ktu informatikos fakultetas 1922

T07. Objektų elgsenos aprašymas ir programos struktūrizavimas

2 ak. val.

ktu informatikos fakultetas 1922

Temos klausimai

1. Metodas kaip programos struktūrizavimo priemonė, pasikartojančių dalių sutraukimas į metodą.
2. Objektas kaip programos struktūrizavimo priemonė.
3. Objektų rinkinio charakteristikų radimo algoritmai: didžiausios (mažiausios) reikšmės su apribojimais paieška, dviejų didžiausių (mažiausių) reikšmių paieška.
4. `ArrayList` klasė ir naudojimas.

ktu informatikos fakultetas 1922

Objektas ir metodas – programos struktūrizavimo priemonės

Prisiminkime: OP bruožai

- Bet ką galima laikyti **objektu**.
- Skaičiavimai vykdomi, kviečiant objektų **metodus**.
- Kiekvienas objektas gali būti **sudarytas iš kitų objektų**.
- Kiekvienas objektas – tai atstovas **klasės**, kuri išreiškia **bendras** kažkokios objektų grupės **savybes**.
- Klasė apibrėžia objekto **elgseną** (metodai). Tos pačios klasės objektai gali atlikti tik tuos pačius veiksmus.

Problemos atvaizdavimas srities objektais (1/5)

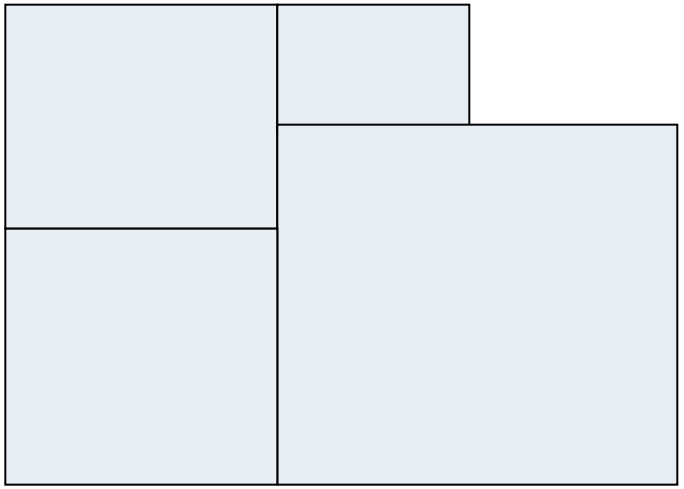
Projektuojant programą tam tikram uždaviniui spręsti reikia užduotį **skaidyti** į lengviau suvokiamas dalis, atvaizduoti uždavinio srities **objektais**.

Tarkime, užduoties formuluotė prasideda taip:

> Butą sudaro keturi kambariai...

Mums pažįstami srities objektai yra butas ir kambarys:

```
class Butas,
class Kambarys...
```



Problemos atvaizdavimas srities objektais (2/5)

Kiekvieną objektą reikia aprašyti jį nusakančiomis **charakteristikomis**.

Kokios charakteristikos pilnai (mūsų uždaviniui) aprašo `Butą`?

Ar užteks pasakyti, kad butas – tai 4 kambariai? Šiuo atveju – 4, o bendru?

Iš pastarosios analizės aišku, kad 4 kambarių butas yra objektas (konkretu), o bendru atveju bute gali būti įvairus kambarių skaičius – klasė, manipuliuojanti kambarių rinkiniu.

Klasėje **`Butas`** bus objektų **`Kambarys`** rinkinys (masyvas).

Problemos atvaizdavimas srities objektais (3/5)

Uždavinio skaidymas į smulkesnes dalis ir problemos padengimas objektais priklauso nuo konkrečios užduoties ir, iš dalies, nuo numatomos galimybės plėsti programos funkcionalumą.

Kaip charakterizuosime **`Kambarį`**?

Jei kambarys stačiakampis, tai užteks kambario ilgio ir pločio. O jeigu kambarys ne stačiakampis?

`Kambarį` gali sudaryti sienos, jis gali turėti langus ir duris. Ar reikia numatyti, kad kambarys bus **sudėtinis objektas**?

Problemos atvaizdavimas srities objektais (4/5)

ktu informatikos fakultetas 1922

Numatyti kiek ateityje reikės plėsti programos funkcionalumą yra sudėtinga. Atskirų objektų aprašymą gali tekti keisti, tačiau **reikia siekti, kad objektai būtų kuo mažiau priklausomi vienas nuo kito ir nuo programos**.

