Lab No. 6

Submit the all Java files developed to solve the problems listed below.

Задача 1

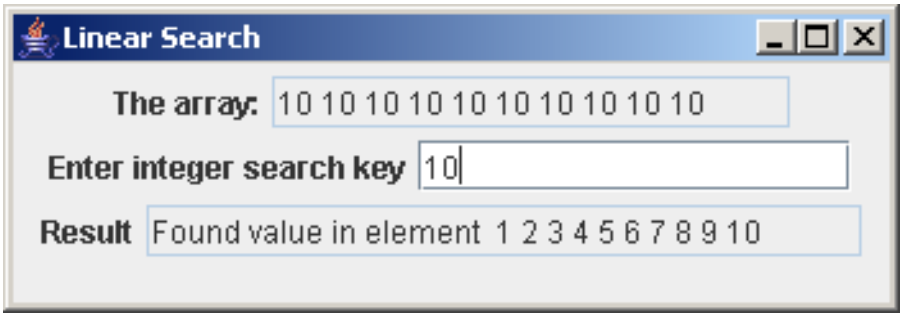
Програмната реализация на методите за търсене, представени в ***class LinearArray*** на **Fig. 16.2** (*последователно търсене*) и в **`class BinaryArray`** на **Fig. 16. 4** (*бинарно търсене*) позволяват да се открие **точно едно съвпадение** на зададения шаблон с елемент от масива **(**добавен за удобство в ***`SampleCodeLab6.rar`***). Тази програмна реализация **не позволява да се открият всички елементи от масива**, които **съвпадат** с дадения шаблон.
За целта **разгледайте реализацията** на ***class LinearArray*** с **графичен потребителски интерфейс**, състевен от **`Swing`** компоненти, даден на лекция във **`Fig. 07.12`** **на** ***`лекция Ch-07Plus`***, (**`Fig. 07.12`** е добавен за удобство в ***`SampleCodeLab12b.rar`*** **)**

- **Добавете** метод
  `public String linearSearchAll( int searchKey )`

в **`class LinearArray`** на **`Fig. 07. 12`**, който да прилага ***метода на последователно търсене*** за съвпадение с **`searchKey`** и да **връща** **`String`** с **всички индекси на елементи** от клас данната **`int[]`** data ,които съвпадат с **`searchKey`** (индексите да са разделени със празен символ). ***Преценете*** *каква стойност да връща метода* ***при липса на съвпадение.***

- **Променете** метод
  `public void actionPerformed( ActionEvent event)`

в **`class LinearArrayTest`** на **`Fig. 07. 12`**, така че да **тествате** новосъздадения метод **`linearSearchAll()`** по аналогия с тестването на метод **`linearSearch()`**
Изпълнете и тествайте приложението с различни данни (предварително инициализирайте **`масива`** *data в class* **`LinearArray`** *с* ***инициализиращ списък****, съдържащ дублиращи се стойности)*



- По аналогичен начин, **добавете** метод
  `public String binarySearchAll( int searchKey )`

в **`class BinaryArray`** на **`Fig 16.04`**, който да прилага ***метода на бинарно търсене*** за съвпадение с **`searchKey`** и да връща **`String`** с всички индекси на елементи от клас данната **`int[] data`** ,които съвпадат с **`searchKey`** (индексите да са разделени със празен символ).
Преценете каква стойност да връща метода при липса на съвпадение

*[Упътване: **Използвайте**, че вече написания метод* **`binarySearch()`**, *а също и, че масива* **`data`** *е **сортиран**, за да **получите списък с индекси** на повторенията]*

