}

/\*\*

\* Returns a {@link Decoder} that decodes using the

\* <a href="#mime">MIME</a> type base64 decoding scheme.

\*

\* @return A Base64 decoder.

\*/

public static Decoder getMimeDecoder() {

return Decoder.RFC2045;

}

/\*\*

\* This class implements an encoder for encoding byte data using

\* the Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

\*

\* <p> Instances of {@link Encoder} class are safe for use by

\* multiple concurrent threads.

\*

\* <p> Unless otherwise noted, passing a {@code null} argument to

\* a method of this class will cause a

\* {@link NullPointerException NullPointerException} to

\* be thrown.

\*

\* @see Decoder

\* @since 1.8

\*/

public static class Encoder {

private final byte[] newline;

private final int linemax;

private final boolean isURL;

private final boolean doPadding;

private Encoder(boolean isURL, byte[] newline, int linemax, boolean doPadding) {

this.isURL = isURL;

this.newline = newline;

this.linemax = linemax;

this.doPadding = doPadding;

}

/\*\*

\* This array is a lookup table that translates 6-bit positive integer

\* index values into their "Base64 Alphabet" equivalents as specified

\* in "Table 1: The Base64 Alphabet" of RFC 2045 (and RFC 4648).

\*/

private static final char[] toBase64 = {

'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',

'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '+', '/'

};

/\*\*

\* It's the lookup table for "URL and Filename safe Base64" as specified

\* in Table 2 of the RFC 4648, with the '+' and '/' changed to '-' and

\* '\_'. This table is used when BASE64\_URL is specified.

\*/

private static final char[] toBase64URL = {

'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',

'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '-', '\_'

};

private static final int MIMELINEMAX = 76;

private static final byte[] CRLF = new byte[] {'\r', '\n'};

static final Encoder RFC4648 = new Encoder(false, null, -1, true);

static final Encoder RFC4648\_URLSAFE = new Encoder(true, null, -1, true);

static final Encoder RFC2045 = new Encoder(false, CRLF, MIMELINEMAX, true);

private final int outLength(int srclen) {

int len = 0;

if (doPadding) {

len = 4 \* ((srclen + 2) / 3);

} else {

int n = srclen % 3;

len = 4 \* (srclen / 3) + (n == 0 ? 0 : n + 1);

}

if (linemax > 0) // line separators

len += (len - 1) / linemax \* newline.length;

return len;

}

/\*\*

\* Encodes all bytes from the specified byte array into a newly-allocated

\* byte array using the {@link Base64} encoding scheme. The returned byte

\* array is of the length of the resulting bytes.

\*

\* @param src

\* the byte array to encode

\* @return A newly-allocated byte array containing the resulting

\* encoded bytes.

\*/

public byte[] encode(byte[] src) {

int len = outLength(src.length); // dst array size

byte[] dst = new byte[len];

int ret = encode0(src, 0, src.length, dst);

if (ret != dst.length)

return Arrays.copyOf(dst, ret);

return dst;

}

/\*\*

\* Encodes all bytes from the specified byte array using the

\* {@link Base64} encoding scheme, writing the resulting bytes to the

\* given output byte array, starting at offset 0.

\*

\* <p> It is the responsibility of the invoker of this method to make

\* sure the output byte array {@code dst} has enough space for encoding

\* all bytes from the input byte array. No bytes will be written to the

\* output byte array if the output byte array is not big enough.

\*

\* @param src

\* the byte array to encode

\* @param dst

\* the output byte array

\* @return The number of bytes written to the output byte array

\*

\* @throws IllegalArgumentException if {@code dst} does not have enough

\* space for encoding all input bytes.

\*/

public int encode(byte[] src, byte[] dst) {

int len = outLength(src.length); // dst array size

if (dst.length < len)

throw new IllegalArgumentException(

"Output byte array is too small for encoding all input bytes");

return encode0(src, 0, src.length, dst);

}

/\*\*

\* Encodes the specified byte array into a String using the {@link Base64}

\* encoding scheme.

