

Zadaća 1

Obratite pažnju na ispravke na dnu dokumenta

Zadaća nosi 2 boda. Rok za slanje rješenja je 15. 11. 2020. u 23:59. Za izradu zadatke potrebno je da prihvatite sljedeću GitHub Classroom pozivnicu koristeći GitHub account koji ste ranije kreirali i unijeli u [Google Form za GitHub accounts](https://classroom.github.com/a/pxmfGjw1):

<https://classroom.github.com/a/pxmfGjw1>

Program koji ćete napraviti treba da omogući korisniku da kreira kvizove koji se onda mogu rješavati u obliku konzolne aplikacije.

Za potrebe ovoga kreirajte klasu **Kviz**. Kviz posjeduje **naziv**, **pitanja** (kolekcija pitanja u kvizu) i sistem bodovanja (**sistemBodovanja**). Za ovu klasu napravite:

- konstruktor koji prima naziv i sistemBodovanja i postavlja ih, dok pitanja postavlja na praznu kolekciju.
- odgovarajuće getere i setere.
- metodu **dodajPitanje** koja prima pitanje i dodaje ga u kolekciju pitanja u kvizu. Ukoliko se u kvizu već nalazi pitanje sa istim tekstom treba baciti izuzetak tipa `IllegalArgumentException` sa porukom “Ne možete dodati pitanje sa tekstom koji već postoji” (tekstovi pitanja “Koliko je sati?” i “koliko je Sati?” su isti)
- odgovarajuću **toString** metodu koja vraća ispisan kviz sa označenim tačnim odgovorima. Format ispisa je sljedeći:

Kviz “Naziv kviza” boduje se binarno/parcijalno/pacijalno sa negativnim bodovima.

Pitanja:

1. Tekst prvog pitanja?(3.0b)
 - a: tacan odgovor(T)
 - b: netacan odgovor
 - c: opet tacan odgovor(T)
2. Tekst drugog pitanja?(2.5b)
 - a: netacan odgovor
 - b: netacan odgovor

c: tacan odgovor(T)

Napomena 1: Ukoliko nije jasan format ispisa (nije jasno je li novi red, tab i sl.) provjerite u testovima - kako je u testovima tako treba da bude kod vas.

Napomena 2: Slova a, b, c su varijabilna i njihovo značenje je objašnjeno u nastavku

- metodu **predajKviz** koja će biti objašnjena na kraju

Sistem bodovanja (**SistemBodovanja**) je pobrojani tip (enum) koji ima sljedeće konstante: BINARNO, PARCIJALNO, PARCIJALNO_SA_NEGATIVNIM od kojih ovisi način bodovanja na kvizu što će biti objašnjeno u nastavku.

Pitanja su kolekcija objekata tipa **Pitanje**. Pitanja imaju **tekst**, broj poena (**brojPoena**) i kolekciju odgovora (**odgovori**). Svaki odgovor ima svoj jedinstveni id unutar pitanja (npr. slovo a, b, c). Odgovori su tipa **Odgovor**. Svaki odgovor ima tekst (**tekstOdgovora**), i oznaku da li je tačan (**tacno**).

U klasi Pitanje implementirajte:

- konstruktor koji prima tekst i brojPoena i postavlja ih a kolekciju odgovora inicijalizira na praznu
- odgovarajuće getere i setere
- metodu **dodajOdgovor** koja prima **id** (String), **tekst**, i boolean **tacno** i dodaje odgovor u kolekciju odgovora koji je opisan podacima proslijeđenim u metodu
- metodu **obrisiOdgovor** koja prima id odgovora koji se želi obrisati. Ukoliko odgovor s tim id-em ne postoji treba se baciti izuzetak tipa `IllegalArgumentException` sa porukom "Id odgovora mora biti jedinstven"
- metodu **dajListuOdgovora** koja vraća kolekciju odgovora u obliku liste
- odgovarajuću **toString** metodu koja vraća pitanje u formatu pogodnom za ispis korisniku koji rješava kviz (bez označenog tačnog odgovora):

Tekst prvog pitanja?(3.0b)

a: tacan odgovor

b: netacan odgovor

c: opet tacan odgovor

- metodu **izracunajPoene** koja prima listu Stringova (id-evi odgovora) koje je korisnik za to pitanje označio kao tačne i sistem bodovanja po kojem treba izračunati poene. Metoda treba da provjeri validnost poslanih zaokruženih odgovora. Ukoliko se u listi nalazi nepostojeći id odgovora treba baciti izuzetak sa porukom "Odabran je nepostojeći odgovor". Ukoliko se u listi nalazi dupliciran id odgovora treba baciti izuzetak sa porukom "Postoje duplikati među odabranim

odgovorima". Oba izuzetka trebaju biti tipa `IllegalArgumentException`. Nakon što se utvrdi validnost liste odgovora, metoda treba da izračuna koliko poena bi korisnik dobio ako zaokruži te odgovore u ovisnosti od sistema bodovanja.

