

Domaća zadaća 1

Poštovane kolege, pred vama je vaša prva domaća zadaća kojoj je cilj provjeriti usvajanje osnovnih koncepata OO paradgime.

Zadaća se sastoji od dva zadatka. Prvi zadatak predstavlja implementaciju Kalkulatora, i zajednički je za svih (ie., svi rješavate isto) dok svoj drugi zadatak odabirete iz skupine danih zadataka na način da uzmete zadnje 4 znamenke svog JMBAG-a i obavite nad tako dobivenim brojem operaciju $\% 7 + 1$ (čime ćete dobiti redni broj zadatka koji trebate riješiti)

Zadatke možete rješavati korištenjem C++, C# ili Java programskih jezika i to korištenjem Visual Studio 2005 ili Eclipse razvojnih okruženja (ukoliko bi netko to baš rado odradio u Pythonu ili Rubyu, nek' mi se direktno javi na mail).

Domaću zadaću šaljete meni mailom na adresu zvonimir.vanjak@fer.hr, i to isključivo s adrese u kojoj je sadržano vaše ime i prezime (preferabilno @fer.hr, ali može i drugo). Prilikom slanja zadaće mailom u subject obavezno napisati „OO DZ1“ (bez navodnika), a u tekstu maila mora biti napisan vaš JMBAG i način izračuna rednog broja zadatka.

U prilogu maila predajete zip-ani direktorij sa solution-om u kojem se nalazi vaše rješenja. Direktorij/solution MORA biti imenovan na sljedeći način – Prezime_Ime_DZ1 i u njemu se očekuju dva projekta za dva riješena zadatka (ako ste riješavali oba – s time da naziv projekta mora odgovarati onome definiranom tekstom zadatka).

Za rješavanje zadaće imate 10 dana (rok predaje je 22.10.2008 u ponoć)

BODOVANJE

Za domaću zadaću je u ukupnom zbroju predviđeno 10% bodova a ova prva će nositi nekih 4-5% od tih 10%

Prvi zadatak (Kalkulator) nosi 30% bodova ove domaću zadaću, dok rješavanje drugog zadatka nosi preostalih 70%. Posebno kvalitetna rješenja će biti nagrađena ekstra bodovima.

VAŽNA NAPOMENA:

S obzirom na preko 50-ak zadaća koje treba ispraviti i ocijeniti, na upogonjavanje svakog rješenja ću potrošiti otprilike 30 sekundi. Dakle, Save as, Unzip, otvaranje u VS/Eclipse, Build, Run i to je to. Ako to ne prođe, znači ukoliko postoje sintaksne pogreške, ili se projekt ne može prevesti, ili se ne može pokrenuti (npr. zbog hardkodiranog patha koji ne

postoji na mom računalu) **automatski slijedi 0 bodova** bez ikakvog gledanja izvornog kôda!!!

VAŽNA, VAŽNA NAPOMENA - PREPISIVANJE!!!

Svako prepisivanje, pokušaj prepisivanja ili sumnja na prepisivanje će se kazniti RADIKALNIM mjerama!

Svakako imajte u vidu da zadaci pružaju poprilične mogućnosti za varijacije u rješenjima tako da primjena poznatog principa „promjena imena varijabli/funkcija/klasa“ (vjerujem da niste toliko blesavi da pokušate proći sa običnom copy-paste metodom!) neće dobro završiti po vas!

Završna napomena – JEL JA TO BAŠ MORAM ?

Ništa se u životu ne mora :-). Rješavanje ove domaće zadaće u cijelom zbiru nosi „tek“ nekih 4-5 % bodova tako da to nije toliko problem, ali posrtanje u ovako ranoj fazi nikako ne bi bio dobar znak :-))).

Na što obratiti pažnju pri rješavanju:

- traži se solidan dizajn fokusiran na rješavanje definiranog problema
- „pametna“ imena varijabli, funkcija i klasa
- komentara treba biti u pravoj mjeri – znači, komentirati složenije komade kôda (a ukoliko dobro definirate i imenujete objekte i pripadne funkcije, komentara skoro pa da i ne treba biti)

Zadatak I (za svih isti) – Kalkulator

Implementirati klasu Kalkulator koja će u funkcionalnom smislu oponašati klasičan kalkulator („digitron“) koji ima LED displej za prikaz zadnjeg unesenog broja (ili rezultata) i tipke kojima se unose brojevi i zahtjevi za izračunavanje. Kalkulator mora imati (bar) sljedeće osnovne funkcije: +, -, *, /, $\sqrt{}$, x^2 , x^3 , $1/x$, sin, cos, tan i exp, a ugrađivanje funkcionalnosti za memoriranje nosi bonus bodove).

