

Objektno-orijentisano programiranje – prvi kolokvijum

Zadatak: RAFChatMessages – prvi termin - grupa 1

Napisati Java konzolnu aplikaciju koja simulira slanje poruka.

Prvi deo – implementacija UML dijagrama klasa

1. (5p) implementirati sve elemente na priloženom dijagramu uz sledeće napomene:

- za sva polja dodati set/get metode sa proverama koje se traže u zadatku (**osim u klasi Emojis**),
- metode iz klase Object, equals i toString nadjačati prema proceni na mestima gde je to neophodno za realizaciju zahtevanih funkcionalnosti
- proizvoljno se mogu dodati konstruktori, metode kojih nema na dijagramu i implementacije ugrađenih interfejsa (na primer java.lang.Comparable)

Primer emoji-ja:

naziv: happy
skracenica: :)

2. (2p) Klasa *Emojis* treba da bude odrađena po singleton šablonu, što znači da će postojati samo jedna instanca te klase u svakom trenutku. Da bi to bilo omogućeno, potrebno je u metodi *getEmojis* pitati da li *instance* već postoji, ako ne postoji instancirati je pomoću odgovarajućeg konstruktora i onda vratiti tu instancu.

- (1.5p) metoda *dodajEmoji(naziv,skracenica)* dodaje emoji u listu ako ne postoji emoji istog naziva ili iste skraćenice. Ukoliko je emoji dodat vraća *true*, u suprotnom *false*.

3. U modelu su predstavljene različite vrste poruka sa različitim načinima formiranja sadržine, odnosno implementacije metode *formirajSadrzinu()*.

- (0.5p) u klasi *TekstualnaPoruka* opisana metoda vraća *tekst*,

- (0.5p) u klasi *EmojiPoruka* opisana metoda vraća *skracenicu emoji-ja*,

- (1p) u klasi *KombinovanaPoruka* vraća se tekst dobijen od *sadrzine* poruke (koju čine druge poruke). U tom tekstu nalaze se sve poruke iz *sadrzine*, razdvojene razmakom (spejs).

4. Metoda *dodajUSadrzinu* u klasi *KombinovanaPoruka* dodaje *poruku* u *sadrzinu* na sledeći način:

- (0.5p) ukoliko *poruka* **nije** *KombinovanaPoruka*, dodaje se samo *poruka* bez dodatnih operacija.

- (1p) ukoliko je *poruka* tipa *KombinovanaPoruka*, sve poruke koje čine njenu *sadrzinu* se posebno dodaju.

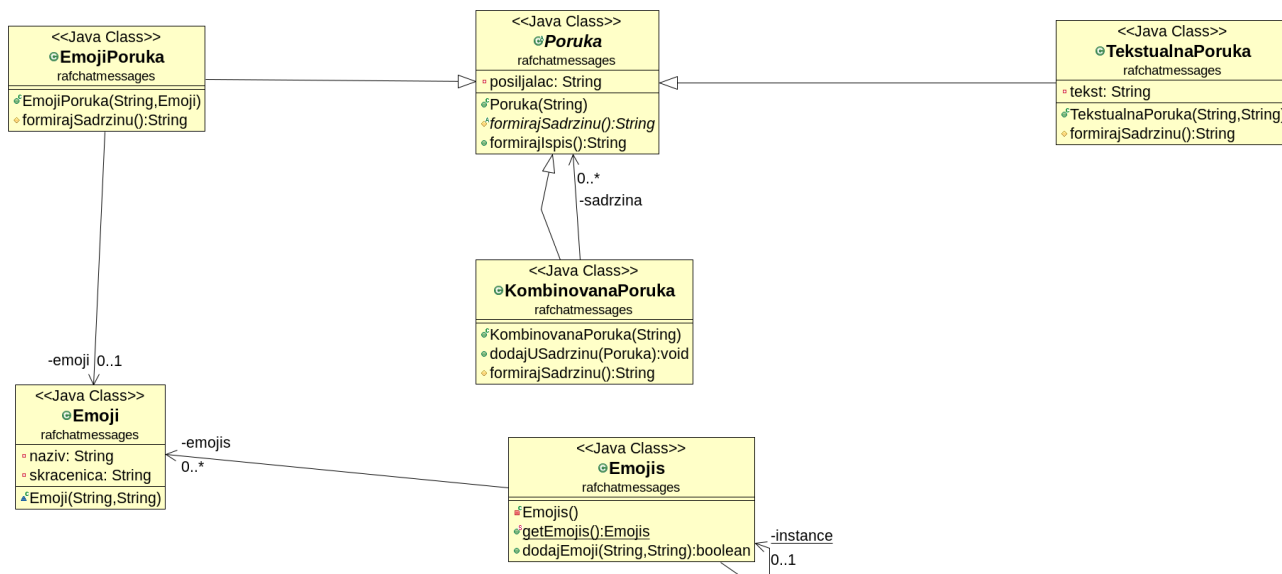
- (0.5p) poruka (kog god tipa) se **ne može dodati** u *KombinovanuPoruku* ako nemaju istog pošiljaoca.

