

Simulujte chladničku:

- Vytvorte triedu `Ingredient` (surovina). Surovina bude mať názov, počet kalórií (celé číslo) a atribút typu `boolean`, ktorý bude udávať, či je surovina vhodná pre vegetariánov. Počet kalórií by nemalo byť záporné číslo.
 - Definujte metódu `equals` (tým pádom treba predefinovať aj `hashCode`) tak, aby dve suroviny boli rovnaké, pokiaľ majú rovnaký názov.
 - Metóda `toString()` vráti názov suroviny.
-
- Vytvorte hlídanú výnimku `FridgeOverflowException`.
-
- Vytvorte triedu `Fridge`, ktorá si bude uchovávať kapacitu chladničky (t.j. maximálny počet surovín, ktoré sa do nej vlezú) a kolekciu surovín, ktoré aktuálne obsahuje (zvoľte vhodný typ: na poradí surovín v chladničke síce nezáleží, ale môžu sa opakovať, atď.). Kapacita by opäť nemala byť záporná. Žiadne ďalšie atribúty nepotrebuje.
 - Pri vytvorení je chladnička prázdna.
 - Chladnička bude mať metódu na pridávanie surovín, ktorá vezme surovinu a ak ešte nie je vyčerpaná kapacita, vloží ju do chladničky. Inak vyhodí `FridgeOverflowException`.
 - Chladnička bude mať metódu na odoberanie surovín, ktorá vráti `true` alebo `false` podľa toho, či daná surovina v chladničke bola, alebo nie.
 - Chladnička vie tiež vrátiť kolekciu (nemodifikovateľnú!) všetkých surovín, ktoré aktuálne obsahuje.
-
- Vytvorte triedu `Main` so statickou metódou `main`, kde si budete skúšať všetko, čo máte naimplementované. Skúste pridávať a odoberať suroviny, priebežne si vypisovať stav chladničky a výsledky volania metód, a sledujte, či sa zhoduje s tým, čo ste očakávali. Proste si poriadne precvičte prácu s kolekciami a zorientujte sa v ich dokumentácii.
-
- Vytvorte triedu `Recipe` (recept). Recept bude mať názov a kolekciu surovín, ktoré naň potrebujeme (opäť sa zamyslite a zvoľte vhodný typ - v recepte typicky neuvádzame viackrát tú istú surovinu, apod.). V konštruktore dostane recept názov aj kolekciu surovín, ktorú si "presype" do svojej kolekcie.
 - Recept vie povedať, či je vegetariánsky. Teda metóda `isVegetarian` vráti `true`, ak všetky suroviny v recepte sú vegetariánske; inak `false`.
 - Metóda `getCalories` vráti počet kalórií daného jedla, teda súčet kalórií všetkých surovín.
 - Recept vie tiež vrátiť (nemodifikovateľnú!) kolekciu všetkých potrebných surovín.

Upravte triedu Fridge:

- Metóda `canWeCookThis(Recipe recipe)` skontroluje, či z toho, čo obsahuje chladnička, vieme uvariť daný recept.
- Metóda `whichOfTheseCanWeCook` vezme kolekciu receptov a vráti tie, ktoré by sa dali uvariť z toho, čo máme v chladničke.
- Metóda `whichOfTheseCANNOTweCook` pre kolekciu receptov vráti tie, ktoré sa nedajú uvariť z toho, čo máme v chladničke. Uvedomte si, v akom vzťahu sú tieto dve metódy a využite to. Všeobecne sa (vždy) snažte čo najviac využívať existujúci kód a metódy poskytované samotnými kolekciami.
- Metóda `shoplistFor(Recipe recipe)` vráti zoznam surovín, ktoré treba dokúpiť (t.j. nie sú aktuálne v chladničke), ak chceme variť zadaný recept.
- Preťažaná metóda `shoplistFor` pre zadanú kolekciu receptov vráti celkový zoznam surovín, ktoré treba dokúpiť, ak by sme chceli variť všetky recepty.
- Metóda `shoplistByRecipe` vráti vlastne to isté, čo predchádzajúca metóda (t.j. nákupný zoznam pre kolekciu receptov), ale triedený podľa receptu. Vráti teda mapu, kde ako kľúče budú recepty, a ako hodnoty budú vždy kolekcie surovín, ktoré treba nakúpiť na daný recept. Nezháňajte recepty, na ktoré máme všetky suroviny doma.

Poznámka: Ak máme napríklad v chladničke tri brokolice a dostaneme kolekciu sto receptov, z ktorých každý potrebuje jednu brokolicu, táto metóda (a asi ani iné) to nemusí zisťovať – pretože na každý recept ZVLÁŠŤ máme dosť surovín. Ale kašlite na to, to je problém návrhu a nie implementácie; my to máme pre jednoduchosť navrhnuté takto. Prípadne si môžete samozrejme tú chladničku reprezentovať nejak inak, kým to bude rozumné :)

Rozšírenia, ak to máte hotové a ešte stále sa vám chce:

- Pridajte si prácu s množstvami. Surovina bude mať aj atribút "množstvo" (od jednotiek zase budeme abstrahovať a počítať s množstvom ako keby to boli balenia/kusy/...), kapacita chladničky teda bude obmedzovať celkový počet balení v chladničke. T.j. predtým bol stav plnej chladničky s kapacitou 5 napríklad: [brokolica, maslo, brokolica, špenát, rum]; pri návrhu s množstvami by bol stav plnej chladničky s kapacitou 5 napríklad: [{brokolica: 2 ks}, {rum: 3 ks}] – teda už nestačí kontrolovať počet prvkov v kolekcii. Tiež by sa zišlo porozmýšľať nad kolekciou surovín v chladničke – už tam nebudú opakujúce sa prvky, zmení to nejakú vašu reprezentáciu?
- Pridávanie a odoberanie surovín z chladničky teraz bude musieť zohľadňovať množstvá. Teda ak pridávam surovinu, ktorá tam ešte nie je, pridám ju s množstvom 1. Ak tam už taká surovina je, zvýším jej len množstvo. Podobne s odoberaním.
- Recept opäť zohľadní množstvá surovín pri výpočte počtu kalórií.
- Aj rozširujúce metódy chladničky na prácu s receptami budú zohľadňovať počty surovín v chladničke. Asi by bolo fajn predefinovať si kvôli nim `equals` na surovinách: dve suroviny budú rovnaké, ak majú rovnaké meno a aj množstvo. Potom sa bude dať napríklad jednoduchšie pýtať chladničky, či obsahuje suroviny na daný recept.
- ...a vymyslíte si samozrejme čokoľvek ďalšie.