

Objektno orijentisano programiranje

– II kolokvijum (grupa A) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Hotel je odlučio da ponudi novu uslugu gostima da odlože svoje dragocenosti u sefove. Dragocenosti se ne smeštaju direktno u sefove već u kutije koje stoje u sefu za koje ključ poseduje jedino gost hotela. Hotel je sve sefove smestio u jedan red. Kako je hotel u razvoju broj sefova nije fiksna (raste kupovinom novih). Kada gost hotela zatraži sef njemu se dodeljuje prvi prazan sef (ako to nije moguće šalje se obaveštenje menadžeru hotela), pamti se broj dana koliko će dragocenosti ostati u sefu i saopštava se redni broj sefa. Preuzimanje dragocenosti se obavlja na osnovu broja rednog broja sefa (ako je redni broj pogrešan aktivira se upozorenje menadžeru hotela), pri čemu se (dragocenosti) kutije iz sefova iza njega pomeraju za jedno mesto ulevo kako popunjeni sefovi bili redom na početku, a prazni na kraju. Na zahtev vlasnika hotela moguće je odrediti prosečnu dužinu boravka trenutno smeštenih dragocenosti u sefu. Hotel ima mogućnost da uredi dragocenosti (kutije) u sefovima u rastući ili opadajući redosled po nekoj osobini. Hotel može da prikaze izveštaj o svim zauzetim sefovima. U slučaju opasnosti od krađe hotel pamti raspored dragocenosti (kutija) po sefovima i premešta ih na sigurno mesto u drugom hotelu. Drugi hotel priprema tačno onoliko sefova koliko je potrebno i stavlja dragocenosti (kutije) u sefove po zapamćenom redosledu (ako se pojavi greška aktivira se upozorenje menadžeru hotela).

Hotel je rešio da podeli uvođenje ove nove usluge u dve faze. U prvoj fazi će se pri smeštanju dragocenosti koristiti samo dužina čuvanja dragocenosti (kako bi se analizirala isplativost usluge) i sve moguće operacije će se obavljati na osnovu nje, dok je na recepcionerima da vode evidenciju o tome kom gostu je dodeljen koji sef. U drugoj fazi će za svaku dragocenost voditi evidenciju o broju sobe gosta, imenu gosta i dužini čuvanja dragocenosti. Hotel će moći da obavi sve predviđene operacije na osnovu ovih podataka o dragocenostima, pri tome će za uređenje u rastući redosled koristiti broj sobe, a za uređenje u opadajući redosled koristiti vreme čuvanja dragocenosti. Na početku svake faze uvođenja usluge hotel će izvršiti testiranje rada tako što će staviti na čuvanje nekoliko dragocenosti, zatim preuzeti bar dve dragocenosti, odrediti prosečnu dužinu čuvanja dragocenosti, urediti dragocenosti (kutije) u rastući pa potom u opadajući redosled i izvršiti vežbu opasnosti od krađe. Nakon svake operacije prikazati sadržaj sefova u hotelu.

PREDMETNI NASTAVNICI

Objektno orijentisano programiranje

– II kolokvijum (grupa B) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Železnica uvodi novu uslugu otpremanja i praćenja teretnih vozova samo na jednoj liniji. Broj vagona koje će voz da povuče je promenljiv i određuje se na početnoj stanici, gde se na voz kače prazni vagoni. Kada se zatraži tovar neke robe, njoj se dodeljuje prvi prazan vagon (ako to nije moguće šalje se obaveštenje glavnom menadžeru), pamti se težina utovarene robe. Roba se može utovariti na početnoj stanici ili na nekoj međustanici. Na svakoj od međustanica moguće je isprazniti neki od vagona na osnovu njegovog rednog broja u kompoziciji, pri čemu se taj vagon prebacuje na kraj kompozicije (ako je redni broj pogrešan aktivira se upozorenje glavnom menadžeru). Na zahtev je moguće u svakom trenutku odrediti ukupnu težinu robe koja se nalazi u punim vagonima. Železnica ima mogućnost da uredi pune vagone u rastući ili opadajući redosled po nekoj osobini na bilo kojoj stanici na putu voza. Železnica može da prikaze izveštaj o punim vagonima. Na svakoj stanici, u slučaju potrebe, moguće je istovariti sadržaj svih vagona i zapamtiti raspored gde je koja roba bila u kompoziciji. Takođe, moguće je natovariti isti ili neki drugi voz istovarenom robom u zapamćenom redosledu (ako se pojavi greška aktivira se upozorenje glavnom menadžeru), pri čemu se formira kompozicija koja će imati tačno tri vagona više od zapamćene kompozicije (dakle biće tačno 3 prazna vagona).

