

OOP - priprema za ispit

Opis:

Kompanija „Clean Code“ raspisala je natječaj i nastoji zaposliti programera koji dobro poznaje objektno orijentirane koncepte. Prilikom prijave na natječaj, bitno je priložiti rješenje problema u nastavku u obliku konzolne aplikacije, korištenjem C# programskog jezika.

Učilište „Obrazovanje na nivou“ ima potrebu za implementacijom aplikacije koja će omogućiti lakše ažuriranje podataka. Trenutno rješenje sastoji se od brojnih Excel datoteka koje su tipizirane prema kontekstu i daju razne mogućnosti koje je, ako je navedeno, treba moći zadržati.

Studente sačinjavaju sljedeći podaci: jedinstveni bročani identifikator koji se automatski generira, ime, prezime i tip (Izvanredni i Redovni). Studenti u sustavu mogu biti sortirani prema jedinstvenom identifikatoru (predodređeno i padajuće), ali i putem prezimena (rastuće). Prilikom unosa podataka o studentu, važno je unijeti cijeli redak odjednom. U slučaju izmjena studenta, važno je paziti se da se jedinstveni identifikator ne može promijeniti. Potrebno je implementirati mogućnost ispisa studenata. Također, izuzetno je bitno da se dva studenta, ako se pojave na raznim mjestima u sustavu moraju smatrati istima ako imaju isti identifikator. Napomena - u sljedećoj će se verziji koristiti i Hash tablice.

Podaci o profesorima, asistentima i dekanu dio su iste Excel datoteke, međutim neka su polja neispunjena. O profesorima je važno bilježiti podatke o imenu, prezimenu te samo jednom kolegiju koji drže. Nadalje, i o asistentima je važno bilježiti podatke o imenu, prezimenu te samo jednom kolegiju na kojem asistira. Napokon o dekanu je važno bilježiti podatke o imenu, prezimenu i tituli. Dekan ima mogućnost potpisivanja za entitete navedene u nastavku. Za profesore, asistente i dekane je također vrlo važno da moraju biti kreirani sa svim potrebnim podacima. Za profesore, asistente i dekane nisu važna sortiranja, ali je važan ispis.

O kolegijima je važno bilježiti šifru, naziv, broj ects bodova te da li je potvrđen od strane dekana. Broj ects bodova mora biti između 20 i 30, uključujući i te vrijednosti. Prilikom ispisa podataka o kolegijima, važno je ispisati podatke na ovaj način: „Šifra: 123, Naziv: OOP, Bodovi: 20ects, Potvrđen: da“. Dakle, izraz '20ects' smatra se jedinstvenim ispisom, a ne formatiranim. Za kolegije nisu važna sortiranja.

Napokon, postoje i dokumenti koji su određeni nazivom, tekstom i podatkom o tome da li su potpisani od strane dekana. Za njih nije potrebno implementirati sortiranje niti ispis.

U kontekstu poslovne logike, važno je napomenuti kako dekan može potpisivati dokumente, ali i odobravati kolegije. Dekanu može u bilo kojem momentu biti pružen dokument kako bi ga potpisao ili pak kolegij kako bi bio potvrđen. S gledišta dekana, on samo potpisuje i nevažno mu je da li je riječ o kolegiju ili dokumentu. Dakle, cijela funkcionalnost dekana u aplikaciji se svodi na potpisivanje, pa možemo smatrati da dokument i kolegij mogu biti *potpisani*. Prilikom potpisivanja kolegija, kolegiju se postavlja svojstvo da je uspješno potvrđen. Prilikom potpisivanja dokumenta, dokumentu se postavlja svojstvo da je uspješno potpisan. Moguće je da će dekan u budućnosti moći potpisivati i neke druge sadržaje.

Zadatak: (I1: 30 bodova, I2: 12 bodova, I3: 13 bodova)

Potrebno je implementirati različite hijerarhije klasa i implementacije sučelja, koristeći najbolje prakse objektno orijentiranih pristupa. Važno je prepoznati prilike za korištenje enumeracija, ekstenzijskih metoda i slično.

OOP - priprema za ispit

Demonstrirajte sve implementirane funkcionalnosti u glavnom programu na sljedeći način:

- Za demonstraciju nije potrebno od korisnika tražiti unos, već *hardkodirati* objekte hijerarhije
- Kreirati tri instance studenta, staviti ih u listu te ih ispisati
- Sortirati listu studentata na predodređeni način i ispisati ih
- Sortirati listu studenata prema prezimenu rastuće i ispisati ih
- Kreirati nepotvrđeni kolegij i ispisati ga
- Kreirati asistenta i profesora za spomenuti kolegij i ispisati ih
- Kreirati dekana i ispisati ga
- Kreirati jedinstvenu listu studenata, asistenata, profesora i dekana, ubaciti postojeće instance i ispisati koliko objekata istog tipa postoji u memoriji
- Kreirati nepotpisani dokument
- Napraviti i popuniti listu stvari za potpis od strane dekana (postojeći kolegij i dokument) te pozvati dekana da ih potpise
- Ispisati kolegij i dokument nakon potpisa