\*

\* <p> This method first encodes all input bytes into a base64 encoded

\* byte array and then constructs a new String by using the encoded byte

\* array and the {@link StandardCharsets#ISO\_8859\_1

\* ISO-8859-1} charset.

\*

\* <p> In other words, an invocation of this method has exactly the same

\* effect as invoking

\* {@code new String(encode(src), StandardCharsets.ISO\_8859\_1)}.

\*

\* @param src

\* the byte array to encode

\* @return A String containing the resulting Base64 encoded characters

\*/

@SuppressWarnings("deprecation")

public String encodeToString(byte[] src) {

byte[] encoded = encode(src);

return new String(encoded, 0, 0, encoded.length);

}

/\*\*

\* Encodes all remaining bytes from the specified byte buffer into

\* a newly-allocated ByteBuffer using the {@link Base64} encoding

\* scheme.

\*

\* Upon return, the source buffer's position will be updated to

\* its limit; its limit will not have been changed. The returned

\* output buffer's position will be zero and its limit will be the

\* number of resulting encoded bytes.

\*

\* @param buffer

\* the source ByteBuffer to encode

\* @return A newly-allocated byte buffer containing the encoded bytes.

\*/

public ByteBuffer encode(ByteBuffer buffer) {

int len = outLength(buffer.remaining());

byte[] dst = new byte[len];

int ret = 0;

if (buffer.hasArray()) {

ret = encode0(buffer.array(),

buffer.arrayOffset() + buffer.position(),

buffer.arrayOffset() + buffer.limit(),

dst);

buffer.position(buffer.limit());

} else {

byte[] src = new byte[buffer.remaining()];

buffer.get(src);

ret = encode0(src, 0, src.length, dst);

}

if (ret != dst.length)

dst = Arrays.copyOf(dst, ret);

return ByteBuffer.wrap(dst);

}

/\*\*

\* Wraps an output stream for encoding byte data using the {@link Base64}

\* encoding scheme.

\*

\* <p> It is recommended to promptly close the returned output stream after

\* use, during which it will flush all possible leftover bytes to the underlying

\* output stream. Closing the returned output stream will close the underlying

\* output stream.

\*

\* @param os

\* the output stream.

\* @return the output stream for encoding the byte data into the

\* specified Base64 encoded format

\*/

public OutputStream wrap(OutputStream os) {

Objects.requireNonNull(os);

return new EncOutputStream(os, isURL ? toBase64URL : toBase64,

newline, linemax, doPadding);

}

/\*\*

\* Returns an encoder instance that encodes equivalently to this one,

\* but without adding any padding character at the end of the encoded

\* byte data.

\*

\* <p> The encoding scheme of this encoder instance is unaffected by

\* this invocation. The returned encoder instance should be used for

\* non-padding encoding operation.

\*

\* @return an equivalent encoder that encodes without adding any

\* padding character at the end

\*/

public Encoder withoutPadding() {

if (!doPadding)

return this;

return new Encoder(isURL, newline, linemax, false);

}

// TODO: HotSpotIntrinsicCandidate

// @HotSpotIntrinsicCandidate

private void encodeBlock(byte[] src, int sp, int sl, byte[] dst, int dp, boolean isURL) {

char[] base64 = isURL ? toBase64URL : toBase64;

for (int sp0 = sp, dp0 = dp ; sp0 < sl; ) {

int bits = (src[sp0++] & 0xff) << 16 |

(src[sp0++] & 0xff) << 8 |

(src[sp0++] & 0xff);

dst[dp0++] = (byte)base64[(bits >>> 18) & 0x3f];

dst[dp0++] = (byte)base64[(bits >>> 12) & 0x3f];

dst[dp0++] = (byte)base64[(bits >>> 6) & 0x3f];

dst[dp0++] = (byte)base64[bits & 0x3f];