Železnica je rešila da podeli uvođenje ove nove usluge u dve faze. U prvoj fazi će se pri otpremanju i praćenju vozova koristiti samo težina robe u vagon i sve moguće operacije će se obavljati na osnovu nje. U drugoj fazi će za svaku robu u vagonu voditi evidenciju o ciljnoj stanici, udaljenost ciljne stanice od polazne i težini robe u vagonu. Železnica će moći da obavi sve predviđene operacije na osnovu ovih podataka o robi u vagonu, pri tome će za uređenje u rastući redosled koristiti udaljenost ciljne od polazne stanice, a za uređenje u opadajući redosled koristiti težinu robe. Na početku svake faze uvođenja usluge železnica će izvršiti testiranje rada tako što će napuniti nekoliko vagona, zatim istovariti bar dva vagona, odrediti ukupnu težinu vagona, urediti kompoziciju u rastući pa potom u opadajući redosled, istovariti robu i utovariti je na drugi voz. Nakon svake operacije prikazati sadržaj vagona u kompoziciji.

PREDMETNI NASTAVNICI

Objektno orijentisano programiranje

– II kolokvijum (grupa C) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Potrebno je modelirati laboratorijski biohemijski analizator. Svaki analizator opisan je tipom i nazivom. Kako bi se postigla efikasnost u analiziranju uzoraka materijala, uzorci se stavljaju u jednodimenzionalne reke. Jedan analizator sadrži samo jedan rek. U reku se stavljaju epruvete sa uzorcima materijala. Svaki rek može da primi određeni broj epruveta sa uzorcima. Za jedan analizator moguće je koristiti reke sa različitim brojem mesta za epruvete. Da bi rad analizatora bio efikasan, rek mora da je kompaktno popunjen epruvetama, bez praznih pozicija između epruveta. U svakom trenutku analizator treba da ima informaciju o tačnom broju epruveta u reku. Dodavanje epruvete u reku radi se tako što se epruveta stavi na prvu slobodnu poziciju ukoliko ima mesta za smeštanje. Nepravilno obeležene epruvete moguće je ukloniti iz reka navodeći poziciju epruvete u reku, nakon čeka je potrebno uraditi prepakivanje reka tako da on ostane kompaktno. Za svaku epruvetu moguće je saznati njenu poziciju u reku. To je veoma korisno kada se želi pronaći određeni uzorak koji se odnosi na određenog pacijenta, prilikom hitnih slučajeva. Ova funkcionalnost se ne sme koristiti za prazna mesta u reku. Epruvete u reku potrebno je urediti po nekom kriterijumu. U svakom trenutku moguće je prikazati sadržaj reka na ekranu ali isto tako i sačuvati istoriju reka u tekstualni fajl. Na osnovu tekstualnog fajla moguće je napuniti rek epruvetama, ukoliko je potrebno da se zbog grešaka u urađenim analizama (analizator ispao iz kalibracije) svi testovi nad uzorcima ponovo izvrše.

Uzorak materijala se stavlja u epruvetu. Svaka epruveta se obeležava jedinstvenim barkodom (bar kod je sedmocifreni jedinstveni broj). Ukoliko se više epruveta odnosi na istog pacijenta, barkodovi na epruvetama sadrže iste prve 6 cifre dok je sedma cifra autoinkrementna cifra, čija je inicijalna vrednost 0. Ukoliko se jedna epruveta odnosi samo na jednog pacijenta onda je sedma cifra uvek nula. Epruvete u reku sortiraju se na osnovu barkodova. Za svaku epruvetu pamti se još i tačno vreme uzorkovanja materijala i test koji se obavlja.

U cilju testiranja biohemijskog analizatora neophodno je napuniti jedan rek sa 5 epruveta sa odgovarajućim uzorcima materijala a zatim istestirati sve moguće funkcionalnosti analizatora. Kako bi bili sigurni da je sve korektno implementirano i funkcionalno, a da se ne bi trošile dodatne epruvete, uzorkovan materijal i skupi reagensi neophodni za testove, moguće je pokrenuti simulaciju rada biohemijskog analizatora tako da on u tom slučaju umesto stvarnih epruveta u rekovima stavlja random generisane realne brojeve i na taj način radi i kalibraciju. Potrebno je u tom slučaju ispitati sve funkcionalnosti kao i kada se radi sa pravim uzorcima, uzorkovanog materijala u epruvetama, kako bi se kalibracija analizatora izvršila do kraja kao i da bi se istestirale sve funkcionalnosti analizatora.

