

## Pismeni ispit iz Oblíkovnih obrazaca u programiranju

	A	B	C	D
1.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
4.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
9.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
12.	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

1. Konkretni skupni objekti modernih biblioteka tipično definiraju:

(a) metodu tvornicu za konkretni iterator

(b) okvirnu metodu

(c) metodu za prelazak na sljedeći element

(d) metodu serialize()

2. Zašto je lakše ispitati programski sustav od digitalnog integriranog sklopa slične složenosti?

(a) jer programske komponente možemo ispitati u izolaciji

(b) jer se programske komponente mogu sveobuhvatno ispitati

(c) jer na ulaze programskog sustava možemo dovoditi proizvoljne podatke

(d) jer se digitalni sklopoli ponašaju nedeterministički

ju metodu tipično deklariraju jedinstveni obiti?

- (a) okvirnu metodu
- (b) javni konstruktor
- (c) metodu instance
- (d) virtualni destruktur

Obrazac Posjetitelj možemo zamijeniti obrascem Most samo ako:

- (a) operacije nad elementima uspijemo prikazati preko zajedničkog implementacijskog sučelja
- (b) sve konkretnе operacije nasljeđuju isti osnovni razred
- (c) za svaku vrstu elementa postoji metoda hijerarhije postupaka koja prima upravo takav argument
- (d) sve operacije nad elementima uspijemo prikazati metodama hijerarhije elemenata

5. O kojoj komponenti ovisi ciljni razred (engl. target) u obrascu Prilagodnik?

- (a) samo o klijentu
- (b) niti jedan od ponuđenih odgovora nije točan
- (c) o prilagodniku (engl. adapter)
- (d) i o klijentu i o vanjskom razredu (engl. adaptee)

6. O kojoj komponenti mora ovisiti Apstrakcija u obrascu Most?

- (a) o konkretnom izvođaču
- (b) o izvođaču
- (c) niti jedan od ponuđenih odgovora nije točan
- (d) o prilagođenoj apstrakciji

7. Koje su posljedice obrasca Stanje?

- (a) omogućava se atomarna i eksplisitna promjena ponašanja objekta
- (b) povećava se broj podatkovnih članova konteksta
- (c) ponašanja vezana uz različita stanja implementirana su u istoj komponenti
- (d) slabljenje jedinstvene odgovornosti

8. Ako je broj ovisnih objekata  $n$ , a broj neovisnih objekata  $m$ , onda obrazac promatrač tipično koristimo kad je:

- (a)  $n \geq 1, m \geq 1$
- (b)  $n = 1, m \geq 1$
- (c)  $n = 1, m = 1$
- (d)  $n \geq 1, m = 1$

9. Implementacija kompozita u programskom jeziku C trebala bi imati sljedeći podatkovni član:

- (a) pokazivač na matični objekt
- (b) polje pokazivača na apstraktne komponente
- (c) pokazivač na metodu osnovnog razreda
- (d) polje pokazivača na kompozite

10. Implementacija metode tvornice tipično se nalazi u:

- (a) konkretnom kreatoru
- (b) apstraktnom kreatoru
- (c) apstraktnom proizvodu
- (d) konkretnom proizvodu

11. Stvaranje prototipa može se zakomplicirati kod objekata koji:

- (a) sadrže posjetitelje
- (b) sadrže iteratore
- (c) sadrže kompozite
- (d) sudjeluju u cirkularnom lancu referenci

12. Okvirna metoda se u praksi često koristi za:

- (a) pozivanje bibliotečnih objekata nakon svake promjene stanja naše komponente
- (b) stvaranje objekata željenog vanjskog razreda
- (c) ubacivanje klijentskog kôda u bibliotečne razrede
- (d) dodavanje odgovornosti razredima vanjskih biblioteka

## Oblikovni obrasci u programiranju

### Pismeni ispit

1. Razmotrite sljedeći program u C++-u. Skicirajte struktturni dijagram razreda, te navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente organizacije sa sudionicima obrasca (ili obrazaca) te komentirajte usklađenost s načelima oblikovanja. Navedite što će se ispisati na ekranu nakon pokretanja programa. Predložite implementaciju ekvivalentnog programa u C-u.

```
class X{
public:
    virtual double action(double) =0;
};

class X1: public X{
public:
    virtual double action(double x){
        return x*x;
    }
};

class X2: public X{
    double offset;
public:
    X2(double val): offset(val) {}
    virtual double action(double x){
        return fabs(x) + offset;
    }
};

class C{
    X* px_;
public:
    C(X* px): px_(px){}
    double operate(
        std::vector<double> vec)
    {
        double rv=0;
        for (auto x: vec){
            rv+=px_->action(x);
        }
        return rv;
    }
};

int main(){
    C c(new X1);
    std::vector<double> data({1,2,1});
    std::cout <<c.operate(data) <<"\n";
}
```

2. Skicirajte dijagram razreda za domenu cestovnog prometa ako su poznate sljedeće činjenice. Važna cestovna vozila su automobili i motocikli. Konkretni automobili su Mazda 2, Ford Focus i Škoda Fabia, dok su konkretni motocikli Kawasaki ZX-12R i Tomos APN 6. Svaki automobil ima točno četiri kotača, dok motocikli imaju dva kotača. Automobili mogu imati dizelski ili četverotaktni benzinski motor. Motocikli mogu imati četverotaktni ili dvotaktni benzinski motor. Neki četverotaktni motori mogu biti korišteni i u motociklima i u automobilima.