Ukoliko je sistem bodovanja `BINARNO` korisnik koji zaokruži sve tačne odgovore dobit će onoliko poena koliko nosi samo pitanje. Ukoliko nisu zaokruženi svi tačni odgovori ili ukoliko se među zaokruženim odgovorima nalazi bar jedan netačan, pitanje se boduje sa 0.

Ukoliko je sistem bodovanja `PARCIJALNO` korisnik koji zaokruži sve tačne odgovore dobit će onoliko poena koliko nosi samo pitanje. Korisnik koji među zaokruženim odgovorima zaokruži bar jedan netačan odgovor dobija 0 poena, a korisnik koji zaokruži samo tačne, ali ne sve tačne odgovore dobija bodove proporcionalno ($\text{ukupno} / \text{broj odgovora}$) * broj tačno odgovorenenih

Ukoliko je sistem bodovanja `PARCIJALNO_SA_NEGATIVNIM` korisnik koji zaokruži sve tačne odgovore dobit će onoliko poena koliko nosi samo pitanje. Korisnik koji među zaokruženim odgovorima zaokruži bar jedan netačan odgovor dobija negativne poene (i to u vrijednosti polovine od ukupnog broja bodova), a korisnik koji zaokruži samo tačne, ali ne sve tačne odgovore dobija bodove proporcionalno ($\text{ukupno} / \text{broj odgovora}$) * broj tačno odgovorenenih.

U sva tri slučaja, korisnik koji ne zaokruži ništa dobija 0 poena.

U klasi `Odgovor` implementirajte:

- konstruktor sa oba parametra
- odgovarajuće getere i setere

Za potrebe implementacije metode **predajKviz** iz klase `Kviz`, implementirajte klasu **RezultatKviza**. Ova klasa ima attribute **kviz** (na koji se rezultati odnose), **total** (double, ukupno poena ostvarenih na tom kvizu) i atribut **bodovi** koji je tipa `Map<Pitanje, Double>` i predstavlja raspored bodova po pojedinačnim pitanjima kviza. Za ovu klasu napravite:

- konstruktor koji prima kviz a total postavlja na 0.
- odgovarajuće getere i setere
- odgovarajuću **toString** metodu koja vraća rezultate kviza u sljedećem formatu:

Na kvizu "Naziv kviza" ostvarili ste ukupno 5.5 poena. Raspored po pitanjima:

Tekst pitanja - 2.0b

Tekst pitanja - 0.0b

Tekst pitanja - 1.5b

Metoda **predajKviz** u klasi Kviz prima parametar tipa `Map<Pitanje, ArrayList<String>>` koji govori za koje pitanje je korisnik označio koje odgovore kao tačne. Ova metoda treba da korištenjem metode **izracunajPoene** iz klase Pitanje izračuna poene korisnika koji je predao kviz i iste vrati kao objekat klase **RezultatKviza**. Rezultat kviza koji se vrati treba da sadrži sva pitanja u kvizu bez obzira da li je korisnik odgovorio na sva. Na ona na koja nije odgovorio dobija 0 bodova.

Konačno, napišite **main** funkciju koja omogućava igranje kviza. U klasi **Main** obavezno koristite statičku metodu **igrajKviz** koja prima kviz koji će korisnik igrati. Napravite dijalog u kojem se ispisuju pitanje po pitanje i korisnik na iste odgovara unošenjem id-eva odgovora koje smatra da su tačni u nekom formatu (npr. a,b,d). Omogućite da korisnik ne odgovori ništa. Po završetku kviza ispišite rezultate i omogućite korisniku da pregleda tačne odgovore. Main će biti ručno pregledan i nosit će odgovarajući broj bodova.

Napomena: za objašnjenja o mogućim rubnim slučajevima prvo provjerite testove. Metoda `igrajKviz` će biti bodovana ručnim pregledanjem i nositi će 0,1 bodova, dok će preostalih 1,9 bodova biti dodijeljeno na osnovu testova.

Izmjene:

- 1. 11. 2020 21:40 - u testovima klase Pitanje došlo je do neusaglašenosti sa tekstom zadatka. Testovi su popravljani, međutim studenti koji su prihvatili pozivnicu prije ove popravke imaju i dalje staru verziju testova na repozitoriju pa ih moraju popraviti ručno: u klasi PitanjeTest, metoda `testIzracunajPoene7 poeni3` i `poeni4` trebaju biti 0, također u metodi `testIzracunajPoene8 poeni3` trebaju biti 0.
- 2. 11. 2020 18:30 - pojašnjen tekst da se u klasi Kviz nalazi metoda `predajKviz`, a ne klasa `RezultatKviza`