PREDMETNI NASTAVNICI

Objektno orijentisano programiranje

– II kolokvijum (grupa D) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Potrebno je kreirati aplikaciju sa unos rasporeda u planeru. U planeru je potrebno pamtit i stavke kao i koliko je ukupno stavki trenutno uneto u njemu. Svaki planer ima inicijalno ograničenje koliko može imati unetih stavki, stim da se na zahtev korisnika može povećati broj stavki koje se mogu naći u planeru. Nova stavka se može dodati u planeru kao poslednja stavka ukoliko ima dovoljno mesta za njeno pamćenje. Iz planera je moguće izbaciti neku stavku ukoliko je greškom uneta ili više nije validna, jednostavnim izborom njenog rednog broja iz planera. Ukoliko stavka u planeru ne postoji na određenoj poziciji potrebno je odgovarajućom porukom obavestiti korisnika o tome. Planer treba da omogući korisniku da na efikasan način pronađe poziciju neke unete stavke u planeru ukoliko takva stavka u njemu postoji. U cilju preciznijeg prikazivanja rasporeda iz planera korisniku, neophodno je da planer obezbedi funkcionalnost koja će urediti stavke na osnovu nekog zadatog kriterijuma u rastućem redosledu. Kriterijum će zavisiti od toga šta će se kao stavka pamtit i u planeru. Stavke iz planera korisniku treba biti dostupne u svakom trenutku i na ekranu ali je potrebno obezbediti i čuvanje stavki iz planera u tekstualni fajl. Ukoliko su stavke iz planera sačuvane u nekom tekstualnom fajlu, korisnik može jednostavno učitati stavke u svoj planer iz fajla i nastaviti dalje sa upotrebom tako učitano rasporeda. U jednom planeru je moguće pamtit samo jedan tip stavki.

Kako je ispitni rok u toku, studentima je neophodan efikasan način planiranja polaganja ispita. U cilju da bi bili što efikasniji i uspešniji, mogu koristiti kreiran planer za pamćenje i planiranje polaganja ispita. Svaki ispit opisan je nazivom, šifrom, datumom i vremenom kada se polaže. Šifra ispita je jedinstveni četvorocifreni identifikator ispita koja se dodeljuje ispitu u trenutku kreiranja novog ispita. Prilikom uređivanja ispita u rasporedu, ispiti se uređuju u planer na osnovu datuma i vremena polaganja.

U cilju testiranja planera ispita neophodno je isprobati aplikaciju tako što će se u njoj uneti 10 različitih ispita i ispitati sve moguće funkcionalnosti planera ispita. Kako bi bili sigurni da je sve korektno implementirano i funkcionalno, potrebno je testirati planer na isti način kao i sa ispitima, ali samo da se sada kao stake iskoriste podaci koji su realni brojevi.

PREDMETNI NASTAVNICI

Objektno orijentisano programiranje

– II kolokvijum (grupa E) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktor),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Potrebno je kreirati aplikaciju koja će simulirati pokretnu proizvodnu traku. Na pokretnoj traci pamte se proizvodi određenog tipa. Na pokretnoj traci se može naći samo jedan tip proizvoda. Svaka pokretna traka inicijalno može da sadrži određeni broj proizvoda. Za specijalne potrebe moguće je stavljati i veći broj proizvoda (traka se može privremeno produžiti). U cilju postizanja efikasnosti u radu pokretne proizvodne trake, vrlo je bitno da se u svakom trenutku dobije informacija o broju proizvoda koji se nalaze na njoj u određenom vremenskom trenutku. Svaka proizvodna traka ima naziv. Proizvodi se dodaju na kraju pokretne trake samo ukoliko ima dovoljno mesta. Ukoliko se primeti neki defektan proizvod, potrebno je ukloniti takav proizvod sa pokretne trake. U svakom trenutku aplikacija treba da omogući da se dobije tačna pozicija nekog proizvoda na pokretnoj traci. Proizvod se može pronaći na pokretnoj traci na osnovu nekih vidljivih karakteristika. Kako bi proces proizvodnje bio efikasniji potrebno je urediti proizvode na pokretnoj traci po nekom kriterijumu. Aplikacija treba da ima mogućnost da sve proizvode na pokretnoj traci, na zahtev korisnika, prikaže na ekranu ali i da snimi trenutno stanje u tekstualni fajl. Moguće je i učitavanje proizvoda iz tekstualnog fajla kako bi simulacija pokretne trake bila jednostavnija.

