Zavod za elektroniku, mikroelektroniku računalne i inteligentne sustave

3. srpnja 2019.

Oblikovni obrasci u programiranju

završni ispit

Napomena uz sve zadatke: povežite vaša rješenja s odgovarajućim **oblikovnim obrascima** i **načelima oblikovanja**; izvorni kôd možete skicirati u C-u, C++-u, C#-u, Javi ili Pythonu ako nije drukčije zadano.

1. Operacijski sustav CoolOS nudi mogućnost nadziranja promjena datotečnog sustava primjenom sučelja sa slijedećim metodama:

```
int coolos_register_watcher(String path); //returns watcher handle
void coolos_cancel_watcher(int handle);
bool coolos_get_event(int handle, bool blocking, Pathevent pevent);
```

   Funkcija coolos_register_watcher započinje nadziranje zadanog kazala. Funkcija coolos_cancel_watcher otkazuje nadziranje prema zadanom ključu. Funkcija coolos_get_event dohvaća sljedeći neobrađeni događaj. Ako je poziv blokirajući (blocking=**true**), povratna vrijednost je uvijek **true**. Ako poziv nije blokirajući te nema neobrađenih događaja funkcija vraća **false**, a *pevent ostaje nepromijenjen.

   Zadatci:

   (a) Oblikujte i implementirajte razred PathWatcher koji klijentima omogućava definiranje nadziranog kazala te blokirajući i neblokirajući dohvat promjena korištenjem gore navedenog programskog sučelja. Vaš razred treba podržavati sljedeći primjer upotrebe:

```
PathWatcher pw = new PathWatcher("/home/perica/dokumenti");
while (PathEvent ev = pw.getNextBlocking()) {
   System.out.println("Promjena: " + ev.fileName + " - vrsta " + ev.changeType);
}
```

   (b) Osigurajte prenosivost razreda PathWatcher na druge operacijske sustave (Windows, Linux, itd). Pretpostavite da drugi operacijski sustavi imaju slična programska sučelja kao i CoolOS.

   (c) Omogućite provođenje nadzora neovisno o klijentskom kodu korištenjem blokirajuće petlje čekalice u zasebnoj dretvi. Pri tome klijenti trebaju moći dinamički definirati zadatke koji se izvršavaju nakon promjene direktorija.

   (d) Nacrtajte dijagram razreda sveukupnog rješenja. Navedite prikladni oblikovni obrazac (ili obrasce) te navedite oblikovna načela s kojima su usklađeni.

2. Razmatramo izraze koji se sastoje od cijelih brojeva, operatora $+$ i $\cdot$ i zagrada. Potrebno je podržati izračunavanje izraza

   - prema normalnim računskim pravilima,
   - po modulu n, te
   - prema "konkatenacijskom" računu, gdje je $2 \cdot 3 = 33$, a $2 + 3 = 23$.

   Na raspolaganju je funkcija parse koja prima izraz u tekstnom obliku, a vraća stablo (korijenski čvor) u kojem listovi predstavljaju vrijednosti, a drugi čvorovi operatore (mogu biti tipa Plus ili Dot).

   a) (4 bodova) Oblikujte rješenje (i skicirajte strukturni dijagram) koje podržava predstavljanje i izračunavanje izraza prema svakoj od 3 navedene konvencije (očekuje se da će takvih konvencija biti još).

   b) (2 boda) Navedite prikladni oblikovni obrazac (ili obrasce) te navedite oblikovna načela s kojima su usklađeni.

   c) (2 boda) Povežite sudionike oblikovnog obrasca (ili obrazaca) s rješenjem.

   d) (2 boda) Skicirajte dijelove programskog koda koji su bitni za različite vrste izračunavanja izraza i za mehaniku obrazaca. Od načina izračunavanja izraza, dovoljno je skicirati kod za jedan od njih.