Jeigu mūsų užduotis suskaičiuoti plotą nesikeis, o keisis tik pradinių duomenų nusakymo būdas:

- **`Kambarys`** ne stačiakampis (atkarpomis aprašytas daugiakampis)
- **`Kambarys`** – sudėtinis objektas, apribotas sienomis (sienų duomenys)

Šie pasikeitimai neturėtų įtakoti visos programos, o tik **`Kambarį`**.

Problemos atvaizdavimas srities objektais (5/5)

Projektiniai uždavinio sprendimai neturi užkirsti kelio plėsti programos funkcionalumą ateityje.

Tarkime, **`Kambarys`** charakterizuojamas ilgiu ir pločiu ir turi plotą.

Padarykime plotą skaičiuojama charakteristika – metodu, kuriame bus ilgio ir pločio sandauga.

Atsiradus poreikiui **`Kambarį`** charakterizuoti kitaip (ne stačiakampis), mums tereikės pakeisti ploto skaičiavimą metode, tačiau `Kambario` naudojimas (ploto sužinojimas) nepasikeis. Programa (**`Butas`**) kreipsis į tą patį metodą.

Pavyzdys programos struktūrizavimo problemai

Duotas krepšinio komandos žaidėjų sąrašas:

žaidėjo numeris, pavardė, vardas, žaidėjo ūgis.

Duotas vienerių rungtynių protokolas, kuriame eilutėmis surašytas žaidėjų laikas, praleistas aikštelėje:

žaidėjo numeris, kėlinys, išėjimo į aikštelę laikas, praleistas laikas.

Išėjimo į aikštelę laikas ir praleistas laikas nurodyti minutėmis. Išėjimo į aikštelę laikas nurodytas nuo kėlinio pradžios.

Surasti, kurie žaidėjai buvo aikštelėje nurodytu laiku (kėlinys ir minutė).

Kokią programą parašytumėte šiam uždaviniui?

Programos struktūrizavimas (1/4)

```
static void Main(string[] args)
{
  // Keletas masyvų
  // Eilė paprastų kintamųjų
  // Visi veiksmai iš eilės
}
```

Keičiantis duomenų nusakymo ir jų pateikimo programai būdui, ir pan., pakeitimus reikės daryti **visoje programoje**.

Programos struktūrizavimas (2/4)

```
class Zaidėjas;
class ProtokoloEilutė;
static void Ivesti...(...);  // Įveda žaidėjus; įveda protokolą
static void Spausdinti...(...);// Išveda žaidėjus; išveda protokolą
static void Sudaryti(...);
static void Main(string[] args)
{
  // Keli objektų rinkiniai (masyvai), eilė paprastų kint.
  // Kreipiniai į metodus
  // Metodai turi panašių ir daug parametrų,
  // nes visiems reikia tų pačių rinkinių
}
```

Keičiantis duomenų pateikimo būdui reikės pakeisti tik **`Ivesti...(...)`** metodus.

Keičiantis rezultatų pateikimo terpei reikės keisti tik **`Spausdinti...(...)`** ir pan.

Keičiantis duomenų nusakymo būdui reikės daryti pakeitimus **visuose metoduose ir klasėse**.

Programos struktūrizavimas (3/4)

```
class Žaidėjas;
class ProtokoloEilutė;
class Komanda;        // Klasė, sauganti Žaidėjus
class Protokolas;     // Klasė, sauganti ProtokoloEilutes
static void Ivesti...( /*parametras - objektas*/ );
static void Spausdinti...( /*parametras - objektas*/ );
static void Sudaryti( /*trys parametrai*/ );
static void Main(string[] args)
{
  // Du klasės Komanda objektai
  // Klasės Protokolas objektas
  // Kreipiniai į metodus
}
```

Programos struktūrizavimas (4/4)

- Pasikeitus žaidėją (protokolo eilutę) nusakantiems duomenims reikės keisti tik **`Žaidėjo`** (**`ProtokoloEilutės`**) klasę.
- Pasikeitus rinkinio saugojimo būdui (kol kas žinome tik masyvą) reikės keisti **`Komandos`** (**`Protokolo`**) klasę.
- Pasikeitus duomenų ir rezultatų pateikimo terpei reikės keisti metodus **`Įvesti()`** ir **`Spausdinti()`**.
- Pakeitus (papildžius) užduoties skaičiavimo veiksmus reikės keisti (papildyti nauju) metodą **`Sudaryti()`**.
- **`Main()`** reikės papildyti objektais ir kreipiniais į metodus tik papildžius užduotį naujais skaičiavimo veiksmais.

