

Sofia University
Department of Mathematics and Informatics

Course : [AOOP-1](#)

Date: May 11, 2020

Student Name:

Домашно No. 6

Напишете Java модулно приложение, наречено **Homework6**, което по даден ключ (цяло число) да генерира масив от произволно избрани символи **A-Z** като предоставя възможност **символите да се генерират по различен начин**. Понеже, методът за генериране на тази последователност трябва да е таен и недостъпен, то **използвайте затваряне(closure)** за скриване на програмната реализация на този метод.

За целта изпълнете следните действия:

1. Създайте модул `com.services` и в подходящо именуван `package` на този модул **напишете** `interface Cipherable`. Този интерфейс да декларира метод `getSecretChars(int seed)`, който взима за аргумент **цяло число** и връща **масив от символи**.
2. Създайте модул `com.providers` и в подходящо именуван `package` на този модул **напишете следните класове:**
 - ✓ `class Wrapper`, който **има целочислена данна `size`**, която задава дължината на извеждания масив от символи. **Напишете конструктор** за общо ползване, **getter** и **setter** метод.
 - ✓ `class Result`, който има **данни `chars` (от `char[]`) и `data` (от `int` тип)**. Напишете **getter** и **setter** за тези данни, конструктор за общо ползване и `toString()` метод.
 - ✓ `class CipherGenerator`, който **има статичен метод**
`Result countDistinct(Cipherable cipher, int seed)`
3. Имплементирайте `interface Cipherable` във вътрешни класове `FixedRandom` и `FixedSelection` на `class Wrapper`.
 - ✓ **Напишете** реализация на метод и `getSecretChars(int seed)` във вътрешен клас `FixedRandom` на `class Wrapper`, където посредством инстанция на клас `Random` да се генерира една и съща последователност от произволно избрани символи в интервала A-Z за всяка конкретна стойност на аргумента `seed` на метода `getSecretNumbers` (спомнете си как се използва конструкторът за общо

ползване на клас `Random`). Символите да се запишат в реда на генерирането им в масива с дължина `size`, връщан от `getSecretChars(int seed)`. **Напишете** метод `.makeFixedSelection()` във външния клас, който връща тип `Cipherable` със съдържание обект от клас `FixedRandom`

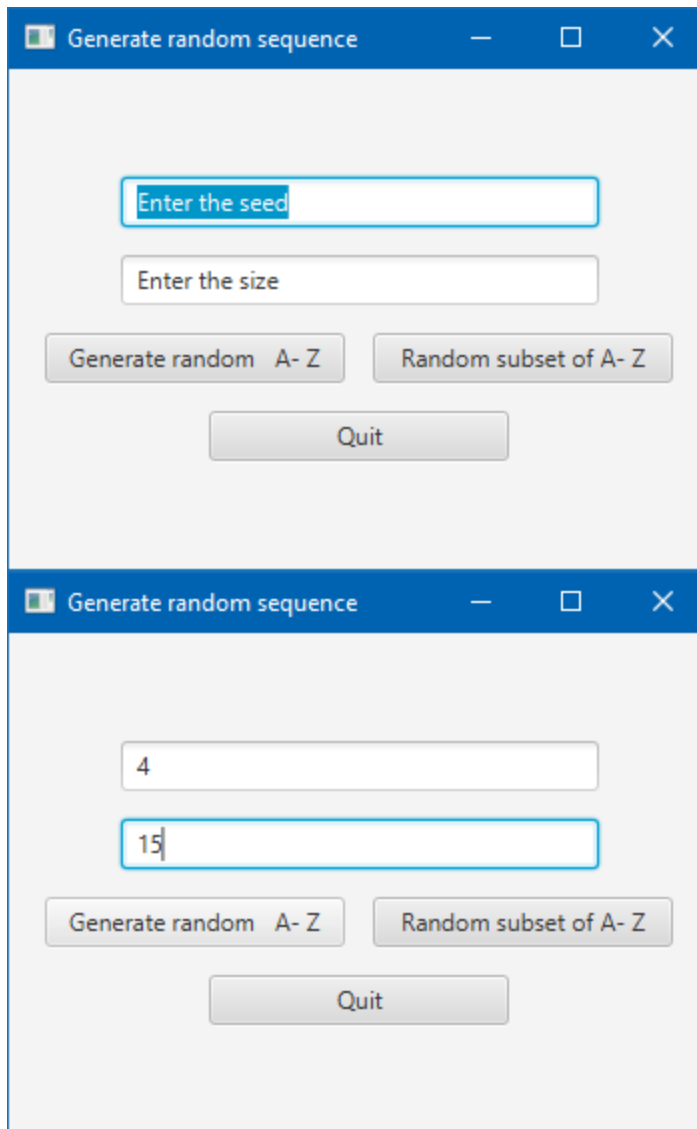
- ✓ **Напишете** реализация на метод `getSecretChars(int seed)` във вътрешен клас `FixedSelection` на `class Wrapper`, където посредством инстанция на клас `Random` да се генерира произволна последователност от `seed` произволно избрани символи в интервала A-Z. Произволна последователност от така избраните символи да се запишат в реда на генерирането им в масива с дължина `size`, връщан от `getSecretChars(int seed)`. **Напишете** метод `makeFixedRandom()` във външния клас, който връща тип `Cipherable` със съдържание обект от клас `FixedSelection`