3. Vim je program koji podržava korisničke makro-naredbe za uređivanje teksta. Primjerice, makro-naredbu za brisanje posljednja tri znaka u tekućem retku te prijelaz u slijedeći redak možemo definirati konfiguracijskom naredbom :map q $2hd$j. Makro-naredba q sastavljena je od 4 elementarne naredbe za uređivanje teksta:

   i) pomicanje kursora do kraja retka ($),
   ii) pomicanje kursora dva znaka lijevo (2h),
   iii) brisanje teksta do kraja retka (d$)
   iv) te pomicanje kursora za jedno mjesto dolje (j).

   Makro naredbe se trebaju pozivati, opozivati i ponovno pozivati atomski, kao jedinstvena naredba za uređivanje teksta.

   Zadatci:

   (a) napisati razred Macro koji implementira funkcionalnost makronaredbi uključujući i kompatibilnost s razredom UndoManager koji omogućava pozivanje i opozivanje naredbi za uređivanje teksta;
   (b) nacrtati strukturni dijagram koji uključuje razrede Macro, UndoManager, MoveCursor i važne apstraktne razrede;
   (c) identificirati obrasce te navesti oblikovna načela koja vaša organizacija podržava.

   Uputa: Pretpostavite da razred UndoManager ima sljedeće sučelje (nije ga potrebno implementirati):

```
class UndoManager{
public:
   void do(EditAction *pcmd);
   void undo();
   void redo();
 }
```

4. Razmatramo program za vizualno oblikovanje koji korisnicima omogućava izradu crteža sastavljenog od grafičkih oblika (pravokutnika, trokuta, krugova itd.). Glavni prozor programa ima alatnu traku koja sadrži gumbe s imenima dostupnih grafičkih oblika. Naš fokus je nadogradnja koja bi omogućila automatsko povezivanje glavnog programa s novim konkretnim grafičkim oblicima.

   Potrebno je oblikovati i implementirati sljedeće komponente (skicirati kod).

   - Funkciju za automatsko popunjavanje alatne trake konkretnim grafičkim oblicima koje se nalaze u datotekama kazala plugins/. Datoteke sadrže bytecode ako pišete u Javi ili Pythonu, odnosno binarni strojni kod ako pišete u C-u ili C++-u. Funkcija bi se pozivala prilikom inicijalizacije programa. Pretpostavite da svaka datoteka sadrži implementaciju jednog grafičkog oblika. Novi oblici se moraju moći dodavati bez potrebe za ponovnim prevođenjem ostatka programa.
   - Funkciju za dodavanje novih objekata u crtež odabirom grafičkog oblika i klikom na platno.

   Skicirajte dijagram razreda i istaknite oblikovne obrasce koje ste koristili. Pojasnite koje sve korake moramo provesti prilikom dodavanja novog grafičkog oblika.

5. Razmatramo robotski sustav koji se sastoji od triju jedinica: upravljačke, senzorske i aktuatorske. Upravljačka jedinica očitava vrijednosti senzora pozivanjem metode SensorUnit.read i izdaje naredbe aktuatorskoj jedinici pozivanjem metode ActuatorUnit.acceptCommand.

   Radi analize ponašanja robota treba omogućiti bilježenje senzorskih vrijednosti i naredbi bez izmjene gotovih komponenata i utjecaja na ponašanje robota.

   a) (4 boda) Predložite oblikovno rješenje prikazanog problema te ga ilustrirajte strukturnim dijagramom. Procijenite usklađenost rješenja s načelima oblikovanja.
   b) (3 boda) Skicirajte implementaciju razreda koji presreće i zapisuje očitavane vrijednosti i naredbe.
   c) (4 boda) Aktuatorska jedinica ima mogućnost paljenja svjetala koju upravljačke jedinica ne koristi. Predložite oblikovno rješenje koje bi omogućilo automatsko paljenje svjetala u mraku bez izmjene drugih komponenata. Procijenite usklađenost rješenja s načelima oblikovanja.