}

}

private int encode0(byte[] src, int off, int end, byte[] dst) {

char[] base64 = isURL ? toBase64URL : toBase64;

int sp = off;

int slen = (end - off) / 3 \* 3;

int sl = off + slen;

if (linemax > 0 && slen > linemax / 4 \* 3)

slen = linemax / 4 \* 3;

int dp = 0;

while (sp < sl) {

int sl0 = Math.min(sp + slen, sl);

encodeBlock(src, sp, sl0, dst, dp, isURL);

int dlen = (sl0 - sp) / 3 \* 4;

dp += dlen;

sp = sl0;

if (dlen == linemax && sp < end) {

for (byte b : newline){

dst[dp++] = b;

}

}

}

if (sp < end) { // 1 or 2 leftover bytes

int b0 = src[sp++] & 0xff;

dst[dp++] = (byte)base64[b0 >> 2];

if (sp == end) {

dst[dp++] = (byte)base64[(b0 << 4) & 0x3f];

if (doPadding) {

dst[dp++] = '=';

dst[dp++] = '=';

}

} else {

int b1 = src[sp++] & 0xff;

dst[dp++] = (byte)base64[(b0 << 4) & 0x3f | (b1 >> 4)];

dst[dp++] = (byte)base64[(b1 << 2) & 0x3f];

if (doPadding) {

dst[dp++] = '=';

}

}

}

return dp;

}

}

/\*\*

\* This class implements a decoder for decoding byte data using the

\* Base64 encoding scheme as specified in RFC 4648 and RFC 2045.

\*

\* <p> The Base64 padding character {@code '='} is accepted and

\* interpreted as the end of the encoded byte data, but is not

\* required. So if the final unit of the encoded byte data only has

\* two or three Base64 characters (without the corresponding padding

\* character(s) padded), they are decoded as if followed by padding

\* character(s). If there is a padding character present in the

\* final unit, the correct number of padding character(s) must be

\* present, otherwise {@code IllegalArgumentException} (

\* {@code IOException} when reading from a Base64 stream) is thrown

\* during decoding.

\*

\* <p> Instances of {@link Decoder} class are safe for use by

\* multiple concurrent threads.

\*

\* <p> Unless otherwise noted, passing a {@code null} argument to

\* a method of this class will cause a

\* {@link NullPointerException NullPointerException} to

\* be thrown.

\*

\* @see Encoder

\* @since 1.8

\*/

public static class Decoder {

private final boolean isURL;

private final boolean isMIME;

private Decoder(boolean isURL, boolean isMIME) {

this.isURL = isURL;

this.isMIME = isMIME;

}

/\*\*

\* Lookup table for decoding unicode characters drawn from the

\* "Base64 Alphabet" (as specified in Table 1 of RFC 2045) into

\* their 6-bit positive integer equivalents. Characters that

\* are not in the Base64 alphabet but fall within the bounds of

\* the array are encoded to -1.

\*

\*/

private static final int[] fromBase64 = new int[256];

static {

Arrays.fill(fromBase64, -1);

for (int i = 0; i < Encoder.toBase64.length; i++)

fromBase64[Encoder.toBase64[i]] = i;

fromBase64['='] = -2;

}

/\*\*

\* Lookup table for decoding "URL and Filename safe Base64 Alphabet"

\* as specified in Table2 of the RFC 4648.

\*/

private static final int[] fromBase64URL = new int[256];

static {

Arrays.fill(fromBase64URL, -1);

for (int i = 0; i < Encoder.toBase64URL.length; i++)

fromBase64URL[Encoder.toBase64URL[i]] = i;

fromBase64URL['='] = -2;

}

static final Decoder RFC4648 = new Decoder(false, false);

static final Decoder RFC4648\_URLSAFE = new Decoder(true, false);

static final Decoder RFC2045 = new Decoder(false, true);

/\*\*

\* Decodes all bytes from the input byte array using the {@link Base64}

\* encoding scheme, writing the results into a newly-allocated output

\* byte array. The returned byte array is of the length of the resulting

\* bytes.

\*

\* @param src

\* the byte array to decode

\*

\* @return A newly-allocated byte array containing the decoded bytes.

