Pismeni ispit iz Oblikovnih obrazaca u programiranju

	A	B	C	D
1.	○	○	○	○
2.	○	○	○	○
3.	○	○	○	○
4.	○	○	○	○
5.	○	○	○	○
6.	○	○	○	○
7.	○	○	○	○
8.	○	○	○	○
9.	○	○	○	○
10.	○	○	○	○
11.	○	○	○	○
12.	○	○	○	○

---

1. Razred koji igra ulogu subjekta u obrascu promatrača tipično sadrži sljedeće metode:
   (a) attach i notify
   (b) clone
   (c) update
   (d) create

2. Cilj nadogradnje bez promjene je:
   (a) omogućiti proširenje funkcionalnosti komponente bez mijenjanja njene implementacije
   (b) spriječiti da stari kod radi s novim kodom
   (c) osigurati da nadogradnja komponente zahtijeva promjene svih ovisnih komponenata
   (d) otvoriti komponente za nadogradnju, a otvoriti za promjene

3. Zadan je razred A u C++-u s dvije virtualne funkcije i jednim cjelobrojnim podatkovnim članom. Koliko bi mjesta na stogu 32-bitne organizacije tipično zauzimalo polje od 100 objekata tipa A?

   (a) 100 bajta

   (b) 1200 bajta

   (c) 1600 bajta

   (d) 800 bajta

4. U obrascu Promatrač eventualna ovisnost promatrača o subjektu nas ne smeta jer:
   (a) promatrači uglavnom imaju jednostavnu izvedbu
   (b) subjekt se znatno manje mijenja od promatrača
   (c) subjekt ima jednostavnu izvedbu
   (d) promatrači se rijetko mijenjaju

5. U krutom programskom sustavu unošenje skromne nove funkcionalnosti tipično za sobom povlači:
   (a) korištenje strukturirane paradigme
   (b) potrebu za neplaniranim izlaskom iz programa
   (c) korištenje objektno orijentirane paradigme
   (d) mijenjanje velikog broja komponenata

6. Obrazac strategija pospješuje inverziju ovisnosti jer:
   (a) strategija ne pospješuje inverziju ovisnosti
   (b) kontekst ne mora poznavati konkretne strategije
   (c) kontekst može raditi s različitim strategijama
   (d) mogu se po volji kombinirati različite porodice strategija

7. Funkcija f() prima referencu na objekt razreda A te poziva njegovu javnu virtualnu metodu m(). Razred B nasljeđuje A, te definira javnu metodu m(). Glavni program poziva f() s objektom razreda B. Koja metoda će se na kraju pozvati?
   (a) B::m()
   (b) doći će do pogreške pri izvođenju
   (c) A::m()
   (d) ovisi o implementaciji razreda A

8. U kojem sudioniku obrasca dekoratora su implementirane dodatne odgovornosti?
   (a) u konkretnim dekoratorima
   (b) u konkretnoj komponenti
   (c) u klijentima apstraktne komponente
   (d) u apstraktnom dekoratoru

9. Koje načelo oblikovanja garantira uspjeh poziva metode konkretnog objekta preko pokazivača na osnovni razred?
   (a) načelo jedinstvene odgovornosti
   (b) načelo nasljeđivanja implementacije
   (c) načelo nasljeđivanja sučelja
   (d) Liskovino načelo supstitucije

10. Dinamički polimorfizam u Pythonu ostvaruje se:
   (a) korištenjem polja pokazivača na funkcije
   (b) dinamički polimorfizam u Pythonu nije moguće ostvariti
   (c) rekurzivnim prozivanjem asocijativnih spremnika metoda
   (d) ponovnim prevođenjem parametriziranog koda

11. Ključne četiri komponente oblikovnog obrasca su:
   (a) zahtjevi, oblikovanje, izvedba, ispitivanje
   (b) naziv, problem, rješenje, rezultat
   (c) zamisao, izvedba, optimizacija, dokumentacija
   (d) strategija, promatrač, dekorator, tvornica

12. Koja je razlika između primitiva i nadomjestivih metoda (engl. hooks) osnovnog razreda u obrascu Okvirna metoda?
   (a) nadomjestive metode su implementirane u izvedenom razredu, a primitivi ne
   (b) primitive ne moramo implementirati u izvedenom razredu
   (c) nadomjestive metode su implementirane u osnovnom razredu, a primitivi ne
   (d) primitivi su razredi za razliku od nadomjestivih metoda

Zavod za elektroniku, mikroelektroniku
računalne i inteligentne sustave

27. travnja 2017.

