

## Auditorne vježbe 10.

### Klase Zaposlenik i Fakultet

Administrativnoj službi je potreban računarski program koji će voditi evidenciju o broju sati koji su pojedini zaposlenici proveli na poslu u toku radne sedmice, i na osnovu toga obračunati njihovu sedmičnu platu i naknadu za topli obrok.

Fakultet radi od ponedjeljka do petka. Normalno radno vrijeme je od 8:00 do 16:00, ali se pojedinim zaposlenicima, u zavisnosti od složenosti posla koji obavljaju (recimo, nastavnom osoblju) može dopustiti manje sedmično opterećenje, bez umanjenja plate za tu sedmicu. Stoga, za svakog zaposlenika postoje individualni podaci o njegovoj nominalnoj sedmičnoj plati kao i minimalnom sedmičnom opterećenju koje zaposlenik mora provesti na poslu da bi zaslužio puni iznos plate za tu sedmicu (ovo opterećenje ne smije biti veće od Zakonom predviđenog sedmičnog opterećenja od 40 sati). Zaposlenicima koji provedu manji broj sati u toku jedne sedmice od minimalnog sedmičnog opterećenja, odbija se za svaki puni sat ispod minimalne norme iznos jedne satnice, koja se dobija kao iznos sedmične plate podijeljen sa tipičnim sedmičnim opterećenjem od 40 sati. Pored toga, svi zaposlenici imaju pravo na dnevnu naknadu za topli obrok, koja je fiksna za sve zaposlenike (tj. ne zavisi od njihove plate). Zaposlenik stiče pravo na dnevnu naknadu za topli obrok za one dane u toku kojih je na poslu proveo veći broj sati od predviđene kvote, koja je također fiksna za sve zaposlenike. Na kraju, predviđena je i mogućnost dodatnog plaćanja prekovremenog rada. Ukoliko neki zaposlenik provede u toku jednog dana više od 8 sati na poslu, za svaki sat prekovremenog rada u toku tog dana isplaćuje mu se naknada u iznosu od 1.5 satnice za svaki sat prekovremenog rada. Treba napomenuti da se fakultet zaključava u 20:00, tako da do tog vremena svi zaposlenici moraju napustiti fakultet.

Za opisanu svrhu potrebno je razviti program u koji će se zasnivati na dvije klase nazvane “Zaposlenik” i “Fakultet”. Klasa “Zaposlenik” čuva podatke o jednom zaposleniku, dok je “Fakultet” kontejnerska klasa koja čuva podatke o kolekciji objekata tipa “Zaposlenik”. Ona treba da sadrži sljedeće elemente:

- a) Privatne attribute koji predstavljaju broj zaposlenika na fakultetu, maksimalan broj zaposlenika koji fakultet može zaposliti, dnevnu naknadu za topli obrok i minimalnu satnicu neophodnu za ostvarivanje prava na topli obrok (s obzirom da su ovi podaci isti za sve zaposlenike, njih nije neophodno čuvati za svakog zaposlenika posebno), kao i atribut koji služi za pristup dinamički alociranom nizu koji čuva pokazivače na objekte tipa “Zaposlenik”.
- b) Konstruktor sa tri parametra, koji predstavljaju maksimalan broj zaposlenika na fakultetu, dnevnu naknadu za topli obrok i minimalnu satnicu neophodnu za ostvarivanje prava na topli obrok. Ovaj konstruktor treba inicijalizirati odgovarajuće attribute i izvršiti neophodnu alokaciju prostora za čuvanje podataka o zaposlenicima. Broj zaposlenika se inicijalizira na 0, s obzirom da će se informacije o zaposlenicima dodavati naknadno.
- c) Destruktor, koji oslobađa sve resurse koji je objekat tipa “Fakultet” zauzeo tokom svog postojanja.
- d) Konstruktor kopije i preklopljeni operator dodjele koji omogućavaju sigurno kopiranje i međusobno dodjeljivanje objekata tipa “Fakultet” (što se tiče preklopljenog operatora dodjele, najbolje je prvo uništiti sadržaj odredišnog objekta a zatim iskopirati izvorni u odredišni objekat, s obzirom da bi svako drugo rješenje koje bi eventualno iskoristilo već postojeći prostor u odredišnom objektu bilo dosta komplicirano).

