

## Oblikovni obrasci u programiranju međuispit

Napomena uz zadatke 2-6: prihvaćaju se programska rješenja u C++-u, Javi, Pythonu i C#-u.

1. U računalnoj igri na temu ratova zvijezda javila se potreba za uniformnim baratanjem robotima s kotačima i humanoidnim robotima. Fragment odgovarajućeg kôda u programskom jeziku C++ je prikazan u nastavku.

```
class Robot {
public:
    virtual void kreni() =0;
    virtual void stani() =0;
    /* ...
    ...
    */
};

void testRobot(Robot* p) {
    p->kreni();
    sleep(2);
    p->stani();
}

class R2D2: public Robot {
    int brzina_;
public:
    R2D2(): brzina_(0) {}
public:
    virtual void kreni() {brzina_=10;}
    virtual void stani() {brzina_=0;}
};

class C_3PO {/*...*/};

int main() {
    testRobot(new R2D2());
}
```

Prikazati odsječak u programskom jeziku C kojim bi se ostvarila funkcionalnost identična prikazanom fragmentu. (12 bodova)

2. Razmatramo funkciju za sortiranje selekcijom prema sljedećem kôdu u Pythonu.

```
# funkcija za sortiranje liste L
def poredaj2(L):
    # petlja po svim indeksima liste
    for i in range(len(L)):
        # pronadi indeks najmanjeg u ostatku liste
        min_j = i
        for j in range(i+1, len(L)):
            if L[j] < L[min_j]:
                min_j = j
        # zamijeni tekuci element s najmanjim
        L[i], L[min_j] = L[min_j], L[i]
```

Predložite rješenje koje bi podržalo mogućnost zadavanja različitih kriterija usporedbe (silazno, uzlazno, prvo neparni pa parni, ...) prema kojima se sortiranje provodi. Navedite koji oblikovni obrazac ste upotrijebili. Riješite zadatak primjenom:

- (a) dinamičkog polimorfizma u jeziku sa statičkim tipiziranjem,
- (b) dinamičkog polimorfizma u jeziku s implicitnim tipiziranjem.
- (c) statičkog polimorfizma u jeziku C++ (bonus).

(12+5 bodova)

3. I u ovom zadatku razmatramo računalnu igru u kojoj se javljaju roboti (`class Robot {/*...*/}`) s kotačima (npr. `class R2D2: public Robot {/*...*/}`) i humanoidi (npr. `class C_3PO: public Robot {/*...*/}`). Igra treba podržati mogućnost da se robot `C_3PO` popne uz ljestve kako bi pokušao pobjeći humanoidnim robotima trgovačke federacije. S obzirom na to da roboti trgovačke federacije trebaju pratiti `C_3PO`-a oni se također moraju moći penjati. Stoga je programer skicirao sljedeći kôd.

```
void penji(Robot *pr){
    while (True){
        pr->hvatajLijevomRukomGore();
    }
}
```

```

    if (! pr->uhvatioLetvicu())
        break; /*dosli smo na vrh*/

    pr->popniDesnomNogom();
    pr->hvatajDesnomRukomGore();
    pr->staniLijevoNogom();
}
}

```

Kako bi riješio probleme koji su se javili prilikom prevođenja prikazane funkcije, programer je metode penjanja i hvatanja odlučio dodati u novu verziju razreda `Robot`. Međutim, tek tada su počeli pravi problemi. Objasnite koje je načelo oblikovanja prekršeno u novoj organizaciji te obrazložite svoj odgovor. Predložite poboljšanu verziju funkcije `penji`, te pokažite kako bi trebalo izmijeniti ostatak programa. (12 bodova)

4. Nacrtajte dijagram mogućeg međusobnog odnosa razreda `JComponent`, `JTextArea`, `JScrollPane`, te `JFrame` prema programskom odsječku na slici. U rješavanju zadatka pretpostavite da metoda `JFrame::getContentPane()` vraća referencu na objekt tipa `ContentPane`.

```

JComponent ta = new JTextArea("Text");
JFrame frame = new JFrame();
if (config.hasScrollbar()){
    JComponent jsp = new JScrollPane(ta);
    frame.getContentPane().add(jsp);
} else{
    frame.getContentPane().add(ta);
}

```

(12 bodova)

5. 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` promijeni (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`. Pretpostavite da se izračunavanje izraza i dohvat 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.

Bonus: predložite rješenje za detekciju kružnih ovisnosti (što ako korisnik u polju `X` upiše izraz koji referencira polje `Y`, a polje `Y` neposredno ili posredno referencira polje `X`?), te automatsko osvježavanje grafova i dijagrama. (12+5 bodova)

6. Kojem oblikovnom obrascu odgovara prikazani izvorni kôd? Povežite razrede `X`, `Y`, `X2`, `Y2` sa su-dionicima obrasca. Nacrtajte odgovarajući dijagram razreda te dijagram ovisnosti komponenata. (10 bodova)

```

class Y { /*...*/ };

class X {
    std::vector<Y*> ys;
public:
    void m1(){
        ys.push_back(m2());
    }
    virtual Y* m2() =0;
};

class Y2:
    public Y
{
    // ...
};

class X2:
    public X
{
    virtual Y* m2(){return new Y2};
};

```