5. Metoda *formirajIspis* klase *Poruka* treba da vrati string koji će zapravo biti ispisan prilikom prikazivanja poruke.

- (0.5p) U prvom redu ispisa treba da stoji ko je pošiljalac poruke, nakon koga sledi jedan prazan red.

- (1p) U ostalim redovima ispisa treba da stoji sadržina poruke. Ona može biti dobijena pozivom odgovarajuće metode. Između svaka dva reda sadržine (delovi sadržine poruke koji su odvojeni znakom za novi red) treba da stoji jedan prazan red.

- (1p) Obezbediti da svaki red ispisa sadržine bude započet tabom.



Drugi deo – proširenje modela i dodatne funkcionalnosti

8. (3p) Dodati novu vrstu poruke – poruku sa više emoji-ja i bez teksta. Prilikom formiranja sadržine ove poruke potrebno je za svaki od emoji-ja izvući skraćenicu, pa sve te skraćenicu razdvojiti po jednim razmakom. Obezbediti operaciju koja dodaje jedan ili više emoji-ja u sadržinu poruke. Prilikom dodavanja ove poruke u kombinovanu poruku, potrebno je dodati sve emoji-je pojedinačno u okviru poruke kojoj je pošiljalac pošiljalac poruke koja se dodaje. **Mala pomoć:** možete u klasi *Emojis* dodati getter za listu emoji-ja, ali ga ne možete nazvati *getEmojis* – nazovite ga nekako drugačije.

9. (4p) Omogućiti da se poruke mogu slati u okviru četa. Čet treba da sadrži listu svih poruka koje su ikada poslate preko njega, kao i listu svih pošiljalaca koji su bar jednom poslali poruku. Obezbediti operaciju za dodavanje poruke tako da ona istovremeno dodaje i pošiljaoca ako on još uvek nije registrovan na tom četu.

10. (5p) Napraviti klasu *ChatApp* koja sadrži *main* metodu kojom će se testirati zadati model. Potrebno je u njoj kreirati bar pet različitih emoji-ja. Emoji-je kreirati korišćenjem odgovarajuće metode u klasi *Emojis*. Treba napraviti da barem jedan od ovih pet emoji-ja ne može da se kreira. Zatim kreirati bar dve od svake vrste poruka. Ukoliko je implementiran deo u zadatku 8, obezbediti čuvanje tih poruka i njihovih pošiljalaca u četu. Takođe obezbediti da bar dve poruke imaju istog pošiljaoca. Na kraju ispisati sve poruke iz četa.

11. (3p) Obezbediti čuvanje emoji-ja u tekstualnom fajlu. Za svaki emoji čuvaju se i naziv i skraćenicu. Emoji-ji treba da budu sortirani po nazivu.