Tehtävä 3

LINQ (ja tapahtumaohjelmointi)

Tee C#-komentokehoteohjelma. Ohjelmassa on seuraavat luokat (Program-luokan lisäksi) Department, Employee, MenuItemEventArgs, MenuItem, Data ja Application. Varusta kaikki luokat näkyvyysmääreellä public.

Department

Luokassa on automaattiset ominaisuudet Id (kokonaisluku), Name (merkkijono), Employees (Employee-olioiden lista) ja EmployeeCount (kokonaisluku). Viimeksi mainitussa on vain get-metodi palauttaen ominaisuuden Employees sisältämän listan olioiden lukumäärän.

Parametrittomassa konstruktorissa alustetaan ominaisuuteen Employees tyhjä lista. Toisessa konstruktorissa on kaksi parametria id (kokonaisluku) ja name (merkkijono). Tämä konstruktori kutsuu edellä mainittua konstruktoria ja alustaa parametrien arvot vastaavien ominaisuuksien arvoiksi.

Ylikirjoitettu metodi ToString palauttaa merkkijonon, jossa olio esitetään merkkijonona muodossa {Name} ({EmployeeCount}), esim. Osasto 5 (23).

Employee

Luokassa on automaattiset ominaisuudet Id (kokonaisluku), FirstName, LastName (merkkijonoja), Department (tyyppiä Department), StartDate (DateTime), EndDate ja DateOfBirth (DateTime?-tyyppiä). Näistä ominaisuus Id sisältää vain get-metodin.

Ominaisuus Name sisältää vain get-metodin palauttaen ominaisuuksien LastName ja FirstName arvot yhdistettynä ja välilyönnillä eroteltuna.

Ominaisuus Age (kokonaisluku) sisältää vain get-metodin ja palauttaa iän laskettuna ominaisuuden DateOfBirth avulla. Jos ominaisuudessa DateOfBirth ei ole arvoa, palautaan arvona nolla.

Ominaisuus Salary (double) kapseloi kenttämuuttujan _salary. Set-metodissa tarkastetaan, ettei arvoksi anneta negatiivista arvoa. Jos näin tehdään, nostetaan poikkeus virhetekstillä ”Negatiivinen palkka”. Get-metodi palauttaa kenttämuuttujan _salary arvon pyöristettynä kahteen desimaaliin.

Konstrukorissa on parametrit id, first, last, dob ja salary, joiden arvot sijoitetaan vastaaviin ominisuuksiin Id, FirstName, LastName, DateOfBirth ja Salary. Lisäksi konstruktori alustaa ominaisuuteen StartDate arvoksi kuluvan päivän ja ominaisuuteen EndDate null-arvon.

Ylikirjoitettu metodi ToString palauttaa merkkijonon, jossa olio esitetään merkkijonona muodossa {Id} {FirstName} {LastName}, esim. 234 Maija Aalto.

MenuItemEventArgs

Luokka periytyy luokasta System.EventArgs ja se sisältää yhden automaattisen ominaisuuden ItemId (kokonaisluku).

MenuItem

Luokka sisältää automaattiset ominaisuudet Id (kokonaisluku) ja Name (merkkijono).

Luokassa on tapahtuma nimeltä ItemSelected, jonka tyyppinä on System.EventHandler<MenuItemEventArgs>.

Luokassa on metodi Select, jonka algoritmissa suoritetaan tapahtuman ItemSelected mahdollisesti sisältämä metodi antaen argumenteiksi MenuItem-olion (this) ja uuden MenuItemEventArgs-olion, jonka ItemId-ominaisuuden arvona on ominaisuuden Id arvo.

Ylikirjoitettu metodi ToString palauttaa merkkijonon, jossa olio esitetään merkkijonona muodossa {Id}. {Name}, esim. 1. Hae kaikki.

Data

Luokka Data on staattinen ja se simuloi datalähdettä. Tässä luokassa on vain kolme julkista elementtiä: staattiset muuttujat Departments ja Employees sekä metodi GenerateData. Luokan koodi on annettu valmiiksi alempana.

Muuttujien Departments ja Employees sisältämiin oliokokoelmiin kohdistetaan Application-luokassa LINQ-kyselyjä.