Oblikovni obrasci u programiranju

Međuispit

1. Mirka je zainteresirala digitalna obrada zvuka. Napisao je razred `AudioClip` koji enkapsulira osnovne podatke o zvučnom signalu (frekvenciju i slijed uzoraka lijevog i desnog stereo kanala) te omogućava čitanje i pisanje prema standardu WAV odnosno niskopropusno filtriranje.

```
class AudioClip{
  AudioClip();
  // metode load_wav i save_wav ovise o biblioteci W!
  void load_wav(std::string file);
  void save_wav(std::string file);
  // metoda low_pass_filter ne ovisi o vanjskim komponentama
  void low_pass_filter(double sigma);
private:
  int frekvencija;
  std::vector<double> lijeviKanal;
  std::vector<double> desniKanal;
}
```

Mirkovi prijatelji su se oduševili ovom idejom i vrlo brzo dopisali sljedeće metode razreda `AudioClip`.

- Valentina je napisala metodu `AudioClip::load_wma` za učitavanje zvuka iz datoteke formata wma (ta metoda ovisi o biblioteci `X`).
- Mirjana je napisala metodu `AudioClip::load_from_mphone` za snimanje zvuka iz mikrofonskog ulaza te metodu `AudioClip::play` za reprodukciju zvuka na analognom izlazu računala (te dvije metode ovise o biblioteci `Y`).
- Jagor je napisao metodu `AudioClip::fft` za Fourierovu analizu zvuka (ta metoda ovisi o biblioteci `Z`).

Međutim, ubrzo su se javili problemi s ovisnostima komponente `AudioClip`. Valentina nije htjela gubiti vrijeme na instaliranje biblioteka `Y` i `Z`, a slično mišljenje je imala i Mirjana o bibliotekama `W`, `X` i `Z` te Jagor o `Y`.

Pomozite Mirku da poboljša distribuciju funkcionalnosti po komponentama tako da zadovolji želje prijatelja. Nacrtajte grafove ovisnosti početne organizacije i vašeg rješenja. U grafovima navedite komponente koje sadrže svu spomenutu funkcionalnost kao i glavne programe Valentine, Mirjane i Jagora te biblioteke `W`, `X`, `Y` i `Z`. Navedite oblikovna načela koja vaše rješenje zadovoljava, a početno nisu bila zadovoljena.

2. Zlatko je odlučio napisati biblioteku za točno računanje s različitim vrstama brojeva: cijelim, racionalnim itd. Osnovna ideja je omogućiti klijentima transparentno rukovanje različitim vrstama brojeva. Kako bi bolje upoznao problem Zlatko je napisao implementaciju funkcije `uvecaj_pa_reciprociraj` koja zadani broj uvećava za cijeli broj `x` i onda postavlja na recipročnu vrijednost:

```
void uvecaj_pa_reciprociraj(struct Broj* pb, int x){
  if (!strcmp(Broj.tip, "cijeli")){
    CijeliBroj* pcb = (CijeliBroj*) pb;
    pcb->broj += x;
    pcb->broj = 1 / pcb->broj;
  } else if (!strcmp(Broj.tip, "racionalni")){
    RacionalniBroj* prb = (RacionalniBroj*) pb;
    pcrb->brojnik += x * pcrb->nazivnik;
    int tmp = pcrb->nazivnik;
    pcrb->nazivnik = pcrb->brojnik;
    pcrb->brojnik = tmp;
  }
}
```

Zlatko očekuje da će se broj vrsta brojeva u budućnosti povećati (npr. modularni brojevi, kompleksni brojevi ili polinomi), kao i da će klijenti htjeti uvećavati i reciprocirati nezavisno o funkciji `uvecaj_pa_reciprociraj`.

(a) Navedite načela oblikovanja koja prikazani izvorni kôd ne zadovoljava.

