

Cvičenia č. 2, úloha č. 4

Cieľom tejto úlohy je implementovať počítanie ostrovov z 18. prednášky zimného semestra v jazyku Java s využitím objektovo orientovaného programovania.

Mapou súostrovia budeme rozumieť obdĺžnikovú mriežku pozostávajúcu z políček dvoch typov zodpovedajúcich *moru* a *pevnine*. Takúto mapu môžeme reprezentovať dvojrozmerným poľom booleovských hodnôt `map`, kde `map[i][j] == true` práve vtedy, keď políčko v *i*-tom riadku a *j*-tom stĺpci zodpovedá pevnine. Dve políčka budeme považovať za susedné, ak zdieľajú spoločnú hranu; každé políčko tak má najviac štyri susedné políčka. *Ostrovom* nazveme maximálnu neprázdnu súvislú oblasť pevniny – dve políčka sú teda súčasťou jedného ostrova, ak obidve zodpovedajú pevnine a súčasne sú spojené postupnosťou pevninských políček, v ktorej každé nasledujúce políčko susedí s predchádzajúcim.

Napíšte triedu `Archipelago` v nepomenovanom balíku reprezentujúcu dynamickú mapu súostrovia a poskytujúcu:

- Konštruktor `public Archipelago(int rows, int columns)`, ktorý vytvorí mapu prázdneho súostrovia o `rows` riadkoch a `columns` stĺpcoch (riadky aj stĺpce budú indexované od nuly), v ktorej všetky políčka zodpovedajú moru. Môžete predpokladať, že `rows` aj `columns` sú *kladné* prirodzené čísla.
- Konštruktor `public Archipelago(boolean[][] map)`, ktorý vytvorí mapu súostrovia na základe dvojrozmerného poľa booleovských hodnôt `map`. Môžete predpokladať, že `map` je obdĺžnikové pole s *kladným* počtom riadkov aj stĺpcov. Pre všetky prípustné `i, j` je `map[i][j] == true` práve vtedy, keď políčko v *i*-tom riadku a *j*-tom stĺpci zodpovedá pevnine.
Neskoršie zmeny v dvojrozmernom poli použitom ako argument tohto konštruktora by nijako nemali ovplyvniť vnútorný stav už vytvorenej inštancie triedy `Archipelago`.
- Metódu `public boolean isLand(int row, int column)`, ktorá vráti `true` práve vtedy, keď políčko v `row`-tom riadku a `column`-tom stĺpci zodpovedá pevnine. Môžete predpokladať, že hodnoty `row` a `column` sú v rozsahu mapy súostrovia reprezentovanej príslušnou inštanciou.
- Metódu `public void setToLand(int row, int column)`, ktorá nastaví stav políčka v `row`-tom riadku a `column`-tom stĺpci na pevninu. Môžete predpokladať, že hodnoty `row` a `column` sú v rozsahu mapy súostrovia reprezentovanej príslušnou inštanciou.
- Metódu `public void setToSea(int row, int column)`, ktorá nastaví stav políčka v `row`-tom riadku a `column`-tom stĺpci na more. Môžete predpokladať, že hodnoty `row` a `column` sú v rozsahu mapy súostrovia reprezentovanej príslušnou inštanciou.
- Metódu `public int islandCount()`, ktorá vráti počet ostrovov na reprezentovanej mape súostrovia. Treba počítať s tým, že sa táto metóda bude volať aj viackrát, pričom medzi dvoma jej volaniami môže dôjsť aj k volaniam metód `setToLand` a `setToSea`.
- Metódu `public int largestIsland()`, ktorá vráti počet políček najväčšieho ostrova na reprezentovanej mape. V prípade, že na mape nie je žiaden ostrov, metóda vráti hodnotu 0. Opäť treba počítať s tým, že sa táto metóda bude volať aj viackrát, pričom medzi dvoma jej volaniami môže dôjsť aj k volaniam metód `setToLand` a `setToSea`.

Na testovač odovzdávajte súbor `Archipelago.java` obsahujúci zdrojový kód vašej triedy.