Metodi GenerateData populoi edellä mainittujen staattisten muuttujien sisältämät listat olioilla. Department-olioita tehdään 15 kappaletta, joista jokaiseen luodaan 1-99 Employee-oliota (määrä on satunnainen). Jokainen Employee-olio saa satunnaisen nimen, syntymäajan ja palkan.

Tämän luokan koodi on annettu valmiiksi (ja se on kopioitavissa):

```
using System;
using System.Collections.Generic;
using System.Linq;

namespace T3
{
    public static class Data
    {
        private static Random r = new Random();

        public static List<Department> Departments;
        public static List<Employee> Employees;

        public static void GenerateData(int depts = 15, int maxempsindept = 100)
        {
            Departments = new List<Department>();
            Employees = new List<Employee>();
            for (int i = 1; i <= depts; i++)
            {
                Departments.Add(GenerateDepartment(i, r.Next(1,maxempsindept)));
            }
        }

        private static Department GenerateDepartment(int id, int emps)
        {
            Department ret = new Department(id, $"Osasto {id}");
            Employee emp;
            for (int i = 1; i <= emps; i++)
            {
                emp = GenerateEmployee();
                ret.Employees.Add(emp);
                emp.Department = ret;
            }
```

```
            return ret;
        }
        private static Employee GenerateEmployee()
        {
            int id = 1;
            if (Employees.Count > 0)
            {
                id += Employees.Max(e => e.Id);
            }
            DateTime start = new DateTime(1950, 1, 1),
                end = new DateTime(1997, 12, 31);
            Employee emp = new Employee(
                id,
                GenerateFirstName(),
                GenerateLastName(),
                GenerateDate(start,end),
                GenerateSalary()
                );
            Employees.Add(emp);
            return emp;
        }
        private static string GenerateFirstName()
        {
            string[] names =
{"Antti","Aulis","Aulikki","Bertta","Eemeli","Eetu","Esko","Hannele","Hannu","Heikki","Harri
","Heli","Helena",
                                        "Iines","Ilkka","Ilmari", "Ida", "Irene",
"Jaakko","Jari","Juha","Jukka","Jaana","Juhani","Kalevi","Kikka","Kaarina","Kalle","Kauko",
                                        "Kimmo", "Liisa","Leevi","Lilli",
"Lumi","Martti","Matti","Mauno","Melina","Miisa","Merja","Marja","Niilo","Nina","Niina","Nin
na",

"Oona","Olavi","Oskari","Pentti","Panu","Pertti","Petri","Paula","Pauliina","Pirkko",
"Raimo", "Rauli", "Riikka", "Ritva", "Raili",
                                        "Sara", "Saara", "Sauli","Sakari","Seppo","Taina",
"Tiina","Tuuli","Tuula","Timo","Tino","Tomas","Tuomas","Unto","Uolevi",
                                        "Veikko","Veera","Vesa","Väinö","Yrjö"};
            return names[r.Next(0,names.Length)];
        }
        private static string GenerateLastName()
        {
            string[] part1 = { "Aalto", "Joki", "Meri", "Järvi", "Virta", "Vuori", "Lehto",
"Leppi", "Koivu", "Salo", "Niemi", "Lahti", "Laiho", "Haka", "Ala", "Ylä", "Väli", "Keski"
};
            string[] part2 = { "", "nen", "talo", "mäki", "pää", "kari", "la" };
            return $"{part1[r.Next(0, part1.Length)]}{part2[r.Next(0, part2.Length)]}";
        }
        private static DateTime GenerateDate(DateTime begin, DateTime end)
        {
            int days = (end - begin).Days;
            return begin.AddDays(r.Next(0,days+1));
        }
        private static double GenerateSalary(double min = 1000, double max = 10000)
        {
            return Math.Round(min + r.NextDouble()*(max-min),2);
        }
    }
}
```

Application

Luokka Application on staattinen sisältäen staattisen muuttujan Menu, joka on tyypiltään MenuItem-olioiden lista.

