

## Oblikovni obrasci u programiranju međuspit

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. (10 bodova) Tvrtka Bogosoft razvija programske projekte koji zahtijevaju dijalog s korisnikom preko tekstnih naredbi, pri čemu svaka naredba odgovara jednom retku teksta. Svi ti proizvodi zahtijevaju:
  - (1) postavljanje odzivnog teksta (prompta) na standardni izlaz (podrazumijevani prompt je: "> "),
  - (2) zaprimanje i izvođenje naredbi sa standardnog ulaza,
  - (3) obradu naredbe `quit` na način da se baci iznimka `SystemExit` (ali to ponašanje klijenti moraju moći promijeniti),
  - (4) obradu praznog retka na način da se ponovo izvrši posljednja naredba (ali to ponašanje klijenti moraju moći promijeniti),

Potrebno je razviti što jednostavnije oblikovno rješenje koje bi minimiziralo ponavljanje i pospješilo dijeljenje koda u različitim projektima u skladu s načelima oblikovanja.

Upute. U predloženom rješenju ne treba razmatrati konkretne naredbe te njihovo parsiranje i izvođenje. Klijentski kod bi trebao moći izgledati ovako (C++):

```
int main {  
    ...  
    MyInterpreter interpreter;  
    interpreter.cmdloop();  
}  
  
// definicija razreda MyInterpreter  
...
```

2. (10 bodova) U okviru složenog programskog sustava neke tvrtke razmatramo modeliranje podsustava za slanje poruka. U trenutnoj inačici sustava na raspolaganju su 4 implementacije komunikacijskih podsustava koji se mogu koristiti i za slanje poruka; očekuje se variranje tog broja s daljnjim inačicama. Konceptualno, poruke se šalju metodom tipa: `bool send(destinationUser, subject, messageBody)`, gdje povratna vrijednost `true` znači uspješno slanje, a `false` neuspjeh.

Prilikom slanja poruke potrebno je podržati i sljedeće funkcionalnosti (koje se trebaju moći kombinirati na proizvoljne načine):

- (1) ako se poruka šalje izvan radnog vremena, o tome je potrebno obavijestiti nadležnu službu,
- (2) tijelu poruke treba automatski dodati dopunu da se radi o povjerljivoj komunikaciji,
- (3) ako slanje poruke ne uspije, treba se izazvati iznimka.

Prilikom daljnjeg razvoja sustava očekuje se da bi broj ovakvih dodatnih zahtjeva također mogao rasti.

- (a) (3 boda) Predložite prikladno oblikovno rješenje podsustava za slanje poruka. Koji je oblikovni obrazac prikladan? Nacrtajte strukturni dijagram.
  - (b) (2 boda) Povežite sudionike oblikovnog obrasca s Vašim rješenjem.
  - (c) (3 boda) Napišite okvirni programski kod u programskom jeziku po izboru.
  - (d) (2 boda) Objasnite načela s kojima je usklađeno Vaše rješenje.
3. (10 bodova) Skicirajte dijagram razreda za domenu izvanplanetarnog prometa. Važna vozila su svemirski brod i sonda. Svemirski brodovi mogu biti straživački i teretni. Neki konkretni modeli brodova su Intrepid (istraživački) i YT-1300 (teretni). Konkretni sonde su Voyager i Cassini. Svako vozilo ima barem 1 potisnik. Potisnik može biti ionski ili raketni. Sonda može imati samo raketne potisnike. Teretni svemirski brod može prevoziti druga vozila (ako ima mjesta).

4. (10 bodova) Razmatramo razred `Trainer` koji implementira učenje postupka za klasifikaciju slike (u daljnjem tekstu: *model*). Model na ulazu prima sliku a na izlazu vraća vektor brojeva ( $p_i$ ). Brojevi  $p_i$  odgovaraju vjerojatnosti  $i$ -tog razreda (vrijedi  $p_i > 0$ ,  $\sum p_i = 1$ ), pri čemu  $i$  označava indeks razreda kojem slika pripada (npr. 0=mačka, 1=miš, 2=pas itd.). Razred `Trainer` učenje provodi kroz 10 epoha. Svaka epoha iterira po listi imena datoteka slika za učenje te svaku sliku učitava, na nju primjenjuje model te računa pogrešku (realni broj) na temelju koje "popravlja" parametre modela. Pogreška se računa na temelju indeksa točnih razreda koji su sastavni dio skupa za učenje. Poznato je da učenje traje danima. Zbog toga je praktično pratiti napredovanje učenja na različite načine kao što su:

