

Menjaža

Treći zadatak iz Jave

Zadatak

Napraviti Java aplikaciju koja će njenim korisnicima omogućiti bržu i efikasniju razmenu sličica. Korisnik koji se prijavi na sistem ima mogućnost pregleda mogućih razmena sa ostalim korisnicima aplikacije, kao i evidentiranje sličica koje poseduje kao duplikate i sličica koje mu nedostaju da popuni album.

Pravila

Smatraćemo da naš album ima 99 sličica, radi pojednostavljenja. Takođe, smatraćemo da svi korisnici koji se uključe u menjažu menjaju sličice iz istog albuma (nećemo omogućiti postojanje više različitih albuma u sistemu). Na početku igre, da bismo simulirali realnu situaciju, svakom igraču se na slučajan način dodeljuju duplikati (broj duplikata kao i same sličice- npr 7 duplikata, brojevi 1, 12, 34, 45, 56, 78, 90) i sličice koje mu trebaju (slično kao i duplikati, npr 6 sličica, brojevi 3, 15, 22, 26, 31, 88). Dok na slučajan način generišete ove brojeve vodite računa o tome da kod igrača ne može ista sličica biti i duplikat i sličica koja mu treba, istovremeno.

Detaljniji opis aplikacije

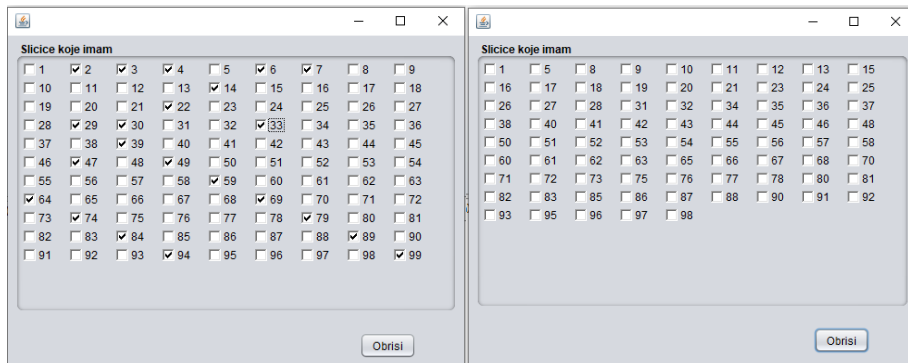
Aplikacija treba da se sastoji iz dva dela:

- Serverski deo aplikacije (konzolna aplikacija)
- Klijentski deo aplikacije (Java GUI aplikacija)

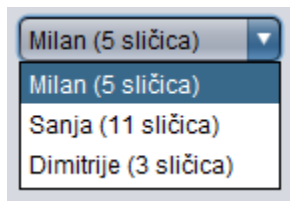
Serverski deo aplikacije je zadužen za registrovanje korisnika koji se prijavljuju u sistemu kao i pružanje informacija u vezi sa mogućim razmenama svakom od korisnika na upit.

Klijentski deo aplikacije treba da omogući korisniku

- da kroz grafički korisnički interfejs (GUI) unese svoje korisničko ime po kojem će biti prepoznat u sistemu;
- da vodi evidenciju o sličicama koje su mu duplikati i sličicama koje mu nedostaju da bi popunio album tako što odgovarajućim GUI kontrolama dodaje/uklanja sličice iz liste. Za svaku sličicu koja je duplikat (kao i za one koje korisnik potražuje) postoji odgovarajući *CheckBox* (pogledajte sliku dole za primer duplikata, na sličan način treba evidentirati i sličice koje se potražuju). Kada korisnik razmeni neku sličicu treba da selektuje odgovarajući *CheckBox* u duplikatima i u sličicama koje mu trebaju, nakon čega se te dve uklanjaju iz liste (npr. menjao je sličicu broj 10 za sličicu broj 23 – to znači da mu sličica 10 više nije duplikat, ali ni sličica broj 23 nije više ona koja mu treba). Da bismo uprostiti stvari, nećemo dozvoliti unos više istih sličica kao duplikata (ili onih koje korisniku trebaju) što se u realnosti možda može desiti;



