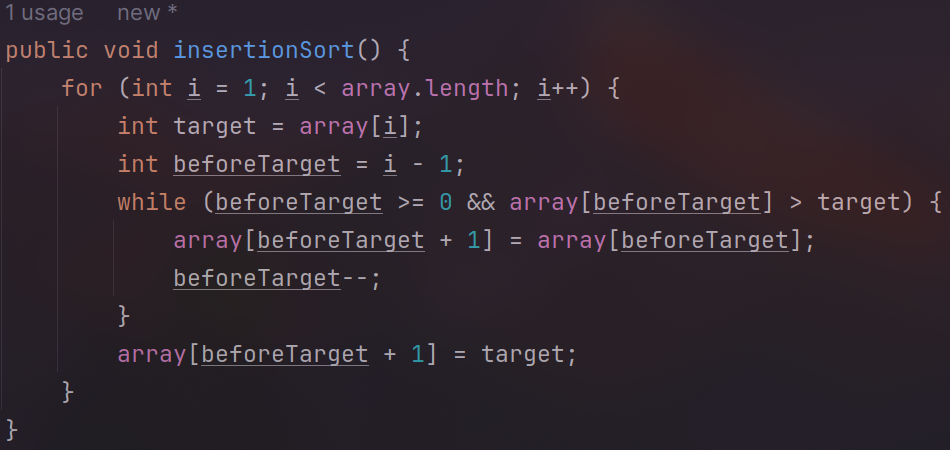
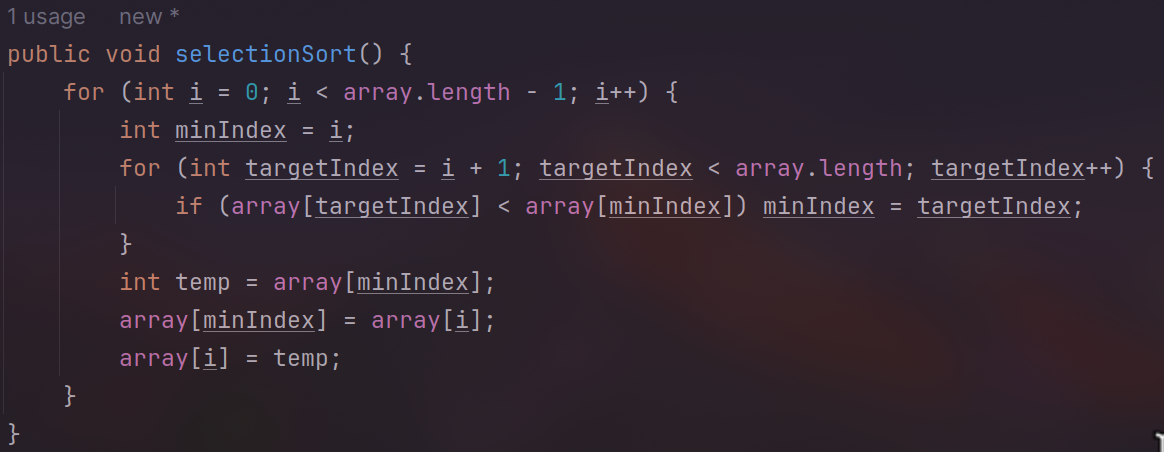
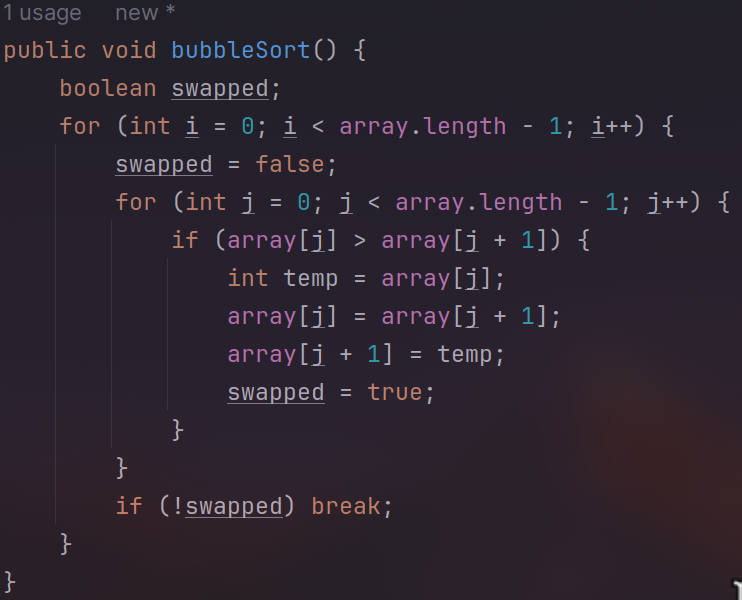
|  |
| --- |
| **Teme:** **Enorazsežne tabele (utrjevanje):**  - enostavni postopki razvrščanj:   * + vstavljanje,   + izbiranje,   + mehurčno razvrščanje.   - metode razreda (knjižnice) Arrays : sort, binarySearch |

