

Během zkoušky aktivně komunikujte se zkoušejícími, nebojte se zeptat. Nad zadáním přemýšlejte, často je těžší zjistit *co* řešit než *jak*. Kód pište čistě a průběžně ho vylepšujte. Nesnažte se vyřešit všechny příklady naráz, hodnotit budeme především podle kvality vašich myšlenek a vašeho kódu.

1. Naimplementujte metodu `int Node.distanceFrom(Node target)` vracující délku nejkratší cesty z `this` do `target`. Pokud žádná taková cesta neexistuje, vraťte `-1`.
2. Dopíšte třídu `GPS` implementující rozhraní `IteratorStrategy` tak, aby instance třídy `GraphIterator` ve spojení s instancí `GPS` postupně vracela uzly na nejkratší cestě z `initial` do `target`. Můžete předpokládat, že nějaká cesta vždy existuje.
3. Upravte váš kód tak, aby část implementující výběr uzlu ležícího na nejkratší cestě do cíle byla ve výsledném kódu naimplementována jen na jednom místě. Pozor: stále platí, že metoda `Node.distanceFrom` musí korektně fungovat i v případě, kdy žádná cesta mezi danými uzly neexistuje.
4. Do třídy `Node` dopíšte podporu pro návrhový vzor pozorovatel (observer/listener). Volání metody `Node.fireNodeEvent` by mělo způsobit doručení události `event` ve všech uzlech do vzdálenosti `intensity` od počátečního uzlu (avšak v každém uzlu jen jednou).
5. Dopíšte kód třídy `NodeBuilder` a kód konstruktoru třídy `Node`. Cílem je, aby si uživatel mohl vytvořit graf uzlů pomocí mutabilní třídy `Node` a pak ji jedním voláním `NodeBuilder.build` zkonvertoval do imutabilního grafu instancí `Node`. Relace tvořená atributy `neighbours` by měla být symetrická, tzn. pokud někdo ukazuje na souseda, pak soused ukazuje zpátky na něj. Nápověda: v průběhu stavění celého grafu si budete muset vést evidenci o tom, které instance třídy `NodeBuilder` už byly zkonvertovány do instancí třídy `Node` a které ne. Případné varování vývojového prostředí o předávání `this` ven z konstrukturu ignorujte.
6. Prakticky každý program v Javě je tvořen grafem objektů: uzly jsou jednotlivé objekty, hrany jsou tvořeny ukazateli uloženými v attributech objektů. Napište metodu `MetaBuilder.build`, která takový graf vytvoří a vrátí. Jména jednotlivých uzlů získajte pomocí metody `Object.toString`.