Staattinen ja geneerinen metodi WriteResult, jossa on tyyppiparametri T ja parametrit itemid (kokonaisluku) ja result (T-tyyppisten olioiden lista). Metodin algoritmissa tulostetaan tulostaulu, jossa on sarakkeina parametrin result sisältämän listan olioiden ominaisuusnimet. Tämän metodin algoritmi on annettu valmiiksi:

```
string row;
//otsikkorivit
WriteLine();
WriteLine(Menu.Where(mi => mi.Id == itemid).First().Name.ToUpper());
WriteLine("-".PadRight(18 * result[0].GetType().GetProperties().Length + 2, '-'));
//sarakeotsikkorivit
if (result.Count > 0)
{
    row = "";
    foreach (PropertyInfo property in result[0].GetType().GetProperties())
    {
        row += $"{property.Name}".PadRight(16) + " | ";
    }
    WriteLine(row);
}
WriteLine("-".PadRight(18 * result[0].GetType().GetProperties().Length + 2, '-'));
//datarivit
foreach (object item in result)
{
    row = "";
    foreach (PropertyInfo property in item.GetType().GetProperties())
    {
        row += $"{property.GetValue(item, null).ToString()}".PadRight(16) + " | ";
    }
    WriteLine(row);
}
WriteLine("-".PadRight(18 * result[0].GetType().GetProperties().Length + 2, '-'));
WriteLine();
Write("Paina Enter jatkaaksesi.");
ReadLine();
```

Staattinen metodi InitializeMenu luo staattiseen muuttujaan Menu MenuItem-olioiden listan ja populoi sen seuraavilla olioilla:

```
new MenuItem() { Id = 1, Name = "50-vuotiaat työntekijät"},
new MenuItem() { Id = 2, Name = "Osastot yli 50 henkilöä"},
new MenuItem() { Id = 3, Name = "Sukunimen työntekijät"},
new MenuItem() { Id = 4, Name = "Osastojen isoimmat palkat"},
new MenuItem() { Id = 5, Name = "Viisi yleisintä sukunimeä"},
new MenuItem() { Id = 6, Name = "Osastojen ikäjakaumat"}
```

Lisäksi metodissa asetetaan jokaiselle edellä tehdylle MenuItem-oliolle tapahtuman ItemSelected käsittelijämetodi, jonka algoritmissa

- haetaan/kysellään haluttu data LINQ-kyselyjen avulla luokan Data oliokokoelmista Departments ja Employees.

- **50-vuotiaat työntekijät**: haetaan kaikki Employee-oliot, joiden ikä (Age) on 50 ja muodostetaan niistä anonyymit oliot, joissa on ominaisuuksina Nimi, Ikä ja Osasto (sisältäen työntekijän osaston nimen).
- **Osastot yli 50 henkilöä**: haetaan kaikki Department-oliot, joiden työntekijämäärä (EmployeeCount) on yli 50, järjestetään nämä oliot laskevaan järjestykseen työntekijälukumäärän mukaan ja muodostetaan olioista anonyymeja olioita, joissa ominaisuuksina on Id, Nimi ja Vahvuus (eli työntekijämäärä).
- **Sukunimen työntekijät**: pyydetään käyttäjältä sukunimi (merkkijono) ja haetaan kaikki Employee-oliot, joiden sukunimi (LastName) on annettu nimi. Muodostetaan näistä olioista anonyymit oliot, joissa on ominaisuudet Id ja Nimi (, jonka arvona on Employee-oliosta saatu sukunimi katenoituna ja välilyönnillä eroteltuna etunimen kanssa.
- **Osastojen isoimmat palkat**: haetaan Department-olioiden sisältämistä Employee-olioista jokaisen osaston työntekijöiden suurin palkka (Salary). Tässä kannattaa käyttää laajennusmetodia SelectMany:

```
Data.Departments.SelectMany(d => d.Employees,
          (d,e) => new { Osasto = d.Name, Palkka = e.Salary })
```

