

Ime i prezime: _____

Matični broj: _____

Ispitna pitanja

1. [3 bod.] Ako su dani statička metoda `RunZ1()` i njen željeni izlaz prema izlistanju, definirajte klasu `Wind` kako bi se metoda mogla u potpunosti ispravno izvesti. Koristiti potpuna svojstva. Za objekte klase `Wind` onemogućiti naknadno mijenjanje stanja izvan objekta.

```
1 public static void RunZ1()
2 {
3     Wind w1 = new Wind(31.5, "North");
4     Wind w2 = new Wind();
5     Console.WriteLine($"Speed: {w2.SpeedKmh} km/h, description: {w2.Description}");
6 }
```

```
1 Speed: 0 km/h, description: Calm
```

2. [3 bod.] Za klasu `Wind` iz zadatka 1 definirajte metodu koja za predanu vrijednost temperature u stupnjevima Celzijusa računa i kao rezultat vraća osjet temperature pod utjecajem vjetrova (engl. *wind chill*). Osjet temperature računa se prema izrazu $w = 0.6215T - 11.37V^{0.16} + 0.3965TV^{0.16} + 13.12$, gdje T predstavlja temperaturu, a V brzinu vjetrova u *km/h*. Ako je predana temperatura veća od 10 stupnjeva, tada vjetar ne utječe na osjet.
3. [3 bod.] U statičkoj klasi `WeatherStatistics` definirajte statičku metodu koja za predano polje objekata proizvoljne duljine tipa `Wind` iz zadatka 1 pronalazi i kao rezultat vraća broj ekstremnih vjetrova. Ekstremnim vjetrom smatra se svaki onaj čija brzina odstupa za barem 20% od prosječne brzine svih vjetrova unutar polja. Ako je dano popunjeno polje vjetrova naziva *ElNino*, pozvati napisanu metodu korištenjem danog polja.



4. [3 bod.] Ako je dana klasa `Portion` prema izlistanju, definirajte vlastitu klasu `FoodTracker` koja omogućuje pohranu porcija u obliku liste. Za klasu `FoodTracker` definirajte metodu `Track()` koja omogućuje pohranu porcije te virtualnu metodu koja omogućuje izračun ukupne zarade.

```
1 class Portion
2 {
3     public Portion(decimal price, double weight, double kcal)
4     {
5         Price = price; Weight = weight; Kcal = kcal;
6     }
7
8     public decimal Price { get; private set; }
9     public double Weight { get; private set; }
10    public double Kcal { get; private set; }
11 }
```

5. [3 bod.] Za klasu `FoodTracker` iz zadatka 4 definirajte metodu koja za predan delegat tipa `Func<Portion, bool>` iz liste praćenih porcija uklanja sve porcije koje zadovoljavaju kriterij definiran delegatom. Napišite kod koji testira danu metodu predajom argumenta koji će filtrirati sve porcije čija je cijena po kilogramu veća od 30 kuna. Na raspolaganju vam je objekt imena *tracker* već popunjan porcijama.
6. [3 bod.] Iz klase `FoodTracker` izvedite klasu `NutritionalTracker` s atributima koji predstavljaju pragove kalorične vrijednosti i težine porcije. Definirajte parametarski konstruktor. Prepišite metodu za izračun ukupne zarade tako na ukupnu zaradu za svaku porciju koja je težinom ispod, a kaloričnom vrijednošću iznad odgovarajućeg praga pridoda još donaciju države u iznosu od deset posto vrijednosti porcije.

Koristiti programski jezik C#. Poštovati pravila enkapsulacije. Svi podimenici imenika `System` su uključeni. Potpisati i predati sve ispitne listove uključujući i ovaj. Ispit se piše 90 minuta.

7. [3 bod.] Napišite statičku metodu koja za polje `FoodTracker` objekata i zadan iznos tražene zarade provjerava koliko njih uspijeva doseći zadanu zaradu. U slučaju ju da niti jedan ne može ostvariti, podići iznimku vlastitog tipa sa stanjem za iznos tražene zarade kojem je moguće pristupiti. Testirati metodu tako da se za potrebe poziva instancira `NutritionalTracker` i popuni s 10 nasumičnih `Portion` instanci. Ispravno rukovati iznimkama i u slučaju iznimke ispisati preko objekta iznimke dosegnutu zaradu.

▼▼▼

8. [3 bod.] Ako je dana klasa prema izlistanju

```
1 class Location
2 {
3     public double Latitude { get; private set; }
4     public double Longitude { get; private set; }
5     public Location(double latitude, double longitude)
6     {
7         Latitude = latitude; Longitude = longitude;
8     }
9 }
```

definirajte sučelje `ILocationManager` koje će za klase koje ga implementiraju omogućiti

- Za predanu trenutnu lokaciju i listu susjednih lokacija pronalazak najbliže lokacije među susjednima.
 - Za predanu lokaciju, listu susjednih lokacija i željenu udaljenost kreiranje nove liste koja sadrži samo one susjedne lokacije unutar željene udaljenosti.
9. [3 bod.] Definirajte klasu `EuclideanManager` koja implementira sučelje iz zadatka 8 tako da se za provjeru udaljenosti koristi Euklidska udaljenost određena izrazom $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

▼▼▼

10. [3 bod.] Ako je dana klasa prema izlistanju i lista imena *articles*, definirajte LINQ upite koji će omogućiti

```
1 public class Article
2 {
3     public string Source { get; private set; }
4     public int CharCount { get; private set; }
5     public List<string> Keywords { get; private set; }
6 }
```

- izdvajanje svih članaka kojima je izvor *HINA*, a koji među ključnim riječima imaju riječ *Trump*,
- dohvaćanje svih jedinstvenih ključnih riječi u člancima duljim od 1000 znakova,
- izračun prosječne duljine članaka koji sadrže ključnu riječ *Potres*.

Koristiti programski jezik C#. Poštovati pravila enkapsulacije. Svi podimenici imenika `System` su uključeni. Potpisati i predati sve ispitne listove uključujući i ovaj. Ispit se piše 90 minuta.