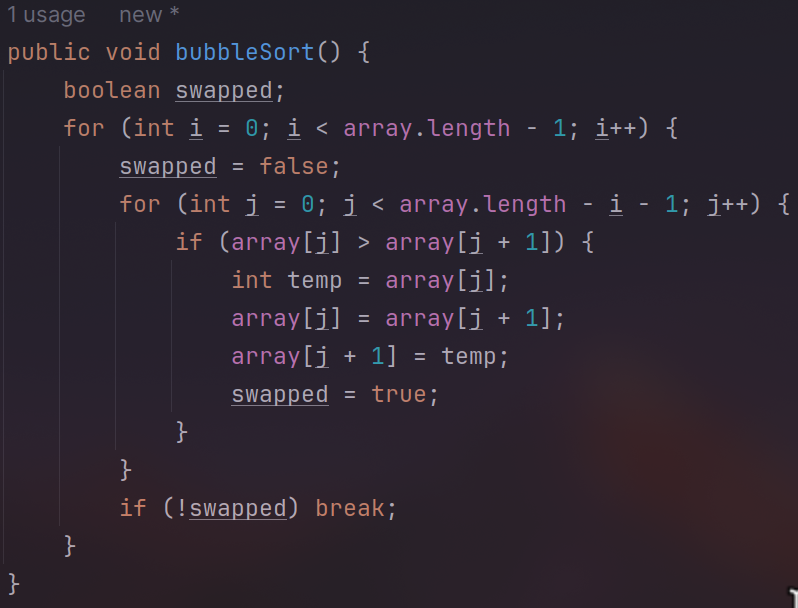
**Naloga 1**

Izvedite in preskusite delovanje naslednjih postopkov :

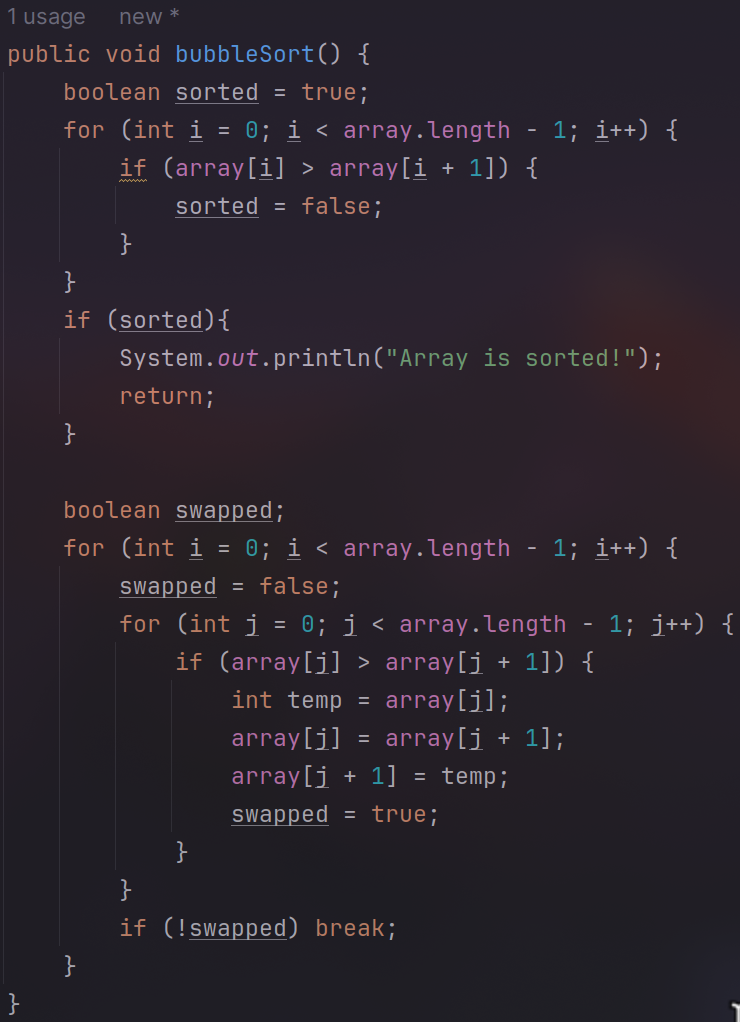
1. razvrščanje z vstavljanjem
2. razvrščanje z izbiranjem
3. razvrščanje z mehurčki
   1. kjer ne upoštevate meje že razvrščenega dela zaporedja