  - ensimmäisenä argumenttina on metodi, joka palauttaa Employee-olioiden listan
  - toisena argumenttina on ensimmäisen argumentin metodin saama Department-olio ja ensimmäisen argumentin metodin palauttaman listan Employee-olio parina. Tämä toisena argumenttina annettava metodi palauttaa oliopareista muodostettuja anonyymeja olioita.
  - Nämä anonyymit oliot ositetaan Osasto-ominaisuuden arvojen perusteella (laajennusmetodi GroupBy) ja osituksesta muodostetaan lopputulokseen anonyymeja olioita, joissa on ominaisuuksina Osasto ja Maksimipalkka.
- **Viisi yleisintä sukunimeä**: haetaan Employee-olioiden listalta osittamalla kaikki oliot sukunimen perusteella. Osituksesta tuotetaan anonyymejä olioita, joissa on ominaisuuksina Sukunimi ja Lkm (= osajoukkoon kuuluvien olioiden määrä laajennusmetodilla Count). Lopuksi anonyymit oliot järjestetään laskevaan järjestykseen Lkm-ominaisuuden mukaan (laajennusmetodi OrderByDescending) ja otetaan järjestyksessä 5 ensimmäistä oliota (laajennusmetodi Take).
- **Osastojen ikäjakaumat**: haetaan Departments-olioiden listalta anonyymejä olioita, joissa on ominaisuudet Nimi, Alle30v (lukumäärä Department-olion Employees-ominaisuuden listan sisältämistä Employee-olioista, joiden Age on pienempi kuin 30), Välillä30_50v (samoin kuin edellä, paitsi Age on välillä 30-50) ja Yli50v (samoin kuin edellä, paitsi Age on suurempi kuin 50).

- tulostetaan haettu data metodin WriteResult avulla antaen argumenteiksi tapahtumakäsittelijän saaman MenuItemEventArgs-olion ItemId-ominaisuus ja edellä kyselty oliokokoelma.
- käsittelijämetodi kannattaa antaa lambda-lausekkeena seuraavaan tapaan (- huomaa, että alla LINQ-kyselyä ei tarkoituksella ole annettu = kohta …):

```
Menu[0].ItemSelected += (obj, a) => {
    var result = ...
    WriteResult(a.ItemId, result);
};
```

Staattinen metodi Initialize sisältää algoritmin, jossa suoritetaan datan muodostaminen kutsumalla staattista metodia Data.GenerateData. Lisäksi kutsutaan metodi InitializeMenu.

Staattinen metodi ReadFromMenu palauttaa kokonaisluvun, joka saadaan käyttäjän syötteestä seuraavasti:

- tyhjennetään komentokehoikkuna (metodi System.Console.Clear)
- tulostetaan teksti Vaihtoehdot ja riveittäin staattisen muuttujan Menu sisältämien MenuItem-olioiden merkkijonoesitykset
- tulostetaan kehoteteksti Valitse (0 = lopetus):
- jos käyttäjän syöte ei ole kelvollinen eli kokonaisluku väliltä 0 – 6 (, missä luku 6 on suurin MenuItem-olioiden Id-ominaisuuksien arvoista), nostetaan poikkeus virhetekstillä Vastaus ei ole kelvollinen.
- jos syöte on kelvollinen, palautetaan se muunnettuna kokonaisluvuksi.

Staattinen metodi Run, jossa kutsutaan metodi Initialize ja toistuvasti pyydetään käyttäjältä syötettä metodin ReadFromMenu avulla, kunnes käyttäjä syöttää nollan. Kun käyttäjä on valinnut vaihtoehdon eli syöttänyt luvun väliltä 1-6, kutsutaan lukua vastaavan kokoelmasta Menu löytyvän MenuItem-olion metodia Select (- joka aiheuttaa MenuItem-oliolle tapahtuman ItemSelected ja sitä kautta tapahtumakäsittelijän suorittumisen).

Program

Luokan Program sisältämässä Main-metodissa kutsutaan metodi Application.Run.

Ohjelman toiminta eri valikkotoiminnoissa – huomaa, että Employee-olioiden arvot ovat satunnaisia:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus):
```

Valitaan vaihtoehto 1:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 1

50-VUOTIAAT TYÖNTEKIJÄT
-----------------------------------------------------
Nimi              | Ikä             | Osasto          |
-----------------------------------------------------
Välikari Veikko   | 50              | Osasto 2        |
Vuorila Ilmari    | 50              | Osasto 3        |
Virtala Nina      | 50              | Osasto 3        |
Niemipää Melina   | 50              | Osasto 4        |
Hakapää Helena    | 50              | Osasto 4        |
Ala Sauli         | 50              | Osasto 4        |
Hakala Saara      | 50              | Osasto 5        |
Koivunen Yrjö     | 50              | Osasto 7        |
Yläpää Ilkka      | 50              | Osasto 8        |
Lahtinen Lilli    | 50              | Osasto 8        |
Jokikari Raili    | 50              | Osasto 9        |
Välila Jaakko     | 50              | Osasto 10       |
Lahtitalo Kalevi  | 50              | Osasto 12       |
Niemipää Petri    | 50              | Osasto 12       |
Koivunen Juha     | 50              | Osasto 13       |
-----------------------------------------------------

Paina Enter jatkaaksesi.
```