\*

\* @throws IllegalArgumentException

\* if {@code src} is not in valid Base64 scheme

\*/

public byte[] decode(byte[] src) {

byte[] dst = new byte[outLength(src, 0, src.length)];

int ret = decode0(src, 0, src.length, dst);

if (ret != dst.length) {

dst = Arrays.copyOf(dst, ret);

}

return dst;

}

/\*\*

\* Decodes a Base64 encoded String into a newly-allocated byte array

\* using the {@link Base64} encoding scheme.

\*

\* <p> An invocation of this method has exactly the same effect as invoking

\* {@code decode(src.getBytes(StandardCharsets.ISO\_8859\_1))}

\*

\* @param src

\* the string to decode

\*

\* @return A newly-allocated byte array containing the decoded bytes.

\*

\* @throws IllegalArgumentException

\* if {@code src} is not in valid Base64 scheme

\*/

// TODO: CHECK VERSION\_CODESKITKAT

@RequiresApi(api = Build.VERSION\_CODES.KITKAT)

public byte[] decode(String src) {

return decode(src.getBytes(StandardCharsets.ISO\_8859\_1));

}

/\*\*

\* Decodes all bytes from the input byte array using the {@link Base64}

\* encoding scheme, writing the results into the given output byte array,

\* starting at offset 0.

\*

\* <p> It is the responsibility of the invoker of this method to make

\* sure the output byte array {@code dst} has enough space for decoding

\* all bytes from the input byte array. No bytes will be written to

\* the output byte array if the output byte array is not big enough.

\*

\* <p> If the input byte array is not in valid Base64 encoding scheme

\* then some bytes may have been written to the output byte array before

\* IllegalargumentException is thrown.

\*

\* @param src

\* the byte array to decode

\* @param dst

\* the output byte array

\*

\* @return The number of bytes written to the output byte array

\*

\* @throws IllegalArgumentException

\* if {@code src} is not in valid Base64 scheme, or {@code dst}

\* does not have enough space for decoding all input bytes.

\*/

public int decode(byte[] src, byte[] dst) {

int len = outLength(src, 0, src.length);

if (dst.length < len)

throw new IllegalArgumentException(

"Output byte array is too small for decoding all input bytes");

return decode0(src, 0, src.length, dst);

}

/\*\*

\* Decodes all bytes from the input byte buffer using the {@link Base64}

\* encoding scheme, writing the results into a newly-allocated ByteBuffer.

\*

\* <p> Upon return, the source buffer's position will be updated to

\* its limit; its limit will not have been changed. The returned

\* output buffer's position will be zero and its limit will be the

\* number of resulting decoded bytes

\*

\* <p> {@code IllegalArgumentException} is thrown if the input buffer

\* is not in valid Base64 encoding scheme. The position of the input

\* buffer will not be advanced in this case.

\*

\* @param buffer

\* the ByteBuffer to decode

\*

\* @return A newly-allocated byte buffer containing the decoded bytes

\*

\* @throws IllegalArgumentException

\* if {@code src} is not in valid Base64 scheme.

\*/

public ByteBuffer decode(ByteBuffer buffer) {

int pos0 = buffer.position();

try {

byte[] src;

int sp, sl;

if (buffer.hasArray()) {

src = buffer.array();

sp = buffer.arrayOffset() + buffer.position();

sl = buffer.arrayOffset() + buffer.limit();

buffer.position(buffer.limit());

} else {

src = new byte[buffer.remaining()];

buffer.get(src);

sp = 0;

sl = src.length;

}

byte[] dst = new byte[outLength(src, sp, sl)];

return ByteBuffer.wrap(dst, 0, decode0(src, sp, sl, dst));

} catch (IllegalArgumentException iae) {

buffer.position(pos0);

throw iae;

}

}

/\*\*

\* Returns an input stream for decoding {@link Base64} encoded byte stream.