* 1. kjer upoštevate mejo in razvrščate le po neurejenem delu

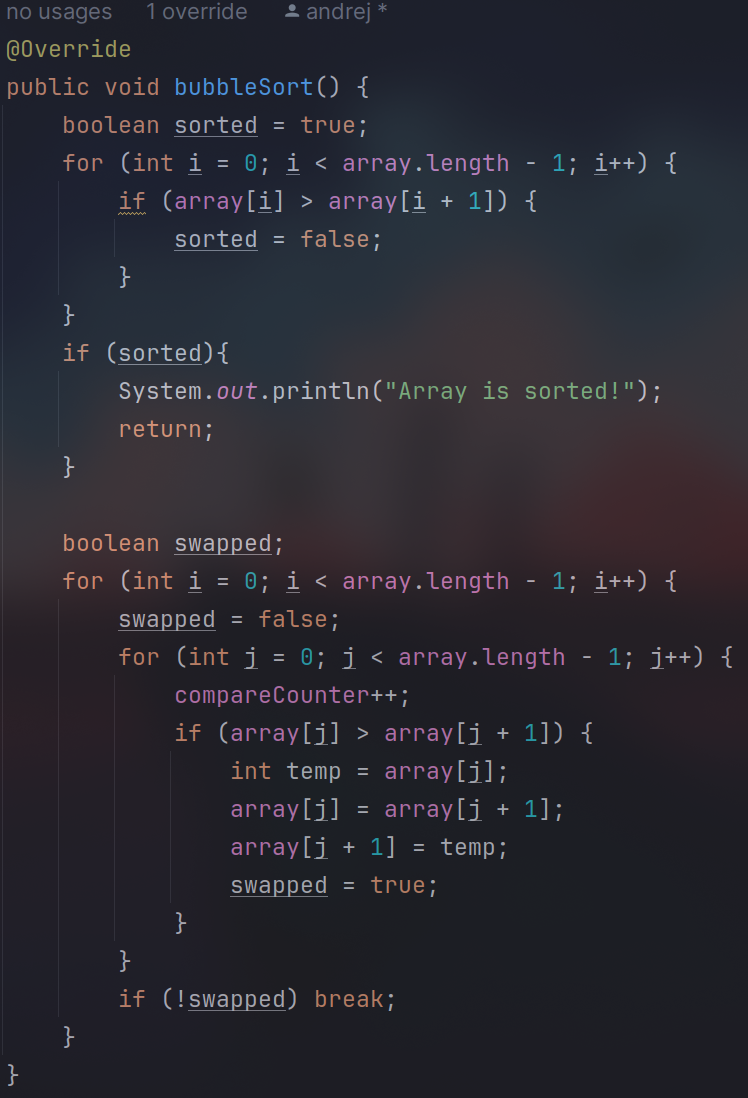
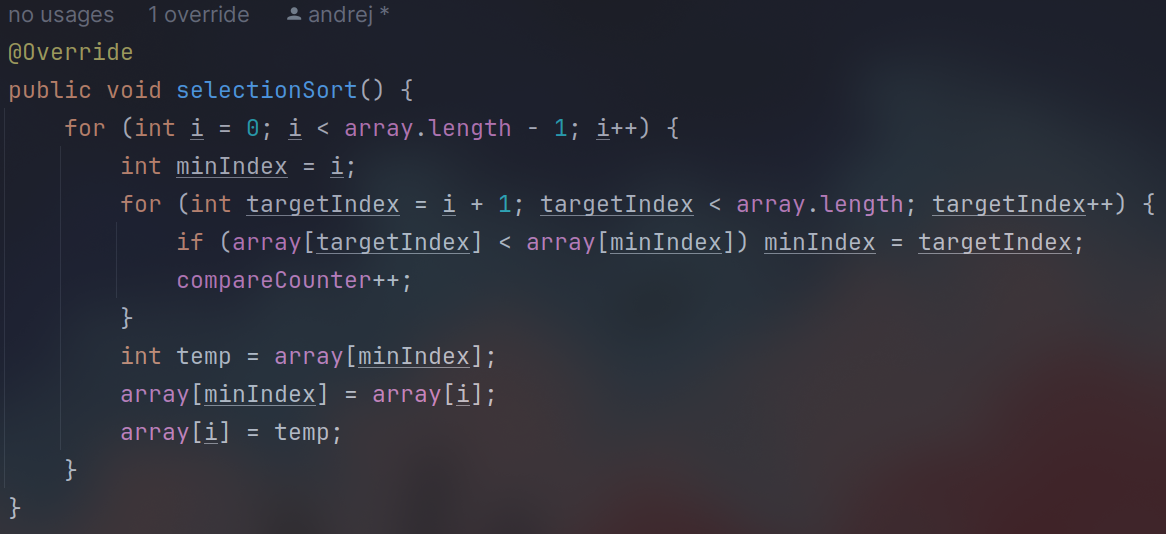
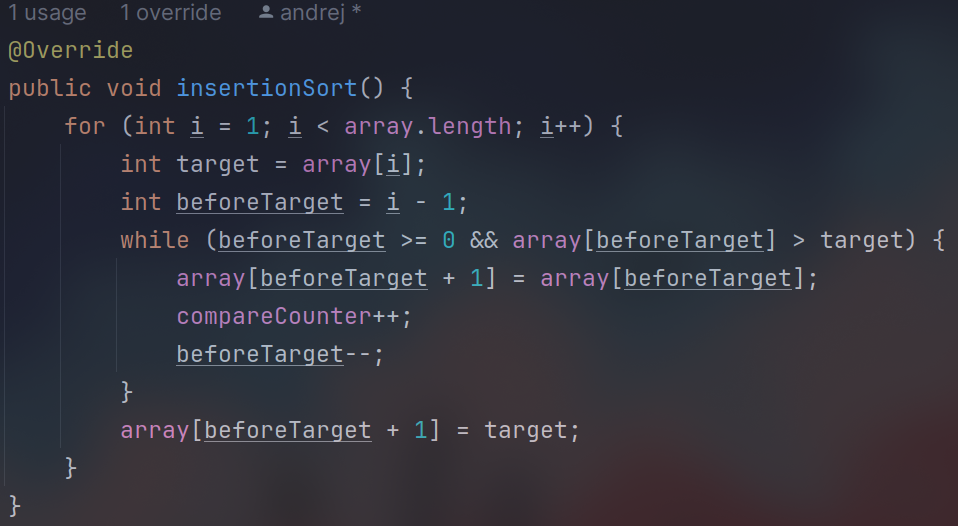