- pohraniti svaku sliku za koju je pogreška veća od unaprijed zadanog praga u obliku `/direktorij/za/spremanje/<ime_slike>-<iznos_pogreske>.jpg`
- ispisati na standardni izlaz koliko slika je točno klasificirano nakon posljednje epohe učenja
- izračunati te na standardni izlaz ispisati srednju vrijednost pogreške i posljednjoj epohi
- podržavati različite načine za računanje pogreške ( $L_2$  udaljenost vjerojatnosti točnog razreda od jedinice, broj netočno klasificiranih slika itd).

Očekuje se da ćemo se u budućnosti sjetiti novih načina za praćenje napredovanja postupka. Predložite oblikovno rješenje koje podržava navedene zahtjeve u skladu s načelima oblikovanja. Uputa: bodove ćemo dijeliti za oblikovnu kvalitetu rješenja, a ne za algoritamske detalje.

5. (10 bodova) U C++ su implementirane klase `Square` i `Identity`, koje redom implementiraju kvadratnu i funkciju identiteta nad cijelim brojevima.

Poznat nam je strojni kod konstruktora razreda `Identity`:

```

1  mov  QWORD PTR [rbp-8], rdi
2  mov  rax, QWORD PTR [rbp-8]
3  mov  rdi, rax
4  call Unary_Function::Unary_Function()
5  mov  edx, OFFSET FLAT:vtable_for Identity+16
6  mov  rax, QWORD PTR [rbp-8]
7  mov  QWORD PTR [rax], rdx

```

Poznat nam je i strojni kod funkcije `main`:

```

1  sub  rsp, 24
2  mov  edi, 1
3  call operator new(unsigned long)
4  mov  QWORD PTR [rbp-24], rax
5  mov  rax, QWORD PTR [rbp-24]
6  mov  esi, 5
7  mov  rdi, rax
8  call Square::negative_value_at(int)
9  mov  edi, 8
10 call operator new(unsigned long)
11 mov  rbx, rax
12 mov  QWORD PTR [rbx], 0
13 mov  rdi, rbx
14 call Identity::Identity()
15 mov  QWORD PTR [rbp-32], rbx
16 mov  rdx, QWORD PTR [rbp-32]
17 mov  rax, QWORD PTR [rbp-32]
18 mov  rax, QWORD PTR [rax]
19 add  rax, 8
20 mov  rax, QWORD PTR [rax]
21 mov  esi, 3
22 mov  rdi, rdx
23 call rax

```

```

1  class Unary_Function {
2  public:
3      virtual int value_at(int x) = 0;
4      virtual int neg_value_at(int x) {...}
5  };
6
7  class _____ {
8  public:
9      int value_at(int x) {...};
10     int neg_value_at(int x) {...}
11 };
12
13 class _____ : public Unary_Function {
14 public:
15     virtual int value_at(int x) {...};
16 };
17
18 int main() {
19     _____ *f1 = new _____();
20     f1->neg_value_at(5);
21     _____ *f2 = new _____();
22     f2->neg_value_at(3);
23     return 0;
24 }

```

- (1 bod) Što je argument, a što povratna vrijednost operatora `new`? Demonstrirajte na strojnom kodu.
- (2 boda) Nadopunite pripadajući C++ kod i obrazložite svoje rješenje.
- (2 boda) Koji su argumenti prilikom poziva funkcije u retku 20 C++ koda. Dokažite pomoću strojnog koda.
- (1 bod) Koji je sadržaj registra `rax` nakon izvođenja instrukcije u retku: 6, 17, 18, 19 i 20
- (2 boda) Povežite linije C++ `main` funkcije s odgovarajućim linijama strojnog koda.
- (1 bod) Kako izgleda strojni kod konstruktora za razred `Square`?
- (1 bod) Koji su problemi trenutne implementacije i kako ju se može poboljšati?