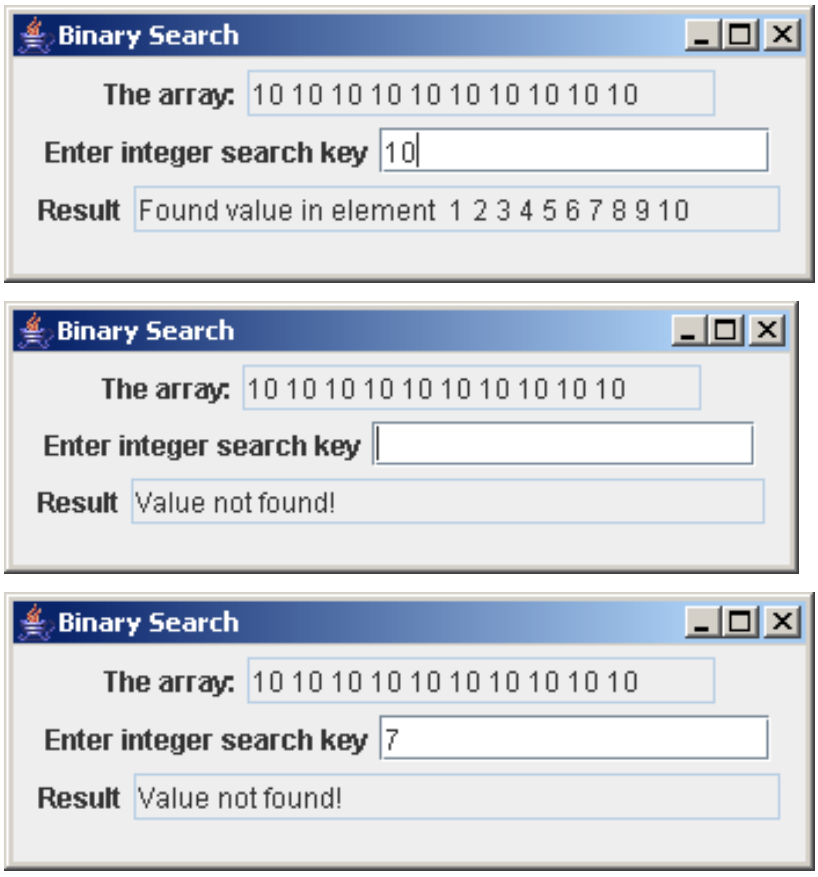
Пренапишете **`BinarySearchTest.java`** от **`Fig16.05`** по аналогия с *class* **`LinearArrayTest`** на **`Fig. 07. 12`** , за да използва същите такива етикети, текстови полета и възможност за въвеждане на шаблон за търсене, както **`class LinearArrayTest`**

- **Променете** метод

  `public void actionPerformed( ActionEvent event)`

в променения **`class BinarySearchTest,`** за да тествате новосъздадения метод **`binarySearchAll()`** по аналогия с тестването на метод **`binarySearch()`**

***Изпълнете** и **тествайте** приложението с различни данни (предварително инициализирайте масива data в* **`class BinaryArray`** *с **инициализиращ списък**, съдържащ **дублиращи** се стойности*)



Problem No. 2a

Да се реализира **алгоритъма за сортиране с преброяване** като приложение на Java.

Напишете *`class CountSort`*, който има

- **Референция** към масив ***`intArr[]`***от цели положителни числа, **set** и **get методи (***да се удовлетвори спесифичното изискване на алгоритъма)*, ***`toString`***() **метод**
- **Конструтор по подразбиране**, който инициализира ***`intArr`*** с ***`20`*** случайни числа, в интервала [10,20]

- **Конструтор за копиране**, който инициализира ***intArr*** със стойностите на референрания обект за копиране
- **Конструтор за общо ползване**, който инициализира ***intArr*** като копие на даден масив
- Метод ***private int getMax()***, който връща най- голямото число k от масива intArr
- Метод ***public int[] sort(),*** който връща масива ***intArr*** след като е **сортиран във възходящ ред** по **алгоритъма са сортиране** с преброяване даден на лекции

Напишете ***class CountSortTest*** за тестване на метода ***sort***() на ***class CountSort***- изведете изходния и сортирания масив на стандартен изход подходящо форматирани

Problem No. 2b

Да се реши задача **Problem No. 2a** като се направят необходимите **промени в алгоритъма за сортиране с преброяване**, така че **даденият масив да се сортира в низходящ ред**