\*

\* <p> The {@code read} methods of the returned {@code InputStream} will

\* throw {@code IOException} when reading bytes that cannot be decoded.

\*

\* <p> Closing the returned input stream will close the underlying

\* input stream.

\*

\* @param is

\* the input stream

\*

\* @return the input stream for decoding the specified Base64 encoded

\* byte stream

\*/

public InputStream wrap(InputStream is) {

Objects.requireNonNull(is);

return new DecInputStream(is, isURL ? fromBase64URL : fromBase64, isMIME);

}

private int outLength(byte[] src, int sp, int sl) {

int[] base64 = isURL ? fromBase64URL : fromBase64;

int paddings = 0;

int len = sl - sp;

if (len == 0)

return 0;

if (len < 2) {

if (isMIME && base64[0] == -1)

return 0;

throw new IllegalArgumentException(

"Input byte[] should at least have 2 bytes for base64 bytes");

}

if (isMIME) {

// scan all bytes to fill out all non-alphabet. a performance

// trade-off of pre-scan or Arrays.copyOf

int n = 0;

while (sp < sl) {

int b = src[sp++] & 0xff;

if (b == '=') {

len -= (sl - sp + 1);

break;

}

if ((b = base64[b]) == -1)

n++;

}

len -= n;

} else {

if (src[sl - 1] == '=') {

paddings++;

if (src[sl - 2] == '=')

paddings++;

}

}

if (paddings == 0 && (len & 0x3) != 0)

paddings = 4 - (len & 0x3);

return 3 \* ((len + 3) / 4) - paddings;

}

private int decode0(byte[] src, int sp, int sl, byte[] dst) {

int[] base64 = isURL ? fromBase64URL : fromBase64;

int dp = 0;

int bits = 0;

int shiftto = 18; // pos of first byte of 4-byte atom

while (sp < sl) {

if (shiftto == 18 && sp + 4 < sl) { // fast path

int sl0 = sp + ((sl - sp) & ~0b11);

while (sp < sl0) {

int b1 = base64[src[sp++] & 0xff];

int b2 = base64[src[sp++] & 0xff];

int b3 = base64[src[sp++] & 0xff];

int b4 = base64[src[sp++] & 0xff];

if ((b1 | b2 | b3 | b4) < 0) { // non base64 byte

sp -= 4;

break;

}

int bits0 = b1 << 18 | b2 << 12 | b3 << 6 | b4;

dst[dp++] = (byte)(bits0 >> 16);

dst[dp++] = (byte)(bits0 >> 8);

dst[dp++] = (byte)(bits0);

}

if (sp >= sl)

break;

}

int b = src[sp++] & 0xff;

if ((b = base64[b]) < 0) {

if (b == -2) { // padding byte '='

// = shiftto==18 unnecessary padding

// x= shiftto==12 a dangling single x

// x to be handled together with non-padding case

// xx= shiftto==6&&sp==sl missing last =

// xx=y shiftto==6 last is not =

if (shiftto == 6 && (sp == sl || src[sp++] != '=') ||

shiftto == 18) {

throw new IllegalArgumentException(

"Input byte array has wrong 4-byte ending unit");

}

break;

}

if (isMIME) // skip if for rfc2045

continue;

else

throw new IllegalArgumentException(

"Illegal base64 character " +

Integer.toString(src[sp - 1], 16));

}

bits |= (b << shiftto);

shiftto -= 6;

if (shiftto < 0) {

dst[dp++] = (byte)(bits >> 16);

dst[dp++] = (byte)(bits >> 8);

dst[dp++] = (byte)(bits);

shiftto = 18;

bits = 0;

}

}

// reached end of byte array or hit padding '=' characters.

if (shiftto == 6) {

dst[dp++] = (byte)(bits >> 16);

} else if (shiftto == 0) {

dst[dp++] = (byte)(bits >> 16);

dst[dp++] = (byte)(bits >> 8);

} else if (shiftto == 12) {

// dangling single "x", incorrectly encoded.

throw new IllegalArgumentException(

"Last unit does not have enough valid bits");

}

// anything left is invalid, if is not MIME.

