

Programske paradigme i jezici
Završni ispit – 21.06.2013.
By: Zvoc :)

1. zadatak (6 bodova):

U namespaceu System.Linq definirana je metoda proširenja ForEach, koja radi nad IEnumerable<T>, nad svakim elementom izvršava funkciju predanu u argumentu. Ona kao podlogu koristi foreach petlju.

Napisati metodu proširenja MyForEach koja radi nad List<T> i kao podlogu koristi for petlju. Metoda, uz samu listu, prima i delegat s argumentima (element, index) koji izvršava nad svakim elementom liste. Metoda kao povratnu vrijednost ima istu listu koja je predana na ulaz.

Npr., primjer poziva koji ispisuje listu cijelih brojeva:

```
List<int> lista = new List<int>() {1,5,6,8,24};  
lista.MyForEach((x, i) => Console.WriteLine ((i + 1).ToString() + “. element je “ +  
x.ToString()));
```

2. zadatak (6 bodova):

Ostvariti pojednostavljeni objektni model škole sa administracijskog stajališta. Škola ima profesore i učenike. Učenici su podijeljeni u razrede, a svaki razred ima svog razrednika (što ne znači i da je svaki profesor razrednik nekom razredu). Škola ima i ravnatelja koji je jedan od profesora. Ostvarenim entitetima potrebno je predvidjeti ostale attribute.

Napisati LINQ upit koji u formatu “Prezime, Ime (OIB)” vraća popis svih učenika kojima je razrednik trenutni ravnatelj, ako takvi postoje.

Paziti da objektni model ima smisla, odnosno, da nema nelogičnosti.

3. zadatak (5 bodova):

U programskom jeziku C# napisati metodu koja paralelno u datotečnom sustavu traži datoteke čije ime sadrži znakove predane argumentom (contains) i vraća njihovu apsolutnu putanju. Metoda treba proći kroz sve poddirektorije predanog direktorija.

Prototip metode: List<string> FindFiles (string absoluteDirectory, string pattern);
Pritom koristiti metode string[] GetFiles (string absoluteDirectory) i string[] GetSubdirectories (string absoluteDirectory) koje vraćaju imena datoteka, odnosno poddirektorija u relativnom obliku.

Npr., za poziv metode FindFiles ("C:\\", "test"), povratne vrijednosti bi mogle biti "C:\\Users\\Student\\Desktop\\test.jpg", "C:\\plan_protesta.txt", itd.

4. zadatak (7 bodova):

a) U programskom jeziku F# napisati funkciju koja prima listu cijelih, obrće je i izbacuje proste elemente iz nje.

Primjer: ako je zadana lista [1;2;3;4;5;6], funkcija vraća [6;4] (obrnuta lista i izbačeni prosti

elementi 1, 2, 3, 5.

Napomena: pretpostavite da postoji gotova funkcija koja provjerava je li broj prost.

b) Objasniti i na vrlo kratkom primjeru ilustrirati Curryjev postupak (eng. Currying)

5. zadatak (3 boda):

Objasniti princip objektnog dizajna Otvoreno/Zatvoreno (eng. Open/Closed). Ilustrirati na proizvoljnom programskom odsječku kršenje tog principa.

6. zadatak (3 boda):

Objasniti pojam kohezije postupaka. Navesti vrste kohezije, te ih klasificirati po poželjnosti.