Cvičenia č. 9, úloha č. 4

Neorientovaný graf G je *bipartitný*, ak jeho množinu vrcholov V možno rozložiť na dve disjunktné podmnožiny V_1, V_2 – tzn. $V_1 \cup V_2 = V$ a $V_1 \cap V_2 = \emptyset$ – tak, že každá hrana grafu spája vrchol z množiny V_1 s vrcholom z množiny V_2. To možno vyjadriť aj tak, že existuje *vrcholové* 2*-farbenie* grafu G – t. j. zobrazenie priraďujúce vrcholom „farby" z množiny $\{0, 1\}$ tak, že každá hrana grafu spája vrcholy s rozdielnou farbou. Vrcholy s farbou 0 tu napríklad môžu zodpovedať množine V_1 a vrcholy s farbou 1 množine V_2.

Priložený archív obsahuje balík `graphs` s triedami pre grafy z prednášky a s kostrou triedy `BipartiteGraphs`. Doprogramujte do triedy `BipartiteGraphs` telo statickej metódy `isBipartite`, ktorej argumentom je neorientovaný graf `g` a ktorá na výstupe vráti `true` práve vtedy, keď je graf `g` bipartitný. Môžete predpokladať, že `g != null`.
Na riešenie tejto úlohy možno použiť algoritmus, ktorý bude postupne všetkým vrcholom grafu priraďovať farby z množiny $\{0, 1\}$ tak, aby bola splnená podmienka vrcholového 2-farbenia. Každý komponent súvislosti možno napríklad prehľadávať do hĺbky alebo do šírky a susedom každého vrcholu `v` vždy priraďovať farbu rôznu od farby vrcholu `v`. Ak takéto priradenie farby niektorému vrcholu spôsobí porušenie podmienky vrcholového 2-farbenia, graf nemôže byť bipartitný. Ak sa naopak podarí ofarbiť všetky vrcholy grafu bez porušenia tejto podmienky, graf bipartitný je.
V prípade potreby môžete v triede `BipartiteGraphs` implementovať aj ďalšie pomocné metódy.

Na testovač odovzdávajte iba súbor `BipartiteGraphs.java` obsahujúci kód vašej triedy.