* 1. kjer ugotavljate, če je zaporedje že urejeno in v primeru urejenosti končate z razvrščanjem



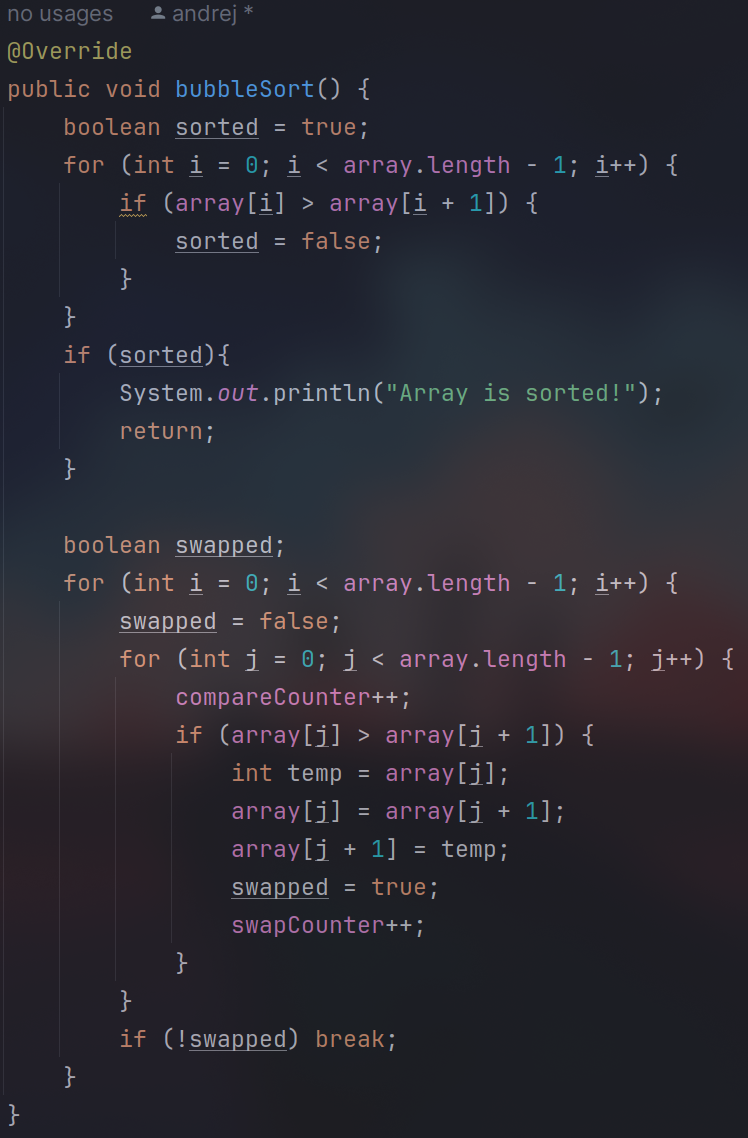
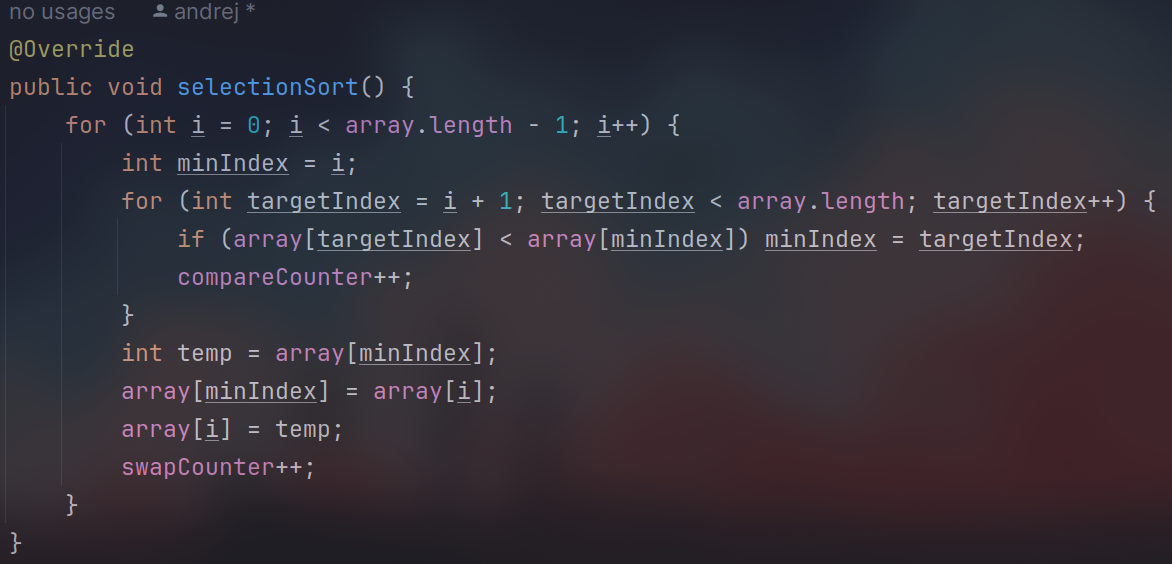
