Encrypter

Erstellen Sie eine JavaFx-Anwendung, mit der ein Text durch einfache Buchstaben-Ersetzung verschlüsselt werden kann.

Dabei wird

- ein Alphabet definiert (Alphabet = all jene Zeichen, die verschlüsselt werden sollen. Hier können neben den Buchstaben auch Ziffern, Sonderzeichen, etc. enthalten sein.
- ein Key generiert, welcher eine zufällige Anordnung der Zeichen des Alphabets darstellt.
- der Text verschlüsselt, indem jeder Buchstabe des Alphabets durch jenen Buchstaben, der sich an der gleichen Stelle im Key befindet, ersetzt. Zeichen im Originaltext, die nicht im Alphabet vorhanden sind, werden unverändert in den verschlüsselten Text übernommen.

Beispiel:

Alphabet	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	ä	ö	ü	ß
Key	g	p	ß	c	s	m	f	n	h	ä	l	t	a	y	ü	u	z	d	r	b	o	i	v	k	ö	w	q	e	x	j

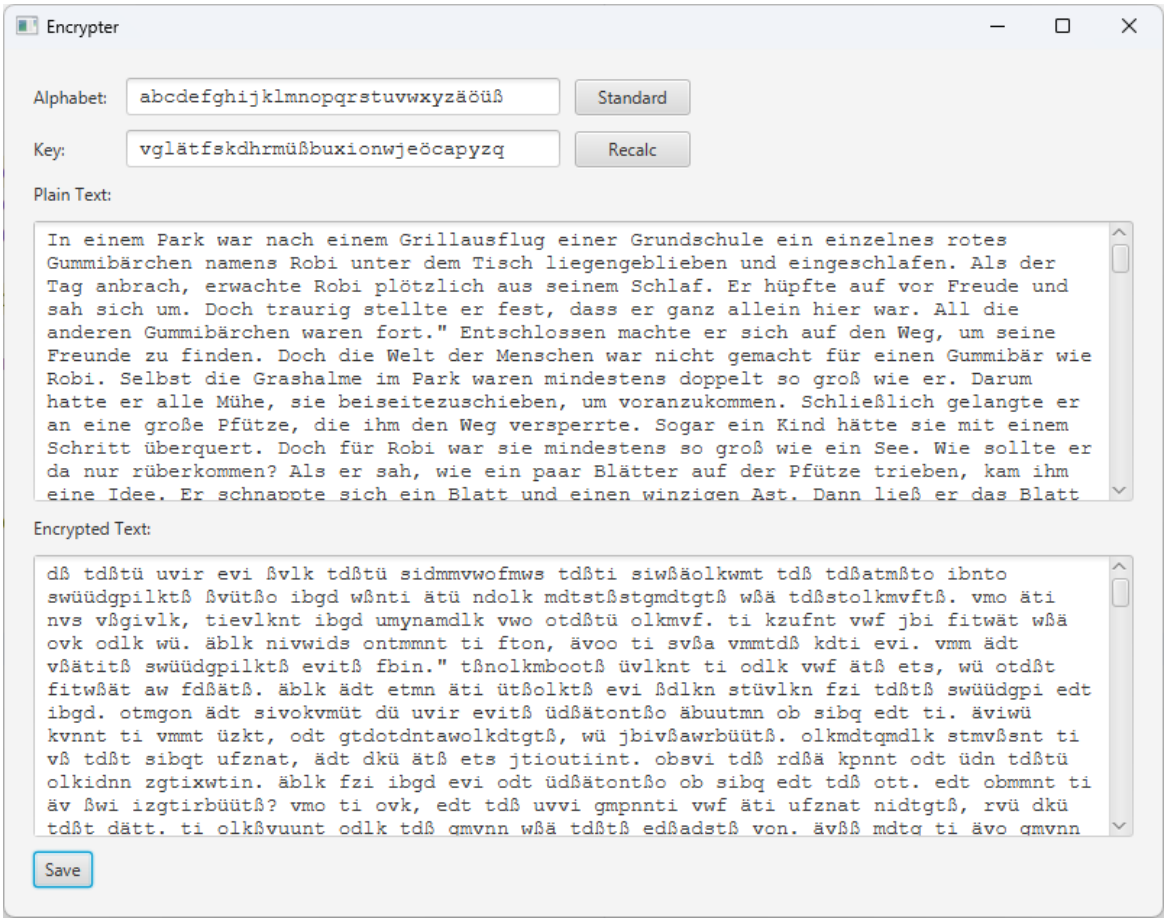
Verschlüsselung:

Originaltext	Verschlüsselter Text
Hallo Welt!	ngttü vstb!

Beachten Sie:

- Der Originaltext wurde zunächst in Kleinbuchstaben übersetzt, da das Alphabet lediglich aus Kleinbuchstaben besteht.
- Zeichen, welche nicht im Alphabet enthalten sind (Ziffern, Leerzeichen, Sonderzeichen, …) sind im verschlüsselten Text erhalten geblieben.

Orientieren Sie sich bei der Anwendung an folgendem Screenshot:

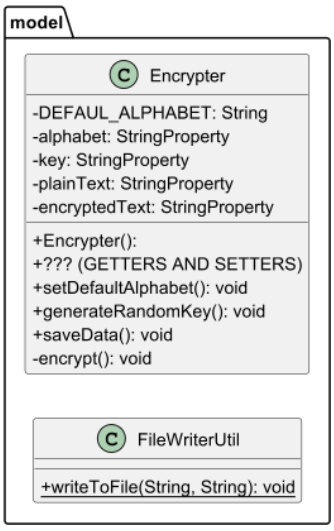


Implementieren Sie folgende Funktionalität:

- Wenn ein Alphabet eingegeben wird, wird dafür automatisch ein zufälliger Key generiert.
- Bei Klick auf „Standard“ wird ein Standardalphabet gesetzt.
- Bei Klick auf „Key“ wird ein neuer zufälliger Key generiert.
- Wird in „Plain Text“ ein Text eingegeben, so wird diese sofort und ständig in den verschlüsselten Text übersetzt.
- Bei Klick auf „Save“ werden der Dateien im Ordner „files“ erstellt:
  - `alphabetAndKey.txt`: Enthält das Alphabet und den Key
  - `plaintText.txt`: Enthält den Originaltext.
  - `encryptedText.txt`: Enthält den verschlüsselten Text.
- Eine Markierung im Originaltext soll auf den verschlüsselten Text (und umgekehrt) übertragen werden.

Task 1: *Encrypter* und *FileWriterUtil* (Model)

Die gesamte Funktionalität zum Verschlüsseln des Textes soll in der Klasse `Encrypter` gebündelt werden. Das Schreiben von Files soll durch die `FileWriterUtil`-Klasse bewerkstelligt werden.



Beachten Sie:

- Die relevanten Felder (alphabet, key, …) sind StringProperty-Objekte. Damit ist es möglich, Eigenschaften von Steuerelementen der View an diese Properties zu binden. Diese Bindung von Properties stellt sicher, dass die View immer den aktuellen Zustand des Modells darstellt.
- Legen Sie nur jene Setter an, die benötigt werden. Überlegen Sie, welche Setter NICHT angeboten werden dürfen.

Wichtige Methoden:

- `setDefaultAlphabet()` setzt das Alphabet auf einen in der Klasse definierten Standardwert.
- `generateRandomKey()` erzeugt einen zufälligen Key.
  - `TIPP:` Nutzen Sie Streams zu einfachen erzeugen einer zufälligen Permutation des Alphabets.
- `encrypt()` verschlüsselt den Originaltext mit dem aktuellen Key.

Im Modell gibt es Abhängigkeiten zwischen den Properties:

- Wenn das Alphabet geändert wird, muss der Key neu generiert werden.
- Wenn sich der Key ändert, muss der Text neu verschlüsselt werden.
- Wenn sich der Originaltext ändert, muss der Text neu verschlüsselt werden.

Diese Abhängigkeiten können durch entsprechende Listener im Modell im Konstruktor definiert werden.

```
public Encrypter() {
    // generate a new random key, whenever the alphabet changes
    alphabet.addListener((observable, oldValue, newValue) -> generateRandomKey());
    ...
}
```

T2 FXML-Datei encrypter-view.fxml (View)

Erstellen Sie die Benutzeroberfläche in der fxml-Datei.

- Nutzen Sie geeignete Layout-Container, um die gewünschte Anordnung der Steuerelemente zu gewährleisten.
- Nutzen Sie geeignete Steuerelemente und Attribute, um die Benutzeroberfläche und deren Bedienung in der gewünschten Form zu ermöglichen.

T3 *EncrypterController.java* (Controller)

Steuern die mit dem Controller die Nutzung des Modells:

- Erstellen Sie eine klassenweite Instanz des Modells.
- Erstellen Sie bidirektionale (= in beide Richtungen wirkende) Bindungen zwischen Properties der Steuerelemente und Properties des Modells. Beispiel:
    - Die Text-Property des Alphabet-TextFelds soll an die Alphabet-Property des Modells gebunden werden.
- Erstellen Sie Event-Handler, welche bestimmte Benutzeraktionen an das Modell übertragen. Beispiel:
    - Der Klick auf „Recalc“ soll ein neuer zufälliger Key generiert werden, also die Methode `generateRandomKey()` der Modells aufgerufen werden.
- Erstellen Sie Listener, welche eine Reaktion auf bestimmte Benutzeraktionen erzeugen. Beispiel:
    - Wenn im verschlüsselten Text eine bestimmte Textstelle markiert wird, soll synchron immer auch die entsprechende Textstelle im entschlüsselten Text markiert werden.