// if MIME, ignore all non-base64 character

while (sp < sl) {

if (isMIME && base64[src[sp++] & 0xff] < 0)

continue;

throw new IllegalArgumentException(

"Input byte array has incorrect ending byte at " + sp);

}

return dp;

}

}

/\*

\* An output stream for encoding bytes into the Base64.

\*/

private static class EncOutputStream extends FilterOutputStream {

private int leftover = 0;

private int b0, b1, b2;

private boolean closed = false;

private final char[] base64; // byte->base64 mapping

private final byte[] newline; // line separator, if needed

private final int linemax;

private final boolean doPadding;// whether or not to pad

private int linepos = 0;

private byte[] buf;

EncOutputStream(OutputStream os, char[] base64,

byte[] newline, int linemax, boolean doPadding) {

super(os);

this.base64 = base64;

this.newline = newline;

this.linemax = linemax;

this.doPadding = doPadding;

this.buf = new byte[linemax <= 0 ? 8124 : linemax];

}

@Override

public void write(int b) throws IOException {

byte[] buf = new byte[1];

buf[0] = (byte)(b & 0xff);

write(buf, 0, 1);

}

private void checkNewline() throws IOException {

if (linepos == linemax) {

out.write(newline);

linepos = 0;

}

}

private void writeb4(char b1, char b2, char b3, char b4) throws IOException {

buf[0] = (byte)b1;

buf[1] = (byte)b2;

buf[2] = (byte)b3;

buf[3] = (byte)b4;

out.write(buf, 0, 4);

}

@Override

public void write(byte[] b, int off, int len) throws IOException {

if (closed)

throw new IOException("Stream is closed");

if (off < 0 || len < 0 || len > b.length - off)

throw new ArrayIndexOutOfBoundsException();

if (len == 0)

return;

if (leftover != 0) {

if (leftover == 1) {

b1 = b[off++] & 0xff;

len--;

if (len == 0) {

leftover++;

return;

}

}

b2 = b[off++] & 0xff;

len--;

checkNewline();

writeb4(base64[b0 >> 2],

base64[(b0 << 4) & 0x3f | (b1 >> 4)],

base64[(b1 << 2) & 0x3f | (b2 >> 6)],

base64[b2 & 0x3f]);

linepos += 4;

}

int nBits24 = len / 3;

leftover = len - (nBits24 \* 3);

while (nBits24 > 0) {

checkNewline();

int dl = linemax <= 0 ? buf.length : buf.length - linepos;

int sl = off + Math.min(nBits24, dl / 4) \* 3;

int dp = 0;

for (int sp = off; sp < sl; ) {

int bits = (b[sp++] & 0xff) << 16 |

(b[sp++] & 0xff) << 8 |

(b[sp++] & 0xff);

buf[dp++] = (byte)base64[(bits >>> 18) & 0x3f];

buf[dp++] = (byte)base64[(bits >>> 12) & 0x3f];

buf[dp++] = (byte)base64[(bits >>> 6) & 0x3f];

buf[dp++] = (byte)base64[bits & 0x3f];

}

out.write(buf, 0, dp);

off = sl;

linepos += dp;

nBits24 -= dp / 4;

}

if (leftover == 1) {

b0 = b[off++] & 0xff;

} else if (leftover == 2) {

b0 = b[off++] & 0xff;

b1 = b[off++] & 0xff;

}

}

@Override

public void close() throws IOException {

if (!closed) {

closed = true;

if (leftover == 1) {

checkNewline();

out.write(base64[b0 >> 2]);

out.write(base64[(b0 << 4) & 0x3f]);

if (doPadding) {

out.write('=');

out.write('=');

}

} else if (leftover == 2) {

checkNewline();

out.write(base64[b0 >> 2]);

out.write(base64[(b0 << 4) & 0x3f | (b1 >> 4)]);

out.write(base64[(b1 << 2) & 0x3f]);

if (doPadding) {

out.write('=');