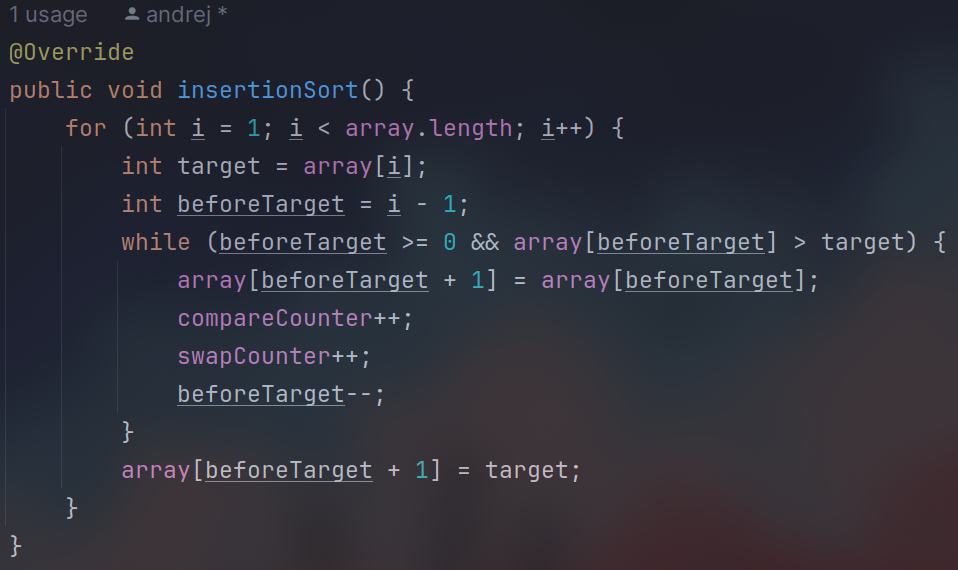
**Naloga 2**

Predelajte primere iz naloge 2 tako, da bodo hkrati šteli število primerjav potrebnih za razvrstitev elementov.

