

Име : Петко Пенчовски

Предмет: Програмиране с Java, част 1

Дата : 27.01.2018

email : busec@abv.bg

GitHub : <https://github.com/PetkoPenchovski/TheCaesarCipher>

# Дешифроване на Цезар Шифър

## 1. Условие

Конзолно приложение за кодиране като всяка буква се замества с друга буква и се изписват 26 варианта на дешифроване. Програмата изисква въвеждане на текст на латиница.

## 2. Въведение

Програмата е реализирана на платформата Java.

## 3. Теория

Шифърът на Цезар е вид шифър с пряка субституция (*субституционен шифър*) - това означава, че всяка буква се замества само с една и съща друга буква, в целия текст.

Според някои исторически източници, този шифър е използван от Гай Юлий Цезар.

Математически запис:

За английската азбука използване **mod=26**, а за българската азбука **mod =30**.

Шифриране:

$$E_n(x) = (x + n) \mod 30.$$

Дешифриране:

$$D_n(x) = (x - n) \mod 30.$$

като **n** е ключа в случая **n = 3**, **x** е стойността на буквата.

## 4. Използвани технологии

В текущата програма алгоритъмът позволява въвеждане на шифрован текст на латиница и показва неговите 26 варианта на дешифроване.

## 5. Инсталация и настройки

Необходима е инсталирана **Java** на компютъра, след което се стартира файла който се намира в папка с име „src“ ,която е в папката на проекта. В „src“ папка се стартира **Java** файл с име **CaeserCipher**.

## 6. Кратко ръководство

Програмата започва с предлагане на потребителя да въведе текст за дешифроване. След въвеждането на текста се изписват 3 варианта на дешифроване и ако потребителят не е доволен от тях, програмата предлага останилите 23 варианта след изписването на думата „next“.

## 7. Примерни данни

Въвежда се примерен шифрован текст „Jynei“ и на шеста позиция се появя дешифрвания вариант „Petko“.

## 8. Описание на програмния код

Дешифрирането е обособено в метода `decrypt` с параметри въведения текст от конзолата и ключът ( цяло число, което показва с колко позиции да бъдат отместени символите в текста). Зададен е празен стринг **decrypted**, в които в последствие ще се запази кодирания текст чрез конкатенация. С **for** цикъл се обхожда въведения текст като символите се запазват като стойности от **ASCII** таблицата `int c = text.charAt(i);` . След това се прави проверка на символа за главни букви с **Character.isUpperCase(c)** и проверка на символа за малки букви с **Character.isLowerCase(c)** . Дешифроването става с `c -= (key % 26);` като се прави и проверка дали **C** надвишава стойността на 'z' от **ASCII** таблицата и ако е да, то тя се увеличава с 26.

При стартиране на **main** метода се изписва името на програмата, след което с класът **Scanner** се въвеждат текста, който да бъде декодиран. В метода също е декларирана променлива от тип **String** с име **otherOptions** , при която ако стойността и е **“next”** се извеждат в конзолата другите варианти. Използваният метод за сравнение на стойността с променливата е **equalsIgnoreCase()**.

## 9. Приноси на курсиста, ограничения и възможности за бъдещи разширения

Програмата може да бъде направена да работи и на кирилица, а също така може да бъде въведен като ъпгрейд „кодиране“ като опция.

## 10. Източници

[https://bg.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%8A%D1%80\\_%D0%BD%D0%B0\\_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80](https://bg.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%8A%D1%80_%D0%BD%D0%B0_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80)

<https://www.youtube.com/watch?v=44ADNK7ggkA&t=12s>

<https://www.youtube.com/watch?v=OLMrCTjiRuk>

<https://www.youtube.com/watch?v=AcddZcb5hBc>