

Stručno vijeće računalstva

Program: *Tehničar za računalstvo
Novi strukovni kurikulum
(041624)*

Razred: *četvrti*

Razredni odjeli: *ABCFGH*

NAPREDNO I OBJEKTNO PROGRAMIRANJE 2

1. ispitna laboratorijska vježba – C2 grupa

Zadatak 1:

U maloj računovodstvenoj firmi, programer je zadužen za izradu jednostavnog sustava za praćenje cijena popravaka računalnih uređaja. Svaki popravak treba imati cijenu, pa programer odlučuje kreirati klasu pod nazivom *****Račun***** koja će imati jedno svojstvo *****Cijena***** koje će pohraniti vrijednost cijene popravka.

U aplikaciji, programer omogućuje unos vrijednosti cijene popravka i ispis te vrijednosti kroz non-static metodu. Ovaj jednostavan sustav omogućava firmi da lako unosi i prati cijene popravaka za svakog korisnika.

****Definicija:****

- ****Klasa:**** `Račun``
- ****Svojstvo:**** `Cijena`` (tip podatka: brojčani tip, poput ``int`` ili ``float``, ovisno o preciznosti)
- ****Metoda:**** `IspisCijene`` (non-static metoda koja ispisuje cijenu)
- ****Objekt:**** `račun``

****Tipovi podataka:****

- ****Svojstvo `Cijena`:** Brojčani tip podataka (``double``).
- ****Metoda `IspisCijene`:** Ova metoda neće imati povratnu vrijednost (tip ``void``), a njezina svrha je ispisati vrijednost svojstva *****Cijena*****.

Zadatak 2:

U istoj firmi, klijent želi pratiti cijene za dva popravka. Programer stvara novu verziju klase *****Račun*****, koja sada može sadržavati jedno svojstvo *****Cijena*****. Za upravljanje s više objekata, programer implementira statičku metodu *****IspisSvihCijena***** koja ispisuje cijene oba popravka.

Nakon što se unesu cijene za oba računa, metoda će ispisati te cijene bez potrebe za pristupanjem svakom objektu posebno. To omogućuje jednostavniji pregled svih popravaka bez potrebe za kreiranjem dodatnih metoda.

****Definicija:****

- ****Klasa:**** `Račun`
- ****Svojstvo:**** `Cijena`
- ****Metoda:**** `IspisSvihCijena` (static metoda koja ispisuje cijene oba objekta, te cijena kojeg računa je veća i za koliko)
- ****Objekti:**** `račun1`, `račun2`

****Tipovi podataka:****

- ****Svojstvo `Cijena`:** Brojčani tip podataka (`double`).
- ****Metoda `IspisSvihCijena`:** Ova statička metoda neće imati povratnu vrijednost (tip `void`), a koristi dva objekta tipa `Račun` za ispis njihovih svojstava ****`Cijena`****.

Zadatak 3:

Kako bi proširio sustav, klijent sada želi dodati i datum kada je račun izdano. Programer mora modificirati klasu ****`Račun`**** da uključuje dva svojstva:

1. ****`Cijena`**** – cijena popravka.
2. ****`DatumIzdavanja`**** – datum kada je račun izdan, korištenjem tipa ****`DateTime`****.

Uz to, programer implementira metodu koja ispisuje obje vrijednosti: cijenu i datum izdavanja. Za datum izdavanja koristi ****`DateTime`****, koji automatski pohranjuje trenutni datum i vrijeme kada je račun izdan.

****Definicija:****

- ****Klasa:**** `Račun`
- ****Svojstva:****
 - `Cijena` (tip podatka: brojčani tip poput `int`, `float` ili `double`).
 - `DatumIzdavanja` (tip podatka: `DateTime`).
- ****Metoda:**** `IspisPodataka` (non-static metoda koja ispisuje oba svojstva).
- ****Objekti:**** `račun1`, `račun2`

****Tipovi podataka:****

- ****Svojstvo `Cijena`:** Brojčani tip podataka (npr. `float` ili `double` za precizne brojke).
- ****Svojstvo `DatumIzdavanja`:** `DateTime`, koji pohranjuje datum i vrijeme.
- ****Metoda `IspisPodataka`:** Ova metoda ispisuje dva svojstva objekta te koliko je dana od datuma računa do početka iduće kalendarske godine, bez povratne vrijednosti (`void`).

Zadatak 4:

Zamislite da firma koristi jednostavan sustav za pohranu svojih izvještaja. Programer treba stvoriti program koji omogućuje:

Kreiranje direktorija za pohranu izvještaja, ako direktorij ne postoji (provjeri postojanje direktorija).

Kreiranje datoteke unutar tog direktorija, ako datoteka ne postoji (provjeri postojanje datoteke).

Upisivanje tekstualnog izvještaja u datoteku.

Ispisivanje sadržaja datoteke ako već postoji.

Upute:

Korisnik unosi naziv direktorija ("Izvjestaji").

Program mora provjeriti postoji li direktorij. Ako ne postoji, kreira ga.

Zatim korisnik unosi naziv datoteke (npr. "izvjestaj.txt").

Program provjerava postoji li ta datoteka u direktoriju. Ako ne postoji, kreira je.

Program omogućava korisniku da unese tekst koji će biti upisan u datoteku.

Na kraju, ako datoteka postoji, program će ispisati njezin sadržaj na ekranu.