****

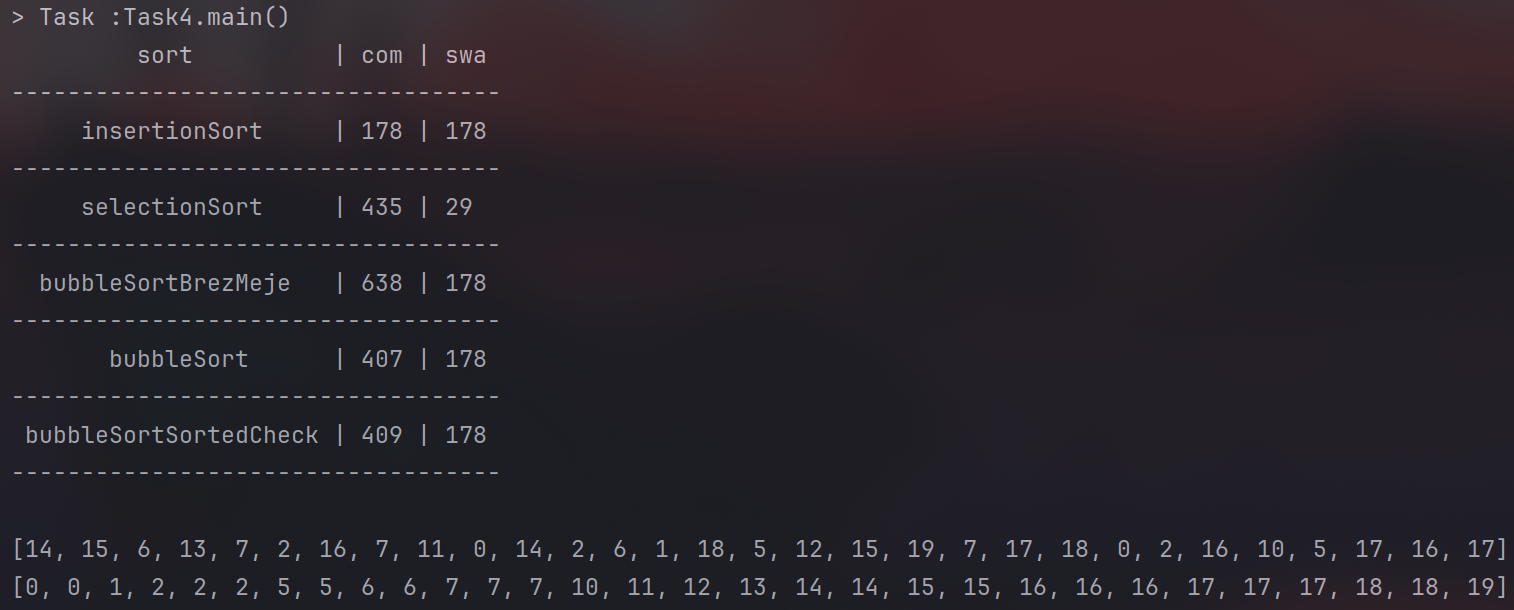
**Naloga 3**

Predelajte primere iz naloge 2 tako, da bodo hkrati šteli število zamenjav elementov potrebnih za razvrstitev elementov (vsak premik elementa v tabeli naj bo tudi zamenjava)

