

Međuispit iz Programskih paradigmi i jezika

3. svibnja 2016. godine

Ispit nosi ukupno 15 bodova i piše se 60 minuta.

Na pitanja je potrebno odgovoriti kratko i precizno.

1. zadatak (5 bodova)

Objasnite što su to **metode proširenja** i po čemu se razlikuju od klasičnih metoda, tzv. „metoda instanci“. U programskom jeziku C# napišite odsječak programa u kojem se **definiraju** i **pozivaju** jedna klasična i jedna metoda proširenja.

2. zadatak (10 bodova)

U upravi grada žele pratiti kulturna zbivanja u gradu. Kulturna zbivanja se sastoje od događaja koji su opisani nazivom, lokacijom i vremenom početka. Događaji mogu biti koncert, predstava, izložba i festival. Koncert je opisan brojem mjesta i popisom izvođača, dok je predstava opisana brojem mjesta, popisom glumaca, redateljem i trajanjem u minutama. Izložba je događaj koja je opisana popisom autora i temom izložbe. Festival je događaj koji se sastoji od više drugih događaja i za koje se dodatno evidentira trajanje u danima.

- a) Potrebno je ostvariti opisani model kulturnih zbivanja u programskom jeziku C#
- b) Napisati LINQ upit koji vraća naziv događaja i vrijeme početka za sve događaje koji se još nisu održali
- c) Napisati LINQ upit koji vraća listu festivala sortiranu prema broj događaja uzlazno

Rješenja:

1.

Metode proširenja (en. Extension methods) su statičke metode koje se pozivaju kao članske (kao „metode instanci“). Umjesto `metoda(objekt);` možemo pisati `objekt.metoda();` Omogućavaju nam proširenje tj. nadogradnju funkcionalnosti javnih sučelja postojećih klasa (naravno, iz tih metoda i dalje možemo raditi samo s postojećim public objektima ili metodama dane klasa).

Da bi se to omogućilo, metode proširenja definiramo ovako:

```
Namespace ExtensionMethods
{
    public static class MyExtensions
    {
        public static int WordCount(this String str)
        {
            return str.Split(new char[] { ' ', '.', '?' },
                             StringSplitOptions.RemoveEmptyEntries).Length;
        }
    }
}
```

Pozivamo:

```
using ExtensionMethods;
...
string s = "Hello Extension Methods";
int i = s.WordCount();
```

ključna stvar (i jedina razlika koju metode proširenja imaju je `this` uz prvi parametar – veza prema objektu nad kojim je pozvana)

2.

a)

```
public class KulturnaZbivanja {
    public string Grad { get; set; }
    public ICollection<Dogadjaj> Dogadjaji { get; set; }
}

public abstract class Dogadjaj {
    public string Naziv { get; set; }
    public string Lokacija { get; set; }
    public DateTime VrijemePocetka { get; set; }
}
```

```

public class Koncert : Dogadjaj {
    public int BrojMjesta { get; set; }
    public ICollection<String> Izvodjaci { get; set; }
}

public class Predstava : Dogadjaj {
    public int BrojMjesta { get; set; }
    public ICollection<String> Glumci { get; set; }
    public string Redatelj { get; set; }
    public int TrajanjeMin { get; set; }
}

public class Izlozba : Dogadjaj {
    public ICollection<String> Autori { get; set; }
    public string Tema { get; set; }
}

public class Festival : Dogadjaj {
    public int TrajanjeUDanima { get; set; }
    public ICollection<Dogadjaj> FestivalskiDogadjaji { get; set; }
}

```

b)

```

KulturniDogadaji kd = new KulturniDogadaji();
IEnumerable<object> popisNeodrzanihDogadaja =
    kd.Dogadjaji.Where(d => d.VrijemePocetka > DateTime.Now).
        Select(k => new { k.Naziv, k.VrijemePocetka });

//ili
IEnumerable<object> popisNeodrzanihDogadaja2 =
    from dogadjaj in kd.Dogadjaji
    where dogadjaj.VrijemePocetka > DateTime.Now
    select new { dogadjaj.Naziv, dogadjaj.VrijemePocetka };

```

c)

```

List<Festival> sortiraniFestivali =
    kd.Dogadjaji.Where(f => f is Festival).
        OrderBy(f => ((Festival)f).FestivalskiDogadjaji.Count).
        Select(f => f as Festival).ToList();

//ili
List<Festival> sortiraniFestivali2 =
    (from dogadjaj in kd.Dogadjaji
     where dogadjaj is Festival
     orderby ((Festival)dogadjaj).FestivalskiDogadjaji.Count ascending
     select dogadjaj as Festival).ToList();

```