Valitaan vaihtoehto 2:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 2

OSASTOT YLI 50 HENKILÖÄ
-----------------------------------------------------
Id                | Nimi            | Vahvuus         |
-----------------------------------------------------
8                 | Osasto 8        | 98              |
1                 | Osasto 1        | 91              |
3                 | Osasto 3        | 80              |
2                 | Osasto 2        | 76              |
14                | Osasto 14       | 75              |
5                 | Osasto 5        | 70              |
10                | Osasto 10       | 66              |
13                | Osasto 13       | 62              |
4                 | Osasto 4        | 58              |
9                 | Osasto 9        | 57              |
12                | Osasto 12       | 52              |
-----------------------------------------------------

Paina Enter jatkaaksesi.
```

Valitaan vaihtoehto 3:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 3

Anna sukunimi: Aalto

SUKUNIMEN TYÖNTEKIJÄT
--------------------------------------
Id              | Nimi              |
--------------------------------------
219             | Aalto Oskari      |
396             | Aalto Panu        |
486             | Aalto Melina      |
487             | Aalto Jaakko      |
545             | Aalto Merja       |
--------------------------------------

Paina Enter jatkaaksesi.
```

Valitaan vaihtoehto 4:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 4

OSASTOJEN ISOIMMAT PALKAT
--------------------------------------
Osasto          | Maksimipalkka     |
--------------------------------------
Osasto 1        | 9977,23           |
Osasto 2        | 9684,53           |
Osasto 3        | 9969,64           |
Osasto 4        | 9944,41           |
Osasto 5        | 9838,33           |
Osasto 6        | 9781,35           |
Osasto 7        | 9964,18           |
Osasto 8        | 9944,04           |
Osasto 9        | 9948,56           |
Osasto 10       | 9947,91           |
Osasto 11       | 9996,17           |
Osasto 12       | 9971,16           |
Osasto 13       | 9784,4            |
Osasto 14       | 9977,7            |
Osasto 15       | 9441,97           |
--------------------------------------

Paina Enter jatkaaksesi.
```

Valitaan vaihtoehto 5:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 5

VIISI YLEISINTÄ SUKUNIMEÄ
-------------------------------------
Sukunimi         | Lkm              |
-------------------------------------
Meritalo         | 15               |
Jokitalo         | 15               |
Alakari          | 13               |
Välila           | 13               |
Virta            | 12               |
-------------------------------------

Paina Enter jatkaaksesi.
```

Valitaan vaihtoehto 6:

```
Vaihtoehdot
1. 50-vuotiaat työntekijät
2. Osastot yli 50 henkilöä
3. Sukunimen työntekijät
4. Osastojen isoimmat palkat
5. Viisi yleisintä sukunimeä
6. Osastojen ikäjakaumat
Valitse (0 = lopetus): 6

OSASTOJEN IKÄJAKAUMAT
-------------------------------------------------------------------------
Nimi             | Alle30v          | Välillä30_50v    | Yli50v           |
-------------------------------------------------------------------------
Osasto 1         | 17               | 33               | 41               |
Osasto 2         | 15               | 31               | 30               |
Osasto 3         | 15               | 34               | 31               |
Osasto 4         | 11               | 22               | 25               |
Osasto 5         | 13               | 31               | 26               |
Osasto 6         | 4                | 3                | 6                |
Osasto 7         | 5                | 14               | 13               |
Osasto 8         | 11               | 48               | 39               |
Osasto 9         | 11               | 21               | 25               |
Osasto 10        | 17               | 23               | 26               |
Osasto 11        | 3                | 11               | 9                |
Osasto 12        | 15               | 28               | 9                |
Osasto 13        | 8                | 30               | 24               |
Osasto 14        | 13               | 38               | 24               |
Osasto 15        | 2                | 1                | 2                |
-------------------------------------------------------------------------

Paina Enter jatkaaksesi.
```