Kako bi se proizvodna traka testirala, neophodno je na njoj stavljati čokolade. Za svaku čokoladu pamti se tip, da li je defektna, naziv, težina, barkod koji je jedinstven (sastoji se od 5 cifre i on se kreira prilikom pakovanja čokolade i nalazi se na vidljivom mestu, na omotaču) kao i tačno vreme proizvodnje. Da bi proizvodnja bila efikasnija, potrebno je urediti čokolade po vremenu proizvodnje. Potrebno je prikazati koliki je broj čokolada na pokretnoj traci čija je težina u zadatom opsegu vrednosti.

U cilju testiranja pokretne trake neophodno je isprobati pokretnu traku tako što će se na njoj staviti 5 čokolada i ispitati sve moguće funkcionalnosti aplikacije. Kako bi bili sigurni da je sve korektno implementirano i funkcionalno, potrebno je testirati aplikaciju na isti način kao i sa čokoladama, ali tako da se sada kao proizvodi iskoriste podaci koji su celi random brojevi i da se nad njima isprobaju sve funkcionalnosti.

PREDMETNI NASTAVNICI

Objektno orijentisano programiranje

– II kolokvijum (grupa F) –

27.1.2017. godine

Opšte napomene za sve klase:

- Atributi ne smeju biti javni,
- Kreirati sve potrebne javne metode koje će obezbediti da program radi (uključujući konstruktore i destruktore),
- Izbeći na svaki način dupliranje koda u celom projektu,
- Pridržavati se osnovnih principa objektno orijentisanog programiranja (nasleđivanje, modularnost, polimorfizam i enkapsulacija).

Manja rečna luka omogućava utovar i istovar kontejnera sa brodova. Luka poseduje jedan kran i jedan red obeleženih lokacija uz obalu reke na kojima se mogu naći kontejneri. Kako je luka u ekspanziji broj lokacija nije fiksna (raste nakon uređenja novih lokacija, ali se i smanjuje usled zatvaranja nekih lokacija na kraju reda zbog proširenja luke). Na jednoj lokaciji se može naći samo jedan kontejner. Pri istovaru sa broda kontejner se smešta na prvu praznu lokaciju u redu (ako to nije moguće aktivira se upozorenje glavnom menadžeru luke). Utovar na brod se obavlja na osnovu rednog broja lokacije na kojoj se kontejner nalazi (ako je lokacija pogrešna aktivira se upozorenje glavnom menadžeru luke). Na zahtev brodovlasnika luka može da odgovori na pitanje da li je traženi kontejner smešten u luci ili ne. Luka ima mogućnost da premešta kontejnere i uredi ih u rastući ili opadajući redosled po nekoj osobini. Luka može da prikaže izveštaj o svim kontejnerima koji se nalaze u luci. U slučaju opasnosti od poplave luka ima mogućnost da zapamti raspored kontejnera u luci i tovari ih na kamione koji ih prevoze u skladište. Skladište koristi isti sistem kao i luka. Ono prvo priprema tačno onoliko lokacija koliko je kontejnera i kontejnere sa kamiona raspoređuje na lokacije u zapamćenom redosledu (ako se pojavi greška aktivira se upozorenje glavnom menadžeru luke).

Luka je rešila da podeli svoj razvoj u dve faze. U prvoj fazi će da za smeštanje kontejnera koristiti samo njihove jedinstvene brojeve i obavljati sve moguće operacije na osnovu njih. U drugoj fazi će za svaki kontejner čuvati jedinstveni broj, preduzeće vlasnika i težinu kontejnera. Luka će moći da obavi sve predviđene operacije na osnovu ovih podataka o kontejnerima, pri tome će za uređenje u rastući redosled koristiti jedinstveni broj, a za uređenje u opadajući redosled koristiti težinu kontejnera. Na početku svake faze razvoja, luka će izvršiti testiranje rada tako što će istovariti nekoliko kontejnera sa brodova, zatim utovariti bar dva na brod, potražiti jedan postojeći i jedan nepostojeći kontejner, urediti kontejnere u rastući pa potom u opadajući redosled i izvršiti vežbu opasnosti od požara. Nakon svake operacije prikazati sadržaj luke.

PREDMETNI NASTAVNICI