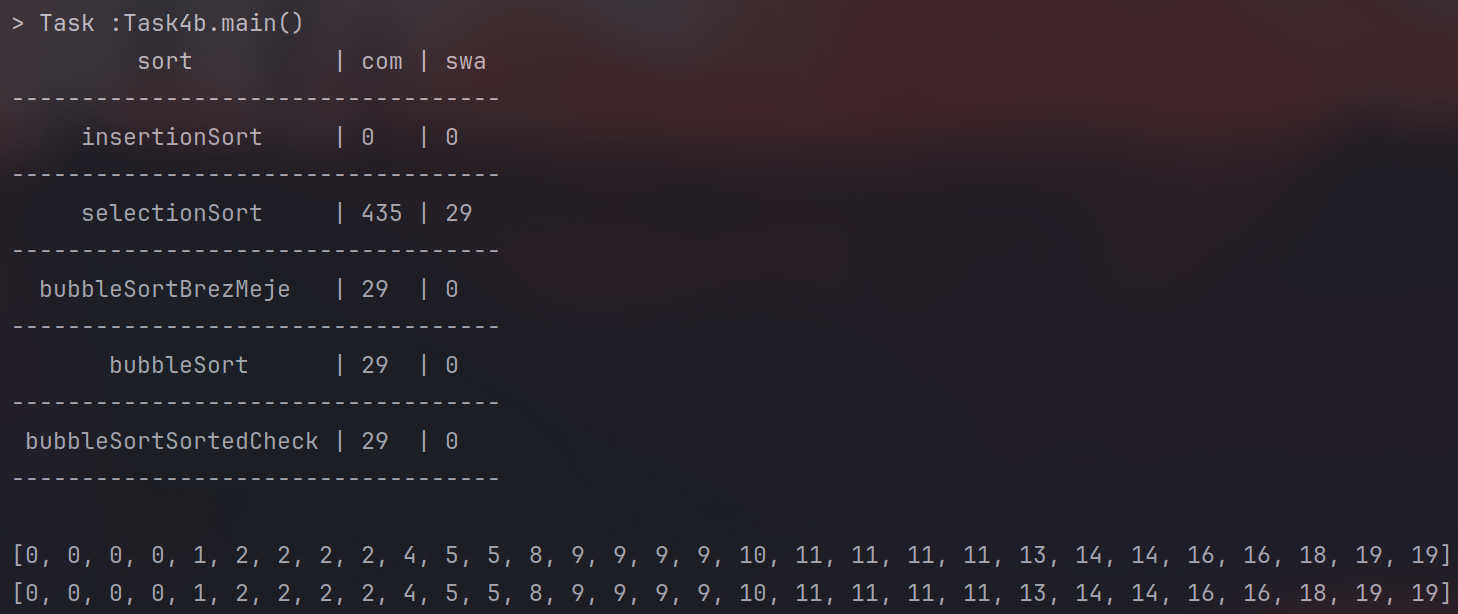
****

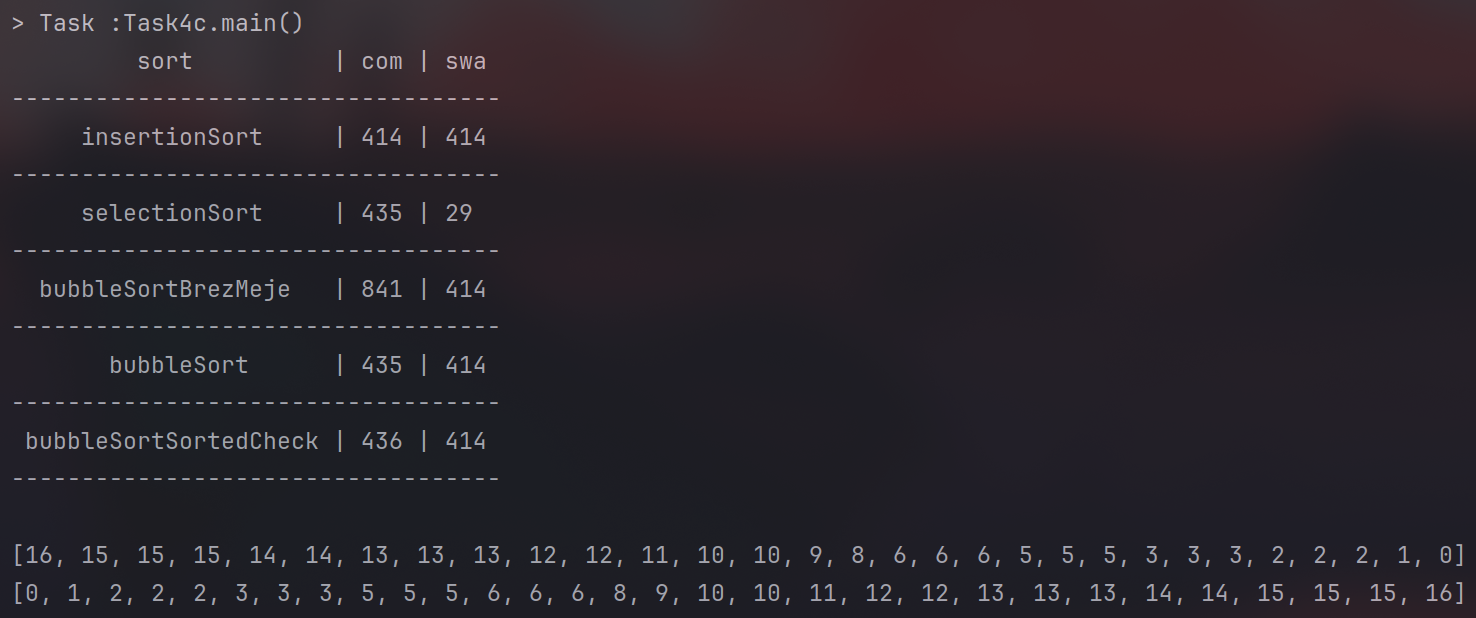
**Naloga 4[[1]](#endnote-2)**Izdelajte primerjalno tabelo rezultatov dobljenih iz št-zamenjav/primerjav za vse postopke naloge 2 (5 postopkov) !

