

Završni ispit - Parcijalni ispit

Ukupno bodova: 20 (Bodovi će se dodijeliti proporcionalno broju uspješnih testova.)

Pozivnica za repozitorij: <https://classroom.github.com/a/SonV6wyh>

Na repozitoriju se nalazi gotov projekat koji sadrži samo praznu Main klasu i testove. Vaš zadatak je da napravite kompletan Java program koji zadovoljava postavku zadatka i prolazi testove.

Napomena: Sve metode zahtijevane ovim ispitom se mogu riješiti u jednoj liniji koda (za neke od njih je potrebno koristiti lambda funkcije i/ili streamove). Za svaku metodu koju budete riješili sa više od jedne linije koda biće oduzet 1 bod. Radi bolje čitljivosti, vi možete preći u sljedeću liniju tipkom enter, ali se u funkciji u pravilu treba nalaziti samo jedan znak tačka-zarez.

Zadatak 1:

Potrebno je kreirati klasu **Recept** koja sadrži podatke o nekom kuharskom receptu:

- Recept sadrži sljedeće podatke: nazivJela (String), kolekciju Sastojaka, te informacije o načinu pripreme.
- Sastojak može biti TecniSastojak ili PraskastiSastojak. Obje vrste sastojaka sadrže **naziv** (String) i **kolicina** (int), konstruktor sa ova dva parametra, te odgovarajuće settere i gettere. Razlika je u tome što kod TečnogSastojka količina je u decilitrima (dl), a kod PraškastogSastojka u gramima (g). Ovo treba iskoristiti prilikom ispisa sastojka na ekran, npr. ako probamo ispisati PraskastiSastojak brašno trebali bismo dobiti "100 g brašna" (pretpostavka je da je naziv u genitivu). Ne treba biti moguće deklarirati sastojak koji nije ni tečni ni praškasti.
- Informacije o načinu pripreme su **vrstaPripreme** i **podatak** (int). vrstaPripreme je pobrojanog tipa VrstaPripreme čije su moguće vrijednosti KUHANJE, PRZENJE i PECENJE. Značenje podatka ovisi od vrste pripreme, za pečenje to je temperatura pećnice u Celzijima, a za kuhanje i prženje su minute.
- Recept treba imati konstruktor koji prima nazivJela, te settere i gettere za sve attribute osim za kolekciju sastojaka za koju treba imati samo getter **getSastojci**.
- Pored toga Recept treba imati metodu **dodajSastojak** koja prima Sastojak. Ako sastojak sa istim nazivom i istog tipa (praškasti ili tečni) već postoji u kolekciji, treba povećati njegovu količinu za količinu primljenog sastojka, u suprotnom treba dodati novi sastojak.
- Konačno, Recept treba imati i metodu **izbaciSastojak** koja prima Sastojak. Ako sastojak sa istim nazivom i istog tipa već postoji u kolekciji, treba ga izbaciti iz kolekcije, u suprotnom treba baciti izuzetak tipa NoSuchSastojakException sa tekstom "Nepoznat sastojak ime" (umjesto ime uvrstiti naziv sastojka).

- Prilikom ispisa Recepta na ekran, najprije se ispisuje tekst "Recept za naziv jela", zatim u svakom narednom redu po jedan sastojak, te na kraju vrsta pripreme što može biti "Peći na X stepeni", "Kuhati X minuta" ili "Pržiti X minuta" (X je podatak).

Zadatak 2:

Pored toga napravite i klasu Kuhar koja drži kolekciju Receptata. Pretpostavka je da se u kuharu neće nalaziti dva recepta za isto jelo (sa istim nazivom jela). Klasa Kuhar treba da sadrži sljedeće metode:

- **dodajRecept** prima Recept i stavlja ga u kolekciju.
- **brojReceptata** vraća broj receptata trenutno pohranjenih u kuharu.
- **dajRecept** prima String koji predstavlja naziv jela, a vraća recept sa tim jelom ili **null** ako takav recept ne postoji. Ova metoda treba biti implementirana na najefikasniji mogući način, u suprotnom će biti oduzeto 2 boda.
- Metoda **receptiSaSastojkom** prima Sastojak a vraća ArrayList kolekciju receptata u kojima se nalazi sastojak sa istim nazivom (ne mora biti istog tipa!)
- Metoda **sviSastojci** vraća skup (Set) svih sastojaka koji se pojavljuju u svim jelima, sortiranih po količini. Obratite pažnju da ako se isti sastojak pojavljuje sa raznim količinama (brašno 100g, brašno 50g...) treba se svaki od ovih primjeraka pojavljivati u skupu Set.
- Metoda **filtriraj** vraća ArrayList kolekciju receptata koji zadovoljavaju uslov dat predikatskom lambda funkcijom koja je parametar ove metode.

Sarajevo, 2. 2. 2021

Vedran Zuborić