3. Oblikujte razred Matrix za predstavljanje matrica realnih brojeva prema sljedećim uputama: i) konstruktor prima dimenzije matrice ( $m,n$ ), ii) metoda initialize postavlja sve elemente na nulu, iii) metoda get dohvata element matrice zadan indeksima, iv) metoda set pohranjuje zadani broj u element matrice zadan indeksima.

Oblikujte funkciju add koja zbraja dvije matrice zadane argumentima te dobiveni rezultat (novu matricu) vraća kao povratnu vrijednost.

Predložite reorganizaciju koja bi omogućila da klijenti transparentno mogu raditi s različitim izvedbama matrica (npr. razred Matrix te razred SubMatrix opisan u nastavku).

Oblikujte razred SubMatrix čiji primjeri se trebaju moći koristiti na svim mjestima gdje i primjeri razreda Matrix. Svaki primjerak razreda SubMatrix predstavlja podmatricu neke druge (pod-)matrice ali ne sadrži kopiju njenih elemenata (sve promjene bilo originala bilo kopije trebaju automatski biti vidljive i u drugom objektu).

Ilustrirajte funkcionalnost razvijenog koda na način da izvedete funkciju reduce\_sum koja vraća matricu dimenzija  $m \times 1$  čiji  $i$ -ti element odgovara zbroju svih elemenata  $i$ -tog retka ulazne matrice  $m \times n$ . U izvedbi koristite razrede Matrix i SubMatrix te funkciju add.

4. Robot Pioneer DX ima dva pogonska kotača kojima možemo neovisno upravljati. Za potrebe upravljanja tim robotom razvijen je sljedeći razred:

```
class Robot{  
private:  
    // data  
public:  
    void setLeftWheelSpeed(double kmh);  
    void setRightWheelSpeed(double kmh);  
};
```

Predložite organizaciju koja bi podržala sljedeće dodatne zahtjeve.

- iscrtavati procijenjeni položaj robota u zasebnom prozoru na zaslonu računala integriranjem informacija o dinamici kotača,
- omogućiti rad sa sličnim robotima drugih proizvođača (Magellan, R2D2, ...)

Zahtjeve je potrebno ostvariti bez duplicitanja koda, te bez mijenjanja postojećih klijenata razreda `Robot` koji nisu zaduženi za stvaranje novih objekata tipa `Robot`, nego reference na postojeće objekte dobivaju od svojih pozivatelja.

5. Razmatramo razred `XyzyButton` koji enkapsulira odgovarajući element grafičkog sučelja (dugme, gumb). Razred `XyzyButton` nije predviđen za nasljeđivanje, a isporučuje se u okviru biblioteke `Xyzy` koja je zamišljena da bez izmjena bude korištena od strane različitih klijenata. Također razmatramo aplikacijski program koji stvara objekte `leftMotor` i `leftButton` tipova `MyMotor` odnosno `XyzyButton`, te provodi potrebne radnje kako bi se na pritisak dugmeta dogodio poziv `leftMotor.start()`.

Skicirajte sučelje i izvedbu razreda `XyzyButton`, ključne detalje izvedbe aplikacijskog programa, kao i sučelje i izvedbu eventualnih dodatnih komponenata. Nacrtajte strukturni dijagram razreda i dijagram ovisnosti komponenata. Navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente organizacije sa sudionicima obrasca (ili obrazaca) te komentirajte usklađenost s načelima oblikovanja.

6. Tablični kalkulator poput Excela sadrži pravokutnu tablicu čija polja mogu sadržavati ili konstantu ili matematički izraz. Matematički izrazi mogu referencirati vrijednosti drugih polja koja pak mogu ovisiti o vrijednostima trećih polja itd. Kad god se sadržaj nekog polja X promjeni (npr. upisivanjem nove konstantne vrijednosti preko grafičkog sučelja) potrebno je ponovo izračunati vrijednosti svih polja čiji izrazi neposredno ili posredno ovise o polju X.

Neka je svako polje tablice modelirano primjerkom razreda `Cell`. Prepostavite da se izračunavanje izraza i dohvati svih polja koja se pojavljuju u izrazu provode pozivom postojećih funkcija `evaluate` i `getReferencedCells` nad znakovnim nizom koji predstavlja izraz.

Napišite sučelje i skicirajte izvedbu razreda `Cell` koja bi omogućila automatsko propagiranje izmjena kroz proizvoljno dugačke lance ovisnosti polja tablice (problem kružnih ovisnosti možete zanemariti). Navedite kojem obrascu odgovara vaše rješenje te nacrtajte dijagram razreda.