1. prva izvedba vseh petih postopkov naj bo nad zaporedjem, ki ste uporabili v nalogah 2-4 (fiksno, neurejeno zaporedje 30 celih števil)



1. druga izvedba naj se izvede nad tabelo 30 števil, ki je nepadajoče urejena



1. tretja izvedba naj se izvede nad tabelo 30 števil, ki je nenaraščajoče urejen

|  |
| --- |
| *Spodnje naloge izvedete s striktno uporabo metod razreda Arrays za vse mehanizme, ki so definirani z metodami tega razreda* |

**Naloga 5**

Izvedite:

1. Napolniti tabelo znakov dolžine 100 z znaki 'A' (/fill)
2. Napolni tabelo necelih števil dolžine 100 z vrednostjo 12.3
3. Preverite delovanje metode equals na dveh tabelah celih števil:
   * Obe tabeli enako dolgi, ena je napolnjena z vrednostmi 10, druga z vrednostmi 11;
   * Prva tabela dolga 10, druga 12 števil, obe napolnjeni z vrednostmi 11;
   * Obe tabeli enako dolgi, napolnjeni z enakimi vrednostmi.
4. Napolnite tabelo celih števil z naključnimi vrednostmi med 0 in 5 (/setAll) :

setAll(tabela,v->(int)(Math.random()\*6)

1. Naredite natančno kopijo tabele iz predhodno naključno generirane tabele (/copyOf)
2. Naredite kopijo predhodno narejene tabele; kopijo naj bo krajša za zadnja dva elementa(copyOfRange)
3. Naredite kopijo predhodno narejene tabele; nova tabela naj bo daljša za 5 elementov

**Naloga 6**

Napišite javanski program, ki ustvari tabelo pozitivnih celih števil velikosti N elementov. N je parameter programa. Tabelo inicializira z naključnimi vrednostmi, manjšimi od 10. Program izpiše vrednosti vseh elementov tabele, nato razvrsti vrednosti po velikosti od najmanjše do največje in vse vrednosti v tabeli ponovno izpiše.

**Naloga 7**

Ponovite nalogo 6 s tem, da program razvrsti le elemente srednje tretjine elementov.

**Naloga 8**

Ponovite nalogo 6 s tem, da program generira vrednosti do 30, da je N vsaj 15. Nato razvrsti elemente med prvo najdeno vrednostjo 9 in prvo najdeno vrednostjo 18. Če nobene od njiju ni v tabeli, ne naredi ničesar, če najde zgolj eno od njiju, razvrsti elemente od najdene vrednosti do konca tabele.

**Naloga 9**

Napišite javansko metodo int[] razvstiParc(int[] tabela, int dolz), ki podzaporedja dolzine 'dolz' v zaporedju tabeli 'tabela' parcialno razvrsti; npr.: vzame prvih 'dolz' in jih razvrsti, nato drugih 'dolz' in jih razvrsti, ….

**Naloga 10**

Napišite javansko metodo, ki bo v tabeli (podani s parametrom metodi) v naraščajočem vrstnem redu razvrstila vse elemente, ki imajo vrednost med 6 in 15, vsi ostali pa naj bodo razvrščeni v padajočem vrstnem redu. Npr.:

1,1,2,4,**15,11,8,7,6**,22,22,38,41.

1. Neobveza; vendar potreben pogoj za boljšo oceno [↑](#endnote-ref-2)