ktu informatikos fakultetas 1922

Koncepcijų atskyrimas

Tinkamai struktūrizuotoje programoje turi būti atskiriamos šios koncepcijos:

- duomenis ir rezultatus nusakantys srities objektai;
- jų rinkiniais manipuliuojantys konteineriai;
- duomenų pateikimą ir rezultatų atvaizdavimą atitinkamose terpėse realizuojantys metodai arba klasės, esant sudėtingoms terpėms, pvz., grafinė aplinka;
- viską valdantis ir surišantis `Main()` metodas.

Objektų charakteristikų radimo algoritmai: didžiausio (mažiausio) su apribojimais ir dviejų didžiausių (mažiausių) paieška

Didžiausios reikšmės su apribojimais paieška masyve

```
int[] Mas = new int[7] { 8, -7, 10, 5, -8, 4, 5 };
```

1. Paieškai atlikti reikalingi *du ciklai*:
   - *pirmame cikle* surandama **pirma tinkama** (apribojimas) masyvo elemento reikšmė, pvz., pirmas neigiamas skaičius,
   - *antrame cikle* toliau ieškoma didžiausia reikšmė **įvertinant jos tinkamumą**, pvz., tikrinama, ar neigiama.

2. Paieškos rezultatai:
   - reikšmė surasta, pvz., neigiamos reikšmės indeksas ,
   - reikšmės nėra, pvz., masyve nėra neigiamų reikšmių, indeksas -1,
   - abiems avejais formuojamas suradimo požymis (`true`, `false`).

Didžiausios reikšmės su apribojimais paieškos masyve metodas

ktu informatikos fakultetas 1922

```
static bool DidžiausiasNeig(int[] Mas, int kiek, out int maxInd)
{
    maxInd = -1;
    int i;
    // Pirmo neigiamo (apribojimas) skaičiaus paieška
    for (i = 0; i < kiek; i++)
    if (Mas[i] < 0) {
        maxInd = i;
        break;
    }
    // Didžiausio neigiamo skaičiaus tolimesnė paieška
    for (; i < kiek; i++)
        if (Mas[i] < 0)
            if (Mas[i] > Mas[maxInd])
                maxInd = i;
    if (maxInd > -1)
        return true;
    return false;
}
```

Metodo su "šalutiniu" efektu pavyzdys

Kaip pašalinti "šalutinį" efektą?

Didžiausios reikšmės su apribojimais paieškos masyve metodo panaudojimas

Ar būtinas lyginimas?

```
int maxInd;
if (DidžiausiasNeig(Mas, kiek, out maxInd) == true)
{
    Console.WriteLine("Didžiausias neigiamas: {0}", Paz[pazMaxInd]);
}
else
{
    Console.WriteLine("Didžiausio neigiamo NĖRA!");
}
. . .
```

Rezultatas ekrane: **`Didžiausias neigiamas: -7`**

		Masyvo dydis: kiek = 7;						
Mas	------>	**8**	**-7**	**10**	**5**	**-8**	**4**	**5**
		0	1	2	3	4	5	6

Dviejų didžiausių reikšmių paieška masyve

```
int[] Mas = new int[7] { 8, 7, 10, 5, 8, 4, 5 };
```

1. Reikalingi du kintamieji reikšmėms saugoti:
   pirmai didžiausiai **max1**,
   antrai didžiausiai **max2**.

2. Kintamiesiems suteikiamos pradinės reikšmės:
   max1 = 8, max2 = 7.

3. Peržiūrimos visos kitos sąrašo reikšmės:
   jeigu **Mas[i] > max1**,
      tuomet **max2 = max1, max1 = Mas[i]**
   kitaip
      jeigu **Mas[i] > max2**,
         tai **max2 = Mas[i]**

Dviejų didžiausių reikšmių paieškos masyve metodas

```
static bool DuMaxS(int[] Mas, int kiek, out int max1, out int max2)
{
    max1 = -1;
    max2 = -1;
    if (kiek < 2) return false;
    //--- Pradinių reikšmių suteikimas ----------------------
    if (Mas[0] > Mas[1]) {
        max1 = Mas[0]; max2 = Mas[1];
    }
    else {
        max1 = Mas[1]; max2 = Mas[0];
    }
    //--- Paieška -------------------------------------------
    for (int i = 2; i < kiek; i++) {
        if (Mas[i] > max1) {
            max2 = max1;
            max1 = Mas[i];
        }
        else if (Mas[i] > max2)
                max2 = Mas[i];
    }
    return true;
}
```

Masyvo dydis: kiek = 7;

Mas	---->	8	7	10	5	8	4	5
		0	1	2	3	4	5	6

Metodą pritaikius masyvui **Mas(kiek)** rezultatas: **10 8**

ktu informatikos fakultetas 1922

Demo

Demonstracinė programa MVS: didžiausios reikšmės su apribojimu ir dviejų didžiausių reikšmių paieška skaičių masyve.

