

Oblikovni obrasci u programiranju međuispit

Napomena uz zadatke 3-6: prihvaćaju se programska rješenja u C++-u, Javi, Pythonu i C#-u.

1. Promotrite kod zadan u nastavku zadatka. Nacrtajte strukturni dijagram razreda i dijagram ovisnosti komponenata. Komentirajte usklađenost organizacije s oblikovnim načelima koja smo spominjali na predavanjima. Navedite o kojem se oblikovnom obrascu radi, te povežite komponente sa sudionicima obrasca.

```
class Logger{
public:
    virtual void log(
        std::string report)=0;
};

class MyBase{
    Logger* plogger_;
public:
    void setLogger(
        Logger* pl)
    {
        plogger_=pl;
    }
    void transaction(){
        std::string report;

        // ...
        plogger_->log(report);
    }
};

class FileLogger: public Logger{
    FILE *fp_;
public:
    FileLogger(std::string filename){
        fp_=fopen(filename.c_str(), "w");
    }
    virtual void log(std::string report){
        fprintf(fp_, "MyBase:_%s\n", report.c_str());
        fflush(fp_);
    }
};

class ConsoleLogger: public Logger{ /*...*/};

class TcpIPLogger: public Logger{ /*...*/};

int main(){
    FileLogger fl("mylog.txt");
    MyBase mydb;
    mydb.setLogger(&fl);
    mydb.transaction();
}
```

2. Predložiti izvorni kôd u programskom jeziku C koji bi imao jednaka funkcionalna i organizacijska svojstvima kao i kôd iz prethodnog zadatka.
3. Predložite programsku organizaciju kojom bismo ostvarili ista svojstva kao i u zadatku 1, ali na način da se fleksibilnost ostvari nasljeđivanjem. Drugim riječima, dozvoljeno je koristiti samo modificirani razred MyBase te iz njega izvedene razrede MyBaseFileLogged, MyBaseConsoleLogged i MyBaseTcpIpLogged. Napišite implementaciju razreda MyBaseFileLogged, i predložite minimalni ispitni kôd. Nacrtajte strukturni dijagram razreda i dijagram ovisnosti komponenata. Usporedite organizacijska svojstva vašeg rješenja u odnosu na rješenje koje je predloženo u zadatku 1. Navedite o kojem se oblikovnom obrascu radi, te povežite komponente sa sudionicima obrasca.
4. 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. Komentirajte usklađenost organizacije s oblikovnim načelima koja smo spominjali na predavanjima. Navedite o kojem se oblikovnom obrascu radi, te povežite komponente sa sudionicima obrasca.
5. Razmatramo program za animaciju koji svoje rezultate može iscrtavati na ekranu računala ili u slijed slika konačne veličine. Program je napisan tako da se položaj animiranog objekta određuje delegiranjem apstraktnoj komponenti IAnimator čije sučelje sadrži samo jednu metodu:


```
public int getPositionAtTime(double time, Point p);
```

Metoda u drugi argument upisuje položaj na kojem se objekt mora prikazati u trenutku *time*, a kao povratnu vrijednost vraća vrijeme koliko objekt na tom položaju mora mirovati. U starom kôdu postojala je samo jedna implementacija ovog sučelja prikazana u nastavku.

```
class Animator1 extends IAnimator {
    public int getPositionAtTime(double time, Point p) {
        if (time<0) throw new IllegalArgumentException();
        p.x = ((int)(time*100) % 200) + 100;
        p.y = 75;
        return 150 + ((int)(time*3) % 50);
    }
}
```

U nekom kasnijem trenutku odlučeno je da gibanje objekta mora biti prirodnije pa je napisana nova implementacija:

```
class Animator2 extends IAnimator {
    public int getPositionAtTime(double time, Point p) {
        if (time<0) throw new IllegalArgumentException();
        p.x = (int)(350*Math.sin(12*time)) + 350;
        p.y = 75;
        return 120+(int)(150*Math.sin(24*time));
    }
}
```

- (a) Predočite međusobne odnose glavnog programa te komponenata za iscrtavanje i animaciju dijagramom razreda. Koje oblikovne obrasce prepoznajete? Povežite elemente dijagrama razreda sa sudionicima obrasca.
- (b) Postoji li mogućnost da nova verzija programa krši neko načelo oblikovanja? Pojasnite vaš odgovor primjerom.

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

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

Međutim, u tom trenutku su se javili dodatni zahtjevi:

- integriranjem informacija o dinamici kotača potrebno je iscrtavati procijenjeni položaj robota u zasebnom prozoru na zaslonu računala,
- potrebno je omogućiti rad i sa sličnim robotima drugih proizvođača (Magellan, R2D2, ...)
- prethodne zahtjeve potrebno je ostvariti bez potrebe za uvođenjem izmjena u kodu klijenata razreda *Robot* koji nisu zaduženi za stvaranje novih objekata tipa *Robot*, nego reference na postojeće objekte dobivaju od svojih pozivatelja.

Predložite organizaciju koja bi podržala ovakve zahtjeve. Nacrtajte strukturni dijagram razreda i dijagram ovisnosti komponenata. Komentirajte usklađenost organizacije s oblikovnim načelima koja smo spominjali na predavanjima. Navedite koje ste oblikovne obrasce koristili, te povežite komponente sa sudionicima obrasca.