Rad izgrađene klase ilustrirati kroz konzolnu aplikaciju koja u main() ispisuje trenutno stanje na displeju, čeka na pritisak tipke (kao pomoć korisniku se ispod ispisa rezultata mogu ispisati kratice za tipke, odnosno što koja tipka radi), prosljeđuje kôd pritisnute tipke kalkulatoru te zatim ispisuje novo stanje LED displeja.

Zadatak II - 1 - Biblioteka

Izgraditi jednostavan sustav za podršku rada biblioteke (knjižnice). Definirati objektni model s klasama Biblioteka, Korisnik, Knjiga, Posudba za podršku realiziranju sljedeće funkcionalnosti:

- dodaj novog člana
- dodaj novu knjigu
- evidentiraj posudbu knjige
- evidentiraj vraćanje knjige
- izlistaj članove koji kasne s vraćanjem (rok za vraćanje je 15 dana)
- izlistaj članove po aktivnosti (ie. sortirati ih po broju knjiga koje su posudili)

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata u main() te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju dva „izvještaja“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom. U ovom slučaju skup funkcionalnosti koje aplikacija podržava se mora proširiti sa sljedećim opcijama:
 - izlistaj sve korisnike
 - izlistaj sve knjige
 - izlistaj sve posudbe
 - učitaj default model – KLJUČNA opcija koja služi tome da ocjenjivač ne mora ručno unositi podatke radi provjere da li sustav dobro radi

Zadatak II - 2 – Željeznički promet

Izgraditi jednostavan sustav za praćenje željezničkog prometa, ukratko opisanog ovako - Kompozicije putuju između Stanica, prevozeći pritom Putnike i Teret.

U objektnom modelu definirati ŽeljezničkuKompoziciju (koja može biti putnička ili teretna i čine ju određen broj vagona te jedna ili dvije lokomotive) koja obavlja Putovanje između Stanica (pazite: tijekom jednog putovanja se prolazi kroz više stanice, odnosno nije dovoljno samo modelirati početnu i završnu!), PutovanjePutnika (znači, prijevoz putnika između dvije stanice) te PrijevozTereta (prijevoz određene količine tereta između dvije stanice).

Ugraditi potrebnu funkcionalnost za rad objektima u modelu (dodaj novu kompoziciju, dodaj putnika, dodaj teret) te dva izvješća:

- izvješće za kompoziciju – ispisuje što je sve prevezeno pojedinom kompozicijom
- sumarno izvješće – za sve kompozicije ispisuje sumarne podatke o tome koliko je „prevezeno“ putnik-kilometara i kilogram-kilometara

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju dva navedena „izvještaja“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom. U ovom slučaju skup funkcionalnosti koje aplikacija podržava se mor proširiti sa sljedećim opcijama:
 - izlistaj sve kompozicije/putovanja
 - ispiši što je sve prevezeno određenom kompozicijom
 - učitaj default model – KLJUČNA opcija koja služi tome da ocjenjivač ne mora ručno unositi podatke radi provjere da li sustav dobro radi

Zadatak II - 3 – Studentska referada

Izgraditi jednostavan sustav za podršku radu studentske referade. Modelirati objekte Student, Predmet, IspitniRok, IzlazakNaIspit, povezati ih i ugraditi podršku za sljedeću funkcionalnost:

- dodaj studenta
- dodaj predmet
- definiraj (dodaj) ispitni rok
- prijavi studenta na rok
- unesi studentovu ocjenu
- ispiši izvješće za ispitni rok (podaci o ispitnom roku, ko je prijavljen i ocjene)
- ispiši izvješće za studenta (podaci o svim ispitima na koje je student izašao)

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju navedeni „izvještaji“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom (uz KLJUČNU OPCIJU „učitaj default model“ koja služi tome da ocjenjivač ne mora ručno unositi podatke radi provjere da li sustav dobro radi)

Zadatak II – 4 – Praćenje prehrane

Izgraditi sustav za praćenje prehrane. U sustavu modelirati osnovne koncepte Obroka (doručak, užina, ručak, večera) te Namirnice. Namirnica je definirana svojim nazivom, vrstom mjere – komad/litra/kilogram, te sastavom: ugljikohidrati, proteini, masti +

dodatno „vitaminski faktor“ koji kaže koliko dnevnih potreba za vitaminima zadovoljava namirnica.

Poželjno je razraditi sistem kvantificiranja unesenih namirnica a dodatno se honorira i razrada „vitaminskog faktora“.

