# Skaitymas iš failo ir rašymas į failą

## Bibliotekos

Šios pamokos tikslas - išmokti skaityti duomenis iš failo, rašyti duomenis į failą ir naudotis įvairiomis .Net platformos bibliotekomis.

## Užduotis

**Nuskaityti failą longestNameInput.txt ir parašyti tris metodus:**

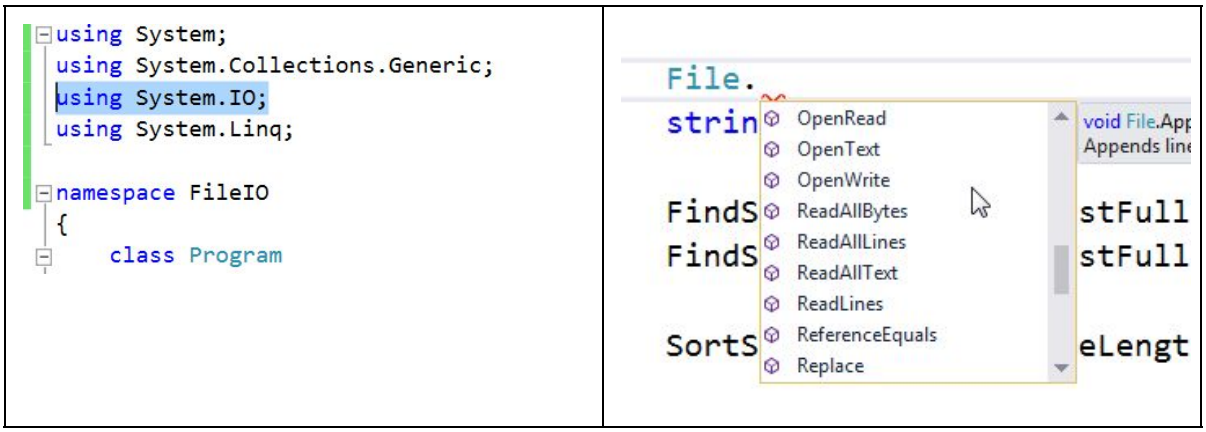
**FindSportsmenWithLongestFullname** - randa ilgiausią vardą, naudojant *string[]* duomenų struktūrą ir *for* ciklą, išveda jį į failą.
**FindSportsmenWithLongestFullnameLikeAPro** - randa ilgiausią vardą, naudojant *List<string>* duomenų struktūrą ir LINQ komponentą, išveda jį į failą.
**SortSportsmenByFullnameLength** - surikiuoja sportininkus pagal vardo ilgį, išveda surikiuotą sąrašą į failą.

Užduoties metu jums reikės prisiminti, kas yra metodai. Išmoksite, kaip panaudoti .Net bibliotekų funkcijas. Palyginsite, kaip skirtingai galite parašyti tą patį metodą - su paprasta duomenų struktūra *string[]* ir su galingesne *List<string>*. Darydami užduotį, pabandykite debuginti savo kodą, kaip tai darėte praėjusią pamoką.

## Failo skaitymas

**string** input = **File.ReadAllText**(@"D:\Ruta\Projects\FileIO\FileIO\longestNameInput.txt");

**File.ReadAllText**(*keliasIkiFailo*) gyvena System.IO bibliotekoje. Nepamirškite jos pridėti, kai dirbate su failais. Mes naudosim tik vieną File metodą, bet pažiūrėkite, kiek dar jų yra.
Kaip manot, kada naudoti ReadAllText ir kada ReadAllLines?




## Rašymas į failą

**string** output = "Jonas, Petras, Antanas";
**File.WriteAllText**(@"D:\Ruta\Projects\FileIO\FileIO\longestNameV1.txt", output );

## String operacijos

### Split

**string** names = "Jonas, Petras, Antanas";
**string**[] namesArray = names.**Split**(',');

*string* tipo kintamajam *Split(kazkoksSimbolis)* metodą naudojam, kai norim jį sukarpyti tose vietose, kur yra koks nors simbolis - tą simbolį įrašom kaip metodo *Split(kazkoksSimbolis)* parametrą. Šiuo atveju namesArray bus:
{"Jonas", " Petras", " Antanas"}

### Trim

**string** name = "  Petras";
**string** nameWithoutWhitespace = name.**Trim**();
//dabar nameWithoutWhitespace reikšmė yra "Petras"

*Trim()* metodas ištrina tarpus string kintamojo pradžioje ir pabaigoje

### Length

**string** name = "  Petras";
**int** nameLength = name.Length; // 6

Atkreipkit dėmesį, kad šiuo atveju nereikia skliaustų!