Objekto išrinkimas pagal kriterijų

Rasti objektų rinkinyje (masyve):

- didžiausią reikšmę
- didžiausios reikšmės indeksą
- didžiausios reikšmės objektą
- . . .

Pavyzdys: užduotis (1/9)

Prisiminkime užduotį:

Faile "Studentai.txt" duota informacija apie vienos grupės studentų pažymius (pvz., kontrolinio darbo įvertinimai): *studento pavardė ir vardas*, *pažymys*.

Reikia rasti:

1. geriausią įvertinimą gavusį studentą.
2. du geriausius įvertinimus gavusius studentus.

Pavyzdys: klasė **Studentas** (2/9)

```
class Studentas
{
    private string pavVrd;  // studento pavardė ir vardas
    private int pazym;      // pažymys (įvertinimas)
    public Studentas(string pavv, int pazym)
    {
        pavVrd = pavv;
        this.pazym = pazym;
    }
    public string ImtiPavv() { return pavVrd; }
    public int ImtiPazym()   { return pazym;  }
}
```

Pavyzdys: klasės **Studentas** objektų masyvas (3/9)

```
const int Cn = 100;   // maksimalus studentų skaičius
static void Main(string[] args)
{
    int kiek;                    // studentų skaičius
    Studentas[] Studentai = new Studentas[Cn];
    ...
}
```

```
Jonaitis Jonas;          8;
Petraitis Petras;        7;
Antanaitis Antanas;      10;
Giedraitis Giedrius;     5;
Onaitytė Ona;            8;
Juozaitis Juozas;        4;
Ramunaitė Ramunė;        5;
```

Pavyzdys: klasės **Studentas** objektų charakteristikos (4/9)

Klasės **Studentas** objektų charakteristikos:

- pavardė ir vardas
- pažymys (įvertinimas)

```
class Studentas
{
    private string pavVrd;  // studento pavardė ir vardas
    private int pazym;      // pažymys (įvertinimas)
    public Studentas(string pavv, int pazym)
    {
        pavVrd = pavv;
        this.pazym = pazym;
    }
    public string ImtiPavv() { return pavVrd; }
    public int ImtiPazym()   { return pazym;  }
}
```

Pavyzdys: geriausias studento pažymys (5/9)

```
static int MaxPazymys(Studentas[] Studentai, int kiek)
{
    int maxPaz = Studentai[0].ImtiPazym();
    for (int i = 1; i < kiek; i++)
        if (Studentai[i].ImtiPazym() > maxPaz)
            maxPaz = Studentai[i].ImtiPazym();
    return maxPaz;
}
...
Console.WriteLine("Geriausias pažymys: {0, 2:d}",
                          MaxPazymys(Studentai, kiek));
...
```

Ekrane matysite: **`Geriausias pažymys: 10`**

Pavyzdys: geriausio studento indeksas (6/9)

ktu informatikos fakultetas 1922

```
static int MaxIndex(Studentas[] Studentai, int kiek)
{
  int maxInd = 0;
  for (int i = 1; i < kiek; i++)
    if (Studentai[i].ImtiPazym() > Studentai[maxInd].ImtiPazym())
       maxInd = i;
  return maxInd;
}
...
Console.WriteLine("Geriausio indeksas: {0, 3:d}",
                          MaxIndex(Studentai, kiek));
...
```

Ekrane matysite: `Geriausio indeksas: 2`

Pavyzdys: geriausias studentas (7/9)

```
static Studentas MaxStudentas(Studentas[] Studentai,
                              int kiek)
{
    Studentas max = Studentai[0];
    for (int i = 1; i < kiek; i++)
        if (Studentai[i].ImtiPazym() > max.ImtiPazym())
            max = Studentai[i];
    return max;
}
...
Studentas studMax = MaxStudentas(Studentai, kiek);
Console.WriteLine("Geriausias studentas: {0} {1}",
                  studMax.ImtiPavv(), studMax.ImtiPazym());
...
```