- da zatraži od servera informacije o mogućim razmenama nakon čega mu server vraća informacije o mogućim razmenama - ako su moguće razmene sa više korisnika, treba nekako prikazati sa kojim korisnikom su moguće razmene, na primer pomoću *ComboBox*-a (eventualno uz informaciju koliko sličina je moguće menjati);



- Kada se zatraže informacije o potencijalnim razmenama, za svakog od korisnika koji je registrovan u sistemu i sa kojim mogu da se menjaju sličice (obratite pažnju da dva korisnika ne moraju nužno imati isti broj sličina raspoloživih), treba prikazati poruku formata:

Možeš da menjaš sličice sa korisnikom Milan.

Ti imaš za njega sličice 12, 35, 46, 47, 53, 86, 92, a on za tebe sličice 9, 14, 22, 41, 62, 63, 97, 99

- Ako korisnik odabere da želi da menja sličice sa nekim drugim korisnikom, pitanje mora da se postavi i tom drugom korisniku, pa tek ako i on pristane razmena se dešava tako što se sa liste duplikata i sličina koje nedostaju uklanjaju zamenjene sličice: npr ako su Dušan i Milan menjali 3, 17, 45 (Dušan) i 36, 56, 91 (Milan), sličice 3, 17, 45 treba skinuti sa Dušanove liste duplikata, a sa Milanove liste sličina koje mu nedostaju, dok se, na sličan način, sličice 36, 56 i 91 skidaju sa Milanove liste duplikata, a sa Dušanove liste sličina koje mu nedostaju;
- U jednostavnijoj implementaciji menjaju se sve sličice, pri čemu ne mora broj sličina na obe strane biti isti, dok u drugoj, složenijoj implementaciji treba omogućiti korisniku koji ima više sličina za razmenu da odabere koje želi da menja sa drugim korisnikom, pa tek nakon toga treba poslati poruku drugom korisniku o eventualnoj razmeni;

Napomene

- Java GUI aplikacije se drugačije kreiraju u odnosu na Java konzolne aplikacije. Tipično, kreira se klasa koja nasleđuje od klase *JFrame*, a u NetBeans-u u tom slučaju postoje dva moda rada: *Source mod* u kojem se edituje izvorni kod, i *Design mod* u kojem se kroz grafički korisnički interfejs komponente slažu u *JFrame* (Prikažaću na vežbama kako se prave GUI aplikacije i koje Java klase se koriste u tom slučaju);

- *CheckBox*-eve treba generisati dinamički (u kodu), jer će biti izuzetno zahtevno ako moramo statički deklarirati direktno referencu na svaki od *CheckBox* objekata (jer ih ima 99!!):

```
CheckBox cb1 = new CheckBox("1");

CheckBox cb2 = new CheckBox("2");

...

CheckBox cb99 = new CheckBox("99");
```

Verovatno najelegantniji način da se ovo uradi je korišćenjem *HashMap* klase o kojoj smo pričali ranije:

```
HashMap<String, CheckBox> duplicates;//atribut naslednice JFrame

void initializeCheckBoxes() {

    int num = 1;

    while (num < 100) {

        Checkbox cb = new Checkbox(num + "");

        duplicates.put("Btn" + num, cb);

        num++;

    }

}
```

- Samo dodavanje *CheckBox*-a u kolekciju (*duplicates*) neće prikazati taj *CheckBox* na monitoru. Da bi se on video u glavnom prozoru, potrebno ga je dodati na panel (npr panel koji se zove *cbPanel* i prethodno je statički postavljen u okviru *JFrame*-a) sa:

```
cbPanel.add(cb);
```