}

}

leftover = 0;

out.close();

}

}

}

/\*

\* An input stream for decoding Base64 bytes

\*/

private static class DecInputStream extends InputStream {

private final InputStream is;

private final boolean isMIME;

private final int[] base64; // base64 -> byte mapping

private int bits = 0; // 24-bit buffer for decoding

private int nextin = 18; // next available "off" in "bits" for input;

// -> 18, 12, 6, 0

private int nextout = -8; // next available "off" in "bits" for output;

// -> 8, 0, -8 (no byte for output)

private boolean eof = false;

private boolean closed = false;

DecInputStream(InputStream is, int[] base64, boolean isMIME) {

this.is = is;

this.base64 = base64;

this.isMIME = isMIME;

}

private byte[] sbBuf = new byte[1];

@Override

public int read() throws IOException {

return read(sbBuf, 0, 1) == -1 ? -1 : sbBuf[0] & 0xff;

}

private int eof(byte[] b, int off, int len, int oldOff)

throws IOException

{

eof = true;

if (nextin != 18) {

if (nextin == 12)

throw new IOException("Base64 stream has one un-decoded dangling byte.");

// treat ending xx/xxx without padding character legal.

// same logic as v == '=' below

b[off++] = (byte)(bits >> (16));

if (nextin == 0) { // only one padding byte

if (len == 1) { // no enough output space

bits >>= 8; // shift to lowest byte

nextout = 0;

} else {

b[off++] = (byte) (bits >> 8);

}

}

}

return off == oldOff ? -1 : off - oldOff;

}

private int padding(byte[] b, int off, int len, int oldOff)

throws IOException

{

// = shiftto==18 unnecessary padding

// x= shiftto==12 dangling x, invalid unit

// xx= shiftto==6 && missing last '='

// xx=y or last is not '='

if (nextin == 18 || nextin == 12 ||

nextin == 6 && is.read() != '=') {

throw new IOException("Illegal base64 ending sequence:" + nextin);

}

b[off++] = (byte)(bits >> (16));

if (nextin == 0) { // only one padding byte

if (len == 1) { // no enough output space

bits >>= 8; // shift to lowest byte

nextout = 0;

} else {

b[off++] = (byte) (bits >> 8);

}

}

eof = true;

return off - oldOff;

}

@Override

public int read(byte[] b, int off, int len) throws IOException {

if (closed)

throw new IOException("Stream is closed");

if (eof && nextout < 0) // eof and no leftover

return -1;

if (off < 0 || len < 0 || len > b.length - off)

throw new IndexOutOfBoundsException();

int oldOff = off;

while (nextout >= 0) { // leftover output byte(s) in bits buf

if (len == 0)

return off - oldOff;

b[off++] = (byte)(bits >> nextout);

len--;

nextout -= 8;

}

bits = 0;

while (len > 0) {

int v = is.read();

if (v == -1) {

return eof(b, off, len, oldOff);

}

if ((v = base64[v]) < 0) {

if (v == -2) { // padding byte(s)

return padding(b, off, len, oldOff);

}

if (v == -1) {

if (!isMIME)

throw new IOException("Illegal base64 character " +

Integer.toString(v, 16));

continue; // skip if for rfc2045

}

// neve be here

}

bits |= (v << nextin);

if (nextin == 0) {

nextin = 18; // clear for next in

b[off++] = (byte)(bits >> 16);

if (len == 1) {

nextout = 8; // 2 bytes left in bits

break;

}

b[off++] = (byte)(bits >> 8);

if (len == 2) {

nextout = 0; // 1 byte left in bits

break;

}

b[off++] = (byte)bits;

len -= 3;

bits = 0;

} else {

nextin -= 6;

}

}

return off - oldOff;

}

@Override

public int available() throws IOException {

if (closed)

throw new IOException("Stream is closed");

return is.available(); // TBD:

}

@Override

public void close() throws IOException {

if (!closed) {

closed = true;

is.close();

}

}

}

}