

Test č. 6, úloha č. 3

Pod (*vrcholovým*) k -farbením neorientovaného grafu, kde $k \geq 0$ je prirodzené číslo, rozumieme zobrazenie, ktoré každému vrcholu grafu priradí číslo – takzvanú *farbu* – z množiny $\{0, \dots, k-1\}$ tak, aby všetky hrany v grafe viedli medzi vrcholmi s rôznymi farbami. Pre graf s množinou vrcholov V teda ide o zobrazenie $f: V \rightarrow \{0, \dots, k-1\}$ také, že $f(u) \neq f(v)$ kedykoľvek v grafe vedie hrana z vrcholu u do vrcholu v . Pre grafy so slučkami špeciálne neexistuje žiadne k -farbenie bez ohľadu na k .

Chromatickým číslom neorientovaného grafu nazývame *najmenšie* prirodzené číslo $k \geq 0$ také, že existuje aspoň jedno k -farbenie tohto grafu. Za chromatické číslo grafov so slučkami, pre ktoré neexistujú žiadne farbenia, budeme pre účely tejto úlohy považovať `Integer.MAX_VALUE`; pre neorientované grafy bez slučiek je naopak zrejmé, že chromatické číslo nemôže byť väčšie, než počet vrcholov grafu.

Priložený archív obsahuje balík `graphs` s triedami pre grafy z prednášky a s kostrou triedy `ChromaticNumber`, ktorá má pre daný neorientovaný graf realizovať výpočet jeho chromatického čísla. V triede `ChromaticNumber` doprogramujte:

- Konštruktor `public ChromaticNumber(UndirectedGraph g)`, ktorý nájde chromatické číslo grafu g (pre grafy so slučkami bude touto hodnotou `Integer.MAX_VALUE`).
- Metódu `public int getChromaticNumber()`, ktorá vráti chromatické číslo vypočítané konštruktorom.

Na samotný výpočet chromatického čísla grafu použite prehľadávanie s návratom.

Na testovač odovzdávajte iba súbor `ChromaticNumber.java` obsahujúci zdrojový kód vami doplnenej triedy. Pri tvorbe kódu rešpektujte konvencie jazyka Java a elementárne zásady objektovo orientovaného programovania.

Príklad 1. Graf pozostávajúci z jedinej neorientovanej kružnice o piatich vrcholoch má chromatické číslo 3.

Príklad 2. Graf pozostávajúci z jedinej neorientovanej kružnice o šiestich vrcholoch má chromatické číslo 2.