A kada ga treba ukloniti, to radimo sa:

```
cbPanel.remove(cb);
```

Nakon dodavanja/uklanjanja komponente, ukoliko se promene ne vide na prozoru, potrebno je pozvati i sledeće metode

```
cbPanel.revalidate();
```

```
cbPanel.repaint();
```

- Ako dinamički postavljamo *CheckBox*-eve moramo voditi računa o dimenzijama panela na koji ih postavljamo, kao i o dimenzijama samog *CheckBox*-a. Dimenzije *CheckBox*-a postavljamo sa metodom (*x* i *y* se odnose na koordinate gornjeg levog ugla *CheckBox*-a, a *width/height* na širinu/visinu):

```
cb.setBounds(x, y, width, height);
```

Poziciju i dimenzije Panel-a dobijamo metodom:

```
cbPanel.getLocation() (x koordinata sa cbPanel.getLocation().x, y koordinata sa cbPanel.getLocation().y)
```

Dok dimenzije panela dobijamo sa:

```
cbPanel.getWidth() odnosno cbPanel.getHeight()
```

- Kada nam kasnije treba neki konkretan *CheckBox*, npr. *CheckBox* koji odgovara sličici 5 dobićemo ga sa

```
Checkbox cb = duplicates.get("Btn" + 5);
```

Ako želimo da proverimo da li postoji sličica sa brojem 7 u duplikatima, to možemo uraditi na sledeći način:

```
String key = "Btn" + 7;  
  
If (duplicates.containsKey(key)) {  
    CheckBox cb = duplicates.get(key);  
    ... //ovde raditi sa tim CheckBox-om šta treba  
}
```

Slično, sve sličice duplikate u datom trenutku možemo dobiti preko odgovarajućeg skupa ključeva (u ovom primeru još proveravam koja od njih je selektovana da bih je uklonio):

```
for (String key : duplicates.keySet()) {  
    //ako je CheckBox selektovan obriši ga  
    if (checkboxes.get(key).getState()) {  
        //ukloni taj CheckBox sa panela  
    }  
}
```

Ocenjivanje

Zadatak nosi ukupno 20 poena koji će biti raspoređeni na sledeći način:

- Komunikacija između klijenta i servera (registrovanje novog korisnika, slanje informacija o mogućim razmenama, slanje zahteva za razmenom korisnika) - **4p**
- Inicijalno određivanje duplikata i sličica koje nedostaju na slučajan način za svakog korisnika - **2p**

- Dinamičko prikazivanje duplikata i sličica koje nedostaju u okviru zasebnih panela (da svi *CheckBox*-evi stanu u panel u skladu sa proračunatim pozicijama u skladu sa veličinom panela - smatraćemo da će panel biti dovoljno velik da smesti svih 99 sličica ako je potrebno) - **2p**
- Ažuriranje liste duplikata i sličica koje nedostaju kroz grafički korisnički interfejs: omogućeno brisanje sličica koje su zamenjene, kako sa liste duplikata, tako i sa liste sličica koje nedostaju selektovanjem sličica i pritiskom na dugme "Obrisi" - **3p**
- Dobijanje i prikaz mogućih razmena sa ostalim korisnicima kao i prihvatanje jedne od ponuđenih razmena pritiskom na dugme "Moguće razmene" - **3p**
- Razmena i ažuriranje liste duplikata i sličica koje nedostaju nakon što je prihvaćena razmena sa nekim od korisnika (obostrano prihvaćena) - **4p**
- Omogućavanje korisniku da odabere sličice koje želi da menja u slučaju da on ima više sličica za razmenu od drugog korisnika sa kojim se menja (npr 10 prema 7, kada treba odabrati 7 sličica od 10 za razmenu). Obavezno proveriti da li je nakon odabira jednak broj sličica za menjanje sa obe strane, pre nego što se informacija prosledi serverskoj aplikaciji - **2p**