4. **Напишете** реализация на статичния метод

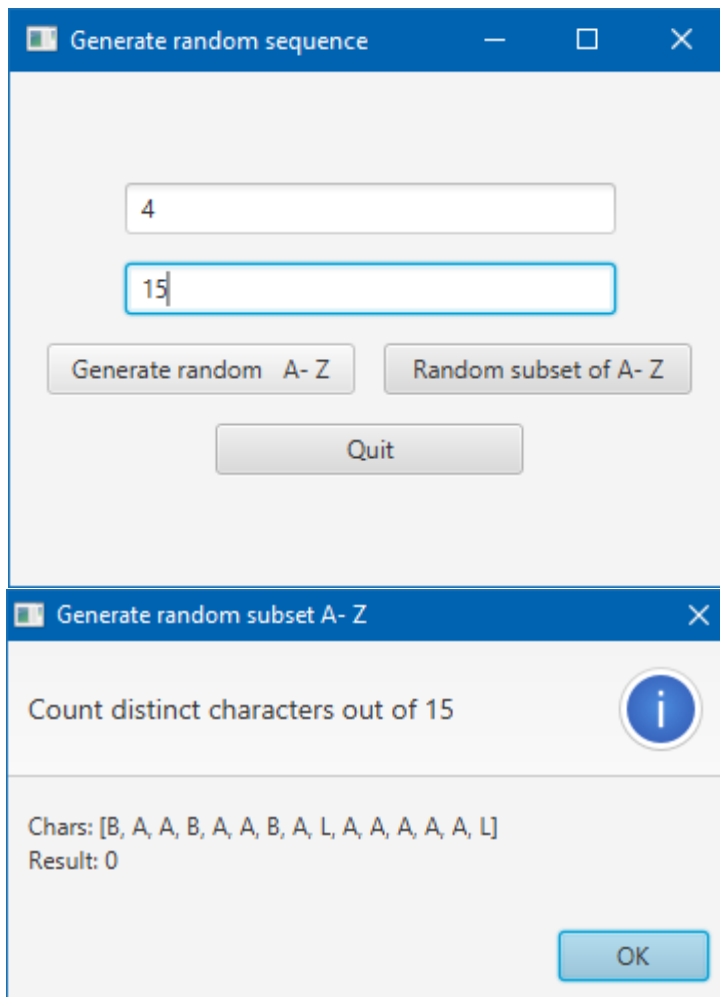
```
Result countDistinct(Cipherable cipher, int seed)
```

в `class CipherGenerator`, използващ различни начини за генериране на символи. Методът пресмята броя на уникалните символи в масива от символи, генерирани от метода `getSecretChars(seed)` на обекта, рефериран с параметъра `cipher`. Методът връща обект от тип `Result`, който капсулира масива с генерираните символи и броят на не повтарящите се символи измежду тях.

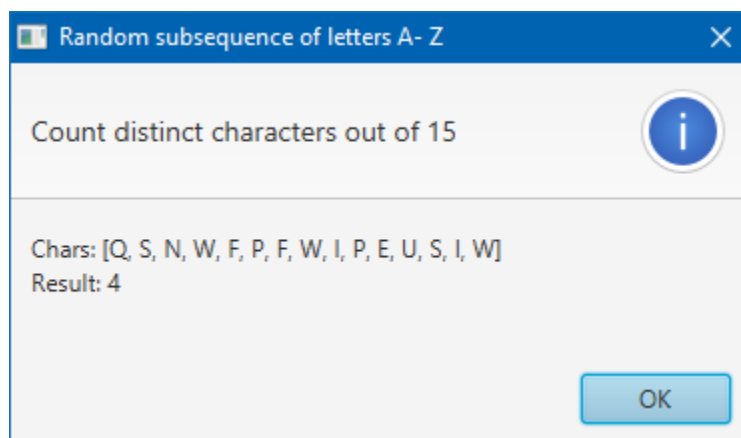
5. **Създайте** модул `com.gui` и в подходящо именуван `package` на този модул за създаване на JavaFX приложение. Нека това JavaFX приложение да има следния дизайн на потребителския интерфейс



6. Бутонът **Generate random A- Z** да извежда **Alert.AlertType.INFORMATION** диалогов прозорец с елементите на масива от символи, генериран от доставчика **makeFixedRandom ()** на тази услуга заедно с броя на неповтарящите се символи в този масив.



- a) Бутонът “Random subset of A- Z” да извежда `Alert.AlertType.INFORMATION` диалогов прозорец с елементите на масива от символи, генериран от доставчика `makeFixeSelection()` на тази услуга заедно с броя на неповтарящите се символи в този масив.



b) Бутонът "Quit" прекратява изпълнението на програмата.

Изпълнете и тествайте приложението с различни входни данни.