

Programski jezici - jun 2010.

1. Napraviti HTML stranicu koja korisniku omogućava da odabere jedan od gradova: Beograd, London, Njujork, Moskva i Tokio, a zatim u gornjem desnom delu prikazuje sat koji pokazuje tačan datum i vreme u tom gradu.
2. (a) Ako je data lista brojeva $[x_1, x_2, \dots, x_n]$, napisati PROLOG predikat koji izračunava $x_1 + 2 \cdot x_2 + \dots + n \cdot x_n$.
(b) Ako je data lista brojeva $[x_1, x_2, \dots, x_n]$, napisati PROLOG predikat koji izračunava $x_1 + x_2^2 + \dots + x_n^n$.
(c) Ako je data lista brojeva $[x_1, x_2, \dots, x_n]$, napisati PROLOG predikat koji gradi listu $[x_1, \underbrace{x_2, x_2, \dots, x_2}_2, \dots, \underbrace{x_n, \dots, x_n}_n]$.
3. Napisati JAVA/Swing aplikaciju koja može da se koristi u prodavnici tehničke opreme. Prodavnica prodaje različite vrste roba (računare, televizore i telefone). Za svaki komad opreme, poznata je šifra, naziv, proizvođač, osnovica za izračunavanje cene (pre poreza) i eventualno slika proizvoda. Uz ovo, za računare je poznat proizvođač procesora, a za televizore dijagonala ekrana. Različita oprema podležu različitim algoritmima izračunavanja poreza. Svakom pojedinačnom proizvodu pridružen je neki specifični algoritam oporezovanja. Jedan algoritam izračunava porez kao određeni procenat osnovne cene (npr. osnovica 100, poreska stopa 18%, onda je ukupna cena 118). Drugi algoritam ovo umanjuje za iznos poreske olakšice, ukoliko ukupna cena ne pređe određeni nivo (npr. porez je 18%, pri čemu se daje olakšica od 3% ukoliko je ukupna cena manja od 5000, pa ako je osnovica 1000, cena je 1150, dok ako je osnovica 4500, cena je 5310).

Podaci o proizvodima se pamte u tekstualnim datotekama. Aplikacija mora korisniku da omogući prikazivanje svih mogućih podataka o odabranim postojećim proizvodima, pri čemu korisnik može da izabere proizvod unošenjem njegovog naziva ili šifre u odgovarajuće tekstualno polje. Ukoliko postoji više proizvoda sa istim odabranim nazivom, istovremeno se prikazuju se podaci o svim. Ukoliko se prilikom unošenja šifre pogreši na najviše jednoj karakterskoj poziciji (dakle, uneta šifra ne postoji, ali postoji šifra koja se razlikuje na tačno jednoj karakterskoj poziciji od unete), uz odgovarajuće upozorenje, korisniku se prikazuju proizvodi sa sličnim šiframa unetoj. Takođe, moguće je tabelarno prikazati najznačajnije informacije o svim proizvodima (šifra, naziv, ukupna cena). Dalje, aplikacija mora da omogući korisniku mogućnost unošenja u aplikaciju podataka o novom proizvodu.

Voditi računa o tome da aplikacija bude što udobnija za korišćenje. Voditi računa o objektno-orijentisanom dizajnu aplikacije — neophodno je lako dodavanje novih vrsta proizvoda i novih algoritama izračunavanja poreza.