**Имена:** Валентин Иванов

**Дата: 2018-01-26 Предмет: Програмиране с Java, част 1**

**имейл:** [**valioto@abv.bg**](mailto:valioto@abv.bg) **GitHub:** <https://github.com/SrVallantines/CezariusCipher>

Разбиване на Цезар шифър

**1. Условие**

При въведен криптиран текст да се изведе първоначалният, без да се задава какво е отместването, като това става така:

1. Знае се, че най-често срещаните букви в английският език са ‘**e**’, ‘**t**’ и ‘**h**’
2. Намира се най-често срещаният символ във въведеният текст
3. Пресмята се колко е отместването от него до ‘**e**’, ‘**t**’ и ‘**h**’
4. Извеждат се трите вариянта за първоначален текст
5. Ако потребителят не е доволен от декодираният текст се извеждат и останалите 23 възможни варианта.

**2. Въведение**

Приложението е реализирано на платформата **Java**.

**3. Теория**

Целият алгоритъм е реализиран в средата на **Intelli J** .

**4. Използвани технологии**

В текущата програма алгоритъмът проверява дали въведените стойности са коректни, и ако е така се търсят зададените в условието варианти.

**5. Инсталация и настройки**

Трябва да имате инсталирана **Java** на компютъра си, както и средата за разработка на приложен софтуер **Intelli J** , след което зареждате и стартирате **Java** проекта с име **CezariusCipher**.

**6. Кратко ръководство на потребителя**

Програмата предлага на потребителя в началото да си криптира сам произволен некст, след което същата програма сама го декриптира по указаните в условието стъпки. Ако потребителя желае само да разкодира свой текст, тогава на първият въпрос отговаря с **<N>**, с което преминава направо към декодирането му.

**7. Примерни данни**

Всякакви коректни входни данни.

**8. Описание на програмния код**

.

Съществуват различни методи за реалезация на алгоритъма, от които аз предпочетох динамичното изчисление на ASCII-кодът на символите, и добавянето на ключ (интервал) „на отместването” на всеки символ по отделно. Това дава възможност по-лесно да се оперира с широк интервал символи, вместо с „вградена азбука”, както е при една част от подобните алгоритми.

Програмата стартира с “main” – метода, а след него – по реда на използването им съм разположил и методите за контрол на входните данни, за намиране на най-често среща-ният символ във въведеният текст, след който се намира и методът за декодиране

При извеждането на декодиранят текст в трите най-вероятни комбинаций, както и за принтиране на всички други варианти, съм използвал референтният тип **“StringBuilder”** в методът **„printingDecriptionText”**,

В писането на кода съм стремял да се подържа контакта с потребителя, изразяващ се в потвърждение на въведената информация със съобщения от типа: **"Your text to encrypting is: "**, което – разбира се – не е задължително, когато не е предвиден графичен интерфейс е полезно.

**9. Приноси на курсиста, ограничения и възможности за бъдещо разширение**

Програмата реализира един древен метод за видоизменяне на информацията с цел да е защитена от неправилно, неправомерно и нежелателно използване. При тази реализация на алгоритъма има възможност за усложняването му с въвеждане на допълнителни символи и допълнителни методи за изменение на изходният текст,

**10. Използвани източници**

- <http://www.java2s.com/Tutorial/Java/>

- <https://www.youtube.com/watch?v=pi58jcbs09k>

- <https://www.youtube.com/watch?v=LVHeW1hcdRk>