(b) Predložite poboljšane implementacije funkcije `uvecaj_pa_reciprociraj` te struktura `Broj` i `RacionalniBroj` u programskom jeziku C. Navedite oblikovna načela koja vaše rješenje zadovoljava.

(c) Napišite glavni program koji primjenjuje funkciju `uvecaj_pa_reciprociraj` na racionalnim brojevima 1/2 i 5/3, oba puta uz x=4. Memoriju za dva racionalna broja potrebno je zauzeti odjednom.

3. Razmatramo računalni poslužitelj `Server` koji obrađuje poslove modelirane razredom `Task`:

```
class Task{
  virtual void execute()=0;
  virtual void store_result()=0;
  virtual std::string status()=0;
}
```

U implementaciji, `Server` sadrži FIFO spremnik (rep) čiji elementi su pokazivači na primjerke razreda `Task`. U programu imamo veliki broj konkretnih tipova zadataka, međutim, `Server` ih sve obrađuje na isti način, tako da prvo pozove metodu `execute`, a zatim metodu `store_results`. U nekim instancama našeg programa razred `Server` izvršava zadatke usporedno, tj. u različitim dretvama.

Javila se potreba za dva proširenja programa. Prvo, potrebno je omogućiti da za neke poslove pozivi metoda `store_results` budu zaštićeni mehanizmom međusobnog isključivanja (npr. `pthread_mutex_lock` itd.). Drugo, potrebno je omogućiti da status izvršavanja nekih zadataka bude dojavljen putem e-maila, objavljivanjem statusa na mrežnim stranicama ili na neki treći način.

Navedeni specijalni tretmani odabranih poslova moraju se moći kombinirati. Odluku o specijalnom tretmanu trebaju donositi klijenti razreda `Server`. Proširenja ne smiju uzrokovati promjene u razredu `Server`, razredu `Task` te u konkretnim razredima izvedenima iz razreda `Task`.

(a) Predložite oblikovno rješenje koje zadovoljava prikazane zahtjeve. Prikažite organizaciju strukturnim dijagramom prema konvenciji OMT i obrazložite njenu prikladnost s obzirom na načela oblikovanja.

(b) Skicirajte ključne dijelove implementacije specijalnih tretmana u programskom jeziku po želji.

(c) Navedite oblikovne obrasce koje ste koristili i povežite sudionike s elementima predložene organizacije.

Uputa: funkcionalni zahtjevi (izvedba međusovnog isključivanja, slanje e-maila, objavljivanje na mrežnim stranicama itd.) nisu predmet zadatka, slobodno ih zadovoljite pozivom prikladne vanjske komponente.

4. Prikažite organizaciju prikazanog izvornog kôda strukturnim dijagramom (OMT) i grafom ovisnosti komponenta. Navedite o kojem se oblikovnom obrascu radi te povežite razrede sa sudionicima obrasca. Napišite minimalni program koji provjerava mogu li se prikazani razredi prevesti i povezati.

```
class F{
  F(){}
  //...
};

class E{
protected:
  virtual void m()=0;
public:
  void n(F& f){
    // ...
    m();
  }
};
```

```
class G: public E{
public:
  G(){}
protected:
  virtual void m(){/* ... */}
};

class C{
  F f;
public:
  void o(G& g){
    g.n(f);
  }
};
```

5. Razmatramo komponentu koja enkapsulira verifikaciju integriteta podataka na disku računala funkcijom MD5:

```
// filepath - put do datoteke, true_hash - tocan hash
bool check_integrity(string filepath, string true_hash) {
    string data = load_data_from_file(filepah);
    return md5_hash(data) == true_hash;
}
```

Predložite novu organizaciju koja će podržati i) nove funkcije zažimanja SHA-1, SHA-2 i SHA-3 i ii) provjeru datoteka koje se nalaze na nekom mrežnom servisu. Nova organizacija treba biti u skladu s načelima oblikovanja te omogućiti klijentu da sam konfigurira na koji će način dohvaćati podatke i računati sažetak datoteke. Nacrtajte OMT dijagram i skicirajte kod u jeziku po želji. Navedite oblikovne obrasce koje ste koristili i povežite sudionike s komponentama vaše organizacije.