Ekrane matysite: **Geriausias studentas: Antanaitis Antanas 10**

Pavyzdys: du geriausi studentai (8/9)

```
static bool DuMaxS(Studentas[] Studentai, int kiek,
                                          out Studentas max1, out Studentas max2)
{
    max1 = new Studentas("", 0);
    max2 = new Studentas("", 0);
    if (kiek < 2) return false;
    //--- Pradinių reikšmių suteikimas ----------------------
    if (Studentai[0].ImtiPazym() > Studentai[1].ImtiPazym()) {
        max1 = Studentai[0]; max2 = Studentai[1];
    }
    else {
        max1 = Studentai[1]; max2 = Studentai[0];
    }
    //--- Paieška -------------------------------------------
    for (int i = 2; i < kiek; i++)
    {
        if (Studentai[i].ImtiPazym() > max1.ImtiPazym()) {
           max2 = max1;
           max1 = Studentai[i];
        }
        else if (Studentai[i].ImtiPazym() > max2.ImtiPazym())
                max2 = Studentai[i];
    }
    return true;
}
```

Pavyzdys: du geriausi studentai (9/9)

```
Studentas studMax1;
Studentas studMax2;
if (DuMaxS(Studentai, kiek, out studMax1, out studMax2))
{
   Console.WriteLine("Geriausias pirmas: {0} {1}",
             studMax1.ImtiPavv(), studMax1.ImtiPazym());
   Console.WriteLine("Geriausias antras: {0} {1}",
             studMax2.ImtiPavv(), studMax2.ImtiPazym());
}
else
{
   Console.WriteLine("Dviejų geriausių NĖRA!");
}
```

Ekrane matysite:

```
Geriausias pirmas: Antanaitis Antanas 10
Geriausias antras: Jonaitis Jonas 8
```

ktu informatikos fakultetas 1922

Demo

Demonstracinė programa MVS: geriausio studento paieška, dviejų geriausių studentų paieška objektų masyve.

ktu informatikos fakultetas 1922

Atsiminkite

Nustatyti, ar du realūs skaičiai yra tiksliai lygūs ne visada galima tokiu būdu:

```
double a, b;
...
if (a == b) Console.WriteLine("Lygūs\n");
else        Console.WriteLine("Nelygūs\n");
```

Ar du **realūs skaičiai yra lygūs** nustatoma tam tikru tikslumu, pavyzdžiui trijų ženklų po kablelio tikslumu:

```
if (Math.Abs(a - b) <= 0.001)
     Console.WriteLine("Lygūs\n");
else Console.WriteLine("Nelygūs\n");
```

Klasė `ArrayList`

Apibrėžimas

- **`ArrayList`** klasė yra skirta saugoti įvairių **bazinių** tipų (int, double, string, ...) arba vartotojo sukurtų tipų (pvz., **class**) objektams (objektų masyvui).
- **`ArrayList`** saugoja nuorodas į objektus.
- **`ArrayList`** masyvas gali dinamiškai kisti, t.y. didėti arba mažėti, papildant masyvą naujais elementais arba šalinant elementus (skirtingai nuo **`array`** masyvo).
- **`ArrayList`** turi daug metodų veiksmams su jame saugomais elementais atlikti (paminėsime tik kelis).
- **`ArrayList`** naudojimui reikalinga **direktyva**:

```
using System.Collections;
```

ktu informatikos fakultetas 1922

Aprašymas

```
using System.Collections;
...
ArrayList Vardas = new ArrayList();
```

Vardas – masyvo vardas

ArrayList metodai ir savybės

Metodo pavadinimas	Paskirtis
`Add(obj);`	Prideda naują objektą `obj` masyvo pabaigoje
`Clear();`	Pašalina visus masyvo elementus (objektus)
`Insert(index, obj);`	Įterpia naują objektą `obj` nurodytoje vietoje `index`
`RemoveAt(index);`	Pašalina objektą, kurio indeksas yra `index`
`Reverse();`	Apsuka masyvą, t.y. surašo elementų reikšmes atvirkščia tvarka

Savybė	Prasmė
`Capacity`	Masyvo dydis, kurį galima ne tik naudoti, bet ir keisti (didinti, mažinti)
`Count`	Masyvo elementų skaičius

ArrayList pavyzdys 1 (1/6)

```
using System.Collections;

...
ArrayList sveikAL = new ArrayList();
ArrayList realAL = new ArrayList();
ArrayList stringAL = new ArrayList();
ArrayList StudentaiAL = new ArrayList();

...
```

ArrayList pavyzdys 1 (2/6)

Užduotis: faile esančius sveikus skaičius (pvz., pažymius) reikia surašyti į **`ArrayList`** masyvą. Po to masyvo elementų reikšmes reikia parodyti ekrane.