### Join

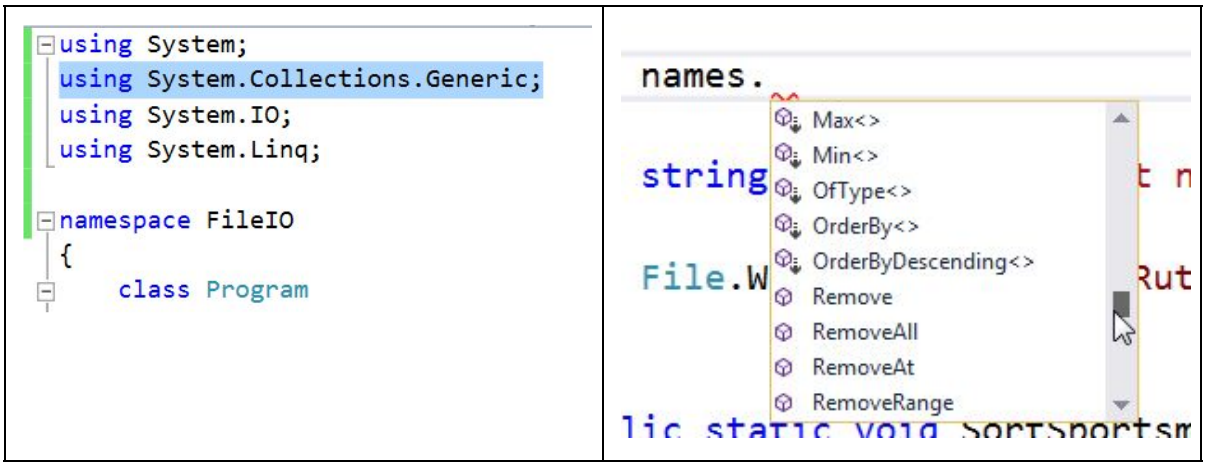
**string**[] namesArray = new **string**[3] {"Matt", "Joanne", "Robert"};
string output = **String.Join**(",", namesArray);
//output taps "Matt,Joanne,Robert"

*String.Join* priima du parametrus - skirtuką ir string tipo masyvą. Tada masyvo elementus sujungia į vieną string tipo kintamąjį, tarp jų padėdamas skirtuką.

## List duomenų struktūra

**string**[] namesArray = new [] {"Jonas", "Petras", "Antanas"};
**List<string>** names = namesArray.**ToList**()

*List* duomenų struktūra gyvena *System.Collections.Generic* “namespace”;
*List* nuo masyvo skiriasi tuo, kad turi kintamą skaičių elementų, dėl to juos pridėti ar išimti yra patogiau. *string[]* galima paversti į *List<string>*, kaip parodyta pavyzdyje.



## List duomenų struktūra ir LINQ komponentas

**List<string>** list = new **List<string>**();
list.Add("Jonas");
list.Add("Petras");
list.Add("Antanas");

//rikiavimas pagal žodžio ilgį
**List<string>** orderedList = list.**OrderByDescending**(name => name.Length).ToList();
*name => name.Length* yra vadinama lambda išraiška (lambda expression). Ją galima įsivaizduoti kaip mini metodą - tai kas rodyklės '=>' kairėje yra parametrai, o dešinėje - rezultatas.

//pasiima pirmą žodį iš surūšiuoto sąrašo - "Antanas"
string **firstNameInList** = list.**First**();

Paprastai kalbant, LINQ tai metodai, kuriuos gali naudoti darbui su tokia duomenų struktūra kaip *List* (yra tokių ir daugiau). Kai nori surūšiuoti elementus, kvieti **Order()** arba **OrderByDescending().** Dar keletas:

- First()
- Last()
- Remove(name => name == "Antanas")
- Find(name => name == "Antanas")

## Prisiminkim

### Ciklai

**string**[] namesArray = new [] {"Jonas", "Petras", "Antanas"};
**for (**int i = 0; i < namesArray.Length ; i++**)**
{
    … // expressions
}

### Kodo sąlyginio vykdymo komanda - If

**if (**name.Length > 0**)**
{
    … // expressions
}

### String'ų formavimas

**string** output = "Vardas: " + " Jonas";