- e) Metodu koja kreira i evidentira novog zaposlenika. Parametri ove metode su evidencijski platni broj zaposlenika (taj broj jednoznačno određuje zaposlenika), njegovo ime i prezime, iznos njegove sedmične plate, kao i minimalno sedmično opterećenje koje zaposlenik treba provesti na poslu da bi zaradio puni iznos plate (ovi podaci se mogu razlikovati od jednog do drugog zaposlenika). Metoda treba da baci izuzetak ukoliko već postoji zaposlenik sa zadanim platnim brojem, ili ukoliko je dostignut maksimalni broj zaposlenika koji fakultet može primiti.
- f) Metodu koja vrši evidenciju dolaska zaposlenika na posao. Parametri metode su platni broj zaposlenika, kao i sat i minuta kada je zaposlenik došao na posao. Metoda treba da baci izuzetak ukoliko zaposlenik sa zadanim platnim brojem ne postoji, ukoliko vrijeme dolaska nije u intervalu od 8:00 do 16:00, ili ukoliko je njegov dolazak za taj dan već evidentiran (novo evidentiranje je moguće tek nakon evidencije odlaska).
- g) Metodu koja vrši evidenciju odlaska zaposlenika sa posla. Parametri su isti kao kod prethodne metode. Metoda treba da baci izuzetak ukoliko zaposlenik sa zadanim platnim brojem ne postoji, ukoliko vrijeme odlaska nije u intervalu od 8:00 do 20:00, ukoliko je vrijeme odlaska manje od vremena dolaska za taj dan, ili ukoliko je odlazak zaposlenika za taj dan već evidentiran (novo odjavljivanje je moguće tek nakon nove evidencije dolaska).
- h) Metodu koja vrši ispis podataka o zaposleniku. Parametar metode je platni broj zaposlenika, a metoda treba da ispiše ime zaposlenika, platni broj, broj normalno provedenih radnih sati (od početka evidentiranja), broj prekovremenih radnih sati (također od početka evidentiranja), platu za normalne radne sate (u skladu sa trenutnim stanjem, koje će biti kompletno tek na kraju sedmice), dodatak za prekovremeni rad, naknadu za topli obrok i ukupan iznos za isplatu (sve u skladu sa trenutnim stanjem).
- i) Metodu koja vrši ispis platne liste za sve zaposlenike na fakultetu (ova metoda bi se tipično trebala pozivati na kraju svake radne sedmice, jer će tek tada svi podaci biti kompletirani). Metoda treba za svakog zaposlenika ispisati iste podatke kao u predhodnoj metodi.
- j) Metodu koja vrši zaključivanje radne sedmice. Nakon poziva ove metode, sve informacije o broju provedenih radnih sati za sve zaposlenike vraćaju se na nulu, čime je sistem spreman za novo evidentiranje u toku sljedeće sedmice.

U gore navedenim specifikacijama nije ništa rečeno o tome šta treba sadržavati klasa "Zaposlenik". Kako je ona samo pomoćna klasa neophodna za funkcioniranje klase "Fakultet" (koja se na nju oslanja), nju možete dizajnirati kako god želite, pod uvjetom da to obezbijedi ispravno funkcioniranje klase "Fakultet" u skladu sa gore postavljenim zahtjevima.

Sve metode implementirajte izvan tijela klase, osim metoda čija implementacija zahtijeva jednu ili dvije naredbe. Obavezno napišite i mali testni program u kojem će se testirati sve navedene metode. Najbolje bi bilo da program manipulira sa podacima koji se unose sa tastature.