U objektni model ugraditi osnovnu funkcionalnost:

- evidentiraj obrok
- definiraj/dodaj vrstu namirnice
- ispis izvješća za tjedan i mjesec – oba izvješća se sastoje od sumarnog dijela (gdje se ispisuju sumarni podaci o unosu namirnica tijekom cijelog perioda) te detaljnog ispisa svih obroka (sortiranog vremenski!)
- best/worst days izvješće – za zadani period ispisuje 5 najboljih odnosno najgorih dana (u smislu količine konzumiranih kalorija – za proračun kalorijske vrijednosti namirnice pretpostavite da gram ugljikohidrata i proteina nosi 4 (kilo)kalorije, a gram masti 9)

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju navedeni „izvještaji“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom (uz KLJUČNU OPCIJU „učitaj default model“ da se ne mora ručno unositi podatke radi provjere da li sustav dobro radi)

Zadatak II – 5 – Trgovina

Izgradite sustav za potporu poslovanja male trgovine. U trgovini se prodaje određeni skup Artikala (definiran nazivom i cijenom, pretpostavite da se prodaju ili po komadu ili po kilogramu), a podaci o kupovini se pamte preko Računa (koji sadrži popis kupljenih artikala).

Ugraditi funkcionalnost za:

- definiranje artikla
- evidentiranje novog računa
- dodavanje stavke (artikla) u račun
- „ispis“ računa (na ekran!)
- ispis izvješća za dan – sumarni dio u kojem se ispisuje koliko je izdato računa, ukupna vrijednost prodane robe, te popis svih računa (s time da se svaki račun ispisuje u jednom retku s vremenom izdavanja, brojem artikala i iznosom računa)
- izvješće po artiklima – sortirani ispis najprodavanijih artikala (po vrijednosti)

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju navedeni „izvještaji“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom (uz KLJUČNU OPCIJU „učitaj default model“ da se ne mora ručno unositi podatke radi provjere da li sustav dobro radi)

Zadatak II - 6 – Iznajmljivanje apartmana

Izgraditi sustav za potporu iznajmljivanju apartmana („mali ugostitelji“). Modelirati koncepte Apartman (naš mali ugostitelj ih može ih imati i više, s različitim brojem kreveta i različitim cijenama za različito „doba godine“, odnosno predsezonu, sezonu i postsezonu), Rezervacija, Gost i ugraditi sljedeću funkcionalnost:

- dodaj apartman
- dodaj rezervaciju
- evidentiraj dolazak gosta
- evidentiraj odlazak gosta
- ispis sumarnog izvješća – ispisuje podatke za sve apartmane: ukupna zauzetost u sezoni (broj iznajmljenih dana), broj gost
- ispis izvješća za određeni apartman – ispisuje sumarne podatke za apartman i izlistava popis svih gostiju u tom apartmanu

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju navedeni „izvještaji“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom (uz KLJUČNU OPCIJU „učitaj default model“ da se ne mora ručno unositi podatke radi provjere da li sustav dobro radi)

Zadatak II - 7 - Burza

Izgraditi sustav za praćenje stanja na burzi. Izgraditi Burzu kao osnovni objekt, na kojoj se trguje određenim skupom Dionica, a čija se cijena formira na tržištu (i za naš sustav je eksterni faktor – znači, unosi se izvana!). Na burzi je definiran i Indeks, koji predstavlja prosječnu (trenutnu!) vrijednost svih dionica na burzi.

Sustav mora imati sljedeću funkcionalnost:

- definiraj/dodaj dionicu

- unesi trenutnu cijenu dionice
- ispiši kretanje vrijednosti indeksa kroz određeni zadani period
- ispiši pet dionica koje su u zadanom danu najviše mijenjale vrijednost (najveća razlika između max. i min. cijene u danu)

Izgraditi jednostavnu konzolnu aplikaciju kroz koju će se ilustrirati izgrađena funkcionalnost na jedan od dva načina:

- ilustrirati rad izgrađenog objektnog modela kroz direktno instanciranje objekata te pozivanje članskih funkcija nad tim objektima (dakle, bez ikakve interakcije s korisnikom, svi podaci se unose direktnim pozivima funkcija nad objektima te se u konačnici ispisuju navedeni „izvještaji“)
- izgraditi „interaktivnu“ konzolnu aplikaciju koja ispisuje meni s ponuđenim funkcijama te omogućuje korisniku da sam „radi“ a aplikacijom (uz KLJUČNU OPCIJU „učitaj default model“ da se ne mora ručno unositi podatke radi provjere da li sustav dobro radi)