Programoje, atitinkamose vietose, užrašysime:

```
using System.Collections;

...

ArrayList sveikAL = new ArrayList();

...
```

ArrayList pavyzdys 1 (3/6)

```
static void SkaitytiAL(string fv, ArrayList SveikiAL)
{
  using (StreamReader srautas = new StreamReader(fv))
  {
    string eilute;  // visa duomenų failo eilutė
    while ((eilute = srautas.ReadLine()) != null)
    {
      string[] eilDalis = eilute.Split(' '); // atskiros eilutės dalys
      for (int j = 0; j < eilDalis.Length; j++)
      {
        int skaicius = int.Parse(eilDalis[j]);
        SveikiAL.Add(skaicius);
      }
    }
  }
}
```

Kreipinys į **`ArrayList`** metodą **`Add`**.

ArrayList pavyzdys 1 (4/6)

```
static void SpausdintiAL(ArrayList sveikiAL,
                                 string koment)
{
    const string virsus =
          "--------------\n"
        + " Ind.  Reikšmė\n"
        + "--------------";
    Console.WriteLine("\n  " + koment);
    Console.WriteLine(virsus);
    int i = 0;
    foreach (int skaicius in sveikiAL)
       Console.WriteLine("{0, 3:d}   {1, 5:d}",
                         i++, skaicius);
    Console.WriteLine("--------------\n");
}
```

ArrayList pavyzdys 1 (5/6)

```
using System.Collections;
...
const string CFd = "..\\..\\Duomenys.txt";
...
ArrayList sveikAL = new ArrayList();
SkaitytiAL(CFd, sveikAL);
SpausdintiAL(sveikAL, "Pažymiai");
Console.WriteLine("Masyvo dydis: {0}",
                                  sveikAL.Capacity);
Console.WriteLine(„Elementų skaičius: {0}",
                                  sveikAL.Count);
```

ArrayList pavyzdys 1 (6/6)

Duomenys faile:

```
8  7  10
5  8
4  5
```

Rezultatai ekrane:

```
 Pažymiai
--------------
 Ind.  Reikšmė
--------------
   0        8
   1        7
   2       10
   3        5
   4        8
   5        4
   6        5
-------------
Masyvo dydis: 8
Elementų skaičius: 7
```

ArrayList pavyzdys 2 (1/8)

Pakoreguokime ankstesnę užduotį:

Faile "Studentai.txt" duota informacija apie vienos grupės studentų pažymius (pvz., kontrolinio darbo įvertinimai): studento pavardė ir vardas, pažymys.

Faile "Pazymiai.txt" duota informacija apie žinių vertinimo sistemą: pažymys, pažymio reikšmė.

Reikia paskaičiuoti, kiek kokių pažymių gavo studentai ?

Reikia failo duomenis surašyti į `ArrayList` masyvą, išspausdinti rezultatų faile bei rasti geriausią pažymį gavusį studentą.

Studentai.txt

```
Jonaitis Jonas; 8;
Petraitis Petras; 7;
...
Ramunaitė Ramunė; 5;
```

Pazymiai.txt

```
10;Puikiai;
9;Labai gerai;
...
2;Nepatenkinamai;
1;Nepatenkinamai;
```

ArrayList pavyzdys 2 (2/8)

```
class Studentas
{
    private string pavVrd;  // studento pavardė ir vardas
    private int pazym;      // pažymys (įvertinimas)
    public Studentas(string pavv, int pazym)
    {
        pavVrd = pavv;
        this.pazym = pazym;
    }
    public string ImtiPavv() { return pavVrd; }
    public int ImtiPazym()   { return pazym;  }
}
```

ArrayList pavyzdys 2 (3/8)

```
using System.Collections;

...
ArrayList studentaiAL = new ArrayList();

...
```

ArrayList pavyzdys 2 (4/8)

```
static void SkaitytiStudAL(string fv, ArrayList StudentaiAL)
{
  using (StreamReader srautas = new StreamReader(fv))
  {
    string eilute;  // visa duomenų failo eilutė
    while ((eilute = srautas