

Cvičenie 12 – Triedenia tabuliek

Cieľ:

Cieľom cvičenia je implementovať aspoň jeden jednoduchý (zložitosť N^2) a aspoň jeden pokročilý algoritmus triedenia tabuliek (zložitosť lepšia ako N^2).

Postup:

1. Stiahnite si zo vzdelávania zdrojové súbory potrebné pre toto cvičenie.
 - Do balíčka gui v projekte DataStructures prekopírujte súbor PanelTable.java (pôvodný súbor nahraďte).
 - Do balíčka datastructures.tables.Sorts nakopírujte súbory ISortableTable.java, Sorter.java a Shuffle.java.
2. Prejdite si rozhranie ISortableTable.
3. Nahraďte rozhranie ITable v triede NonsortedSequenceTable rozhraním ISortableTable. Implementujte chýbajúce metódy tohto rozhrania (swap, move a getKeyAtIndex).
4. V triede FrameMain je potrebné zviazať triediace algoritmy (potomkovia triedy Sorter, napr. trieda Shuffle) s príslušnými panelmi, ktoré ovládajú tabuľky (teda triedy PanelTable). Potomkov triedy Sorter je možné registrovať len takým panelom PanelTable, ktoré ovládajú tabuľky implementujúce rozhranie ISortableTable. Algoritmy zaregistrujete tak, že panelu pošlete správu registerSort s parametrami:
 - inštancia triediaceho algoritmu (napr. new Sorter()),
 - inštancia komparátora kľúčov tabuľky (generický typ komparátora sa musí zhodovať s generickým typom kľúčov tabuľky, teda ITable<Integer, Object> je možné triediť len komparátorom KeyComparer<Integer> - my môžeme využiť IntegerComparer) a
 - reťazec reprezentujúci popis triediaceho algoritmu.

Príklad registrácie „triediaceho“ algoritmu Shuffle do panelu panelNonSortedTable (ovláda inštanciu triedy NonortedSequenceTable, ktorej kľúče sú inštancie Integer) je:

```
panelNonSortedTable.registerSort(new Shuffle(), new IntegerComparer(), "Shuffle");
```

5. Vytvorte potomka triedy Sorter, ktorý bude v metóde sort implementovať jednoduchý triediaci algoritmus (InsertSort, SelectSort alebo BubbleSort). Vytvorenú triedu zaregistrujte príslušnému panelu podľa bodu 4.
6. Vytvorte potomka triedy Sorter, ktorý bude v metóde sort implementovať pokročilý triediaci algoritmus (QuickSort, HeapSort, ShellSort, RadixSort alebo MergeSort). Vytvorenú triedu zaregistrujte príslušnému panelu podľa bodu 4.
7. Porovnajte výkon Vašich algoritmov na veľkej tabuľke. Preskúmajte vlastnosti triediacich algoritmov (prirodzenosť, neutrálnosť a triedenie na mieste).

Čo sa dnes môže hodiť:

Na moodle-i predmetu sú v 12. týždni dostupné pseudokódy triediacich algoritmov a demonštračný program, ktorý znázorňuje princíp jednotlivých algoritmov.

Čo môžete spraviť doma:

Implementujte aj ostatné varianty triediacich algoritmov, zaregistrujte ich do príslušného panelu a porovnajte ich výkon a vlastnosti podľa bodu 7.