Dictionary

Die FxIds sind exakt so zu benennen wie die Variablen des entsprechenden Tests. Zur Ausführung des Tests muss die Variable `PATH_TO_FXML` so verändert werden, dass sie auf das richtige `fxml`-File verweist.

1 Dictionary

Erstelle eine Klasse `Dictionary`, welche als Instanzvariable zwei Maps enthält:

- `Map<Integer, Set<String>> wordsGroupedByLength` - enthält für jede Zahl alle Strings, welche diese Länge besitzen
- `Map<Integer, Set<String>> wordsGroupedByDistinctCharCount` - enthält für jede Zahl alle Strings, welche so viele verschiedene Buchstaben besitzen; **nicht** casesensitive

```
wordsGroupedByLength.get(3) -> ["Zug", "ist", "zug", "BGB", ...]
wordsGroupedByDistinctCharCount.get(2) -> ["toto", "Nennen", "Mama", "WC", ...]
```

Dem Konstruktor dieser Klasse wird ein Dateiname übergeben; der Konstruktor liest die Datei zeilenweise und füllt die Maps.
Die übergebene Datei ist UTF-8 codiert und muss auch entsprechend gelesen werden.
Einige Zeilen müssen verworfen, andere "bereinigt" werden. Es sollten 71904 Wörter übrigbleiben.

Zu implementierende Methoden; **nicht** casesensitive:

- `Set<String> getPermutations(String word)` - gibt alle Wörter zurück, welche exakt dieselben Buchstaben wie `word` haben; alphabetisch sortiert
- `Set<String> getWordsWithSameLetters(String word)` - gibt alle Wörter zurück, welche exakt dieselben Buchstaben wie `word` haben, wobei jedoch Buchstaben öfter oder weniger oft vorkommen können; alphabetisch sortiert

```
getPermutations("esi") -> ["Eis", "eis", "sei", "sie"]
getWordsWithSameLetters("esi") -> ["Eies", "Eis", "eies", "eis", "sei", "sie"]
```

2 FX

Erstelle eine graphische Oberfläche für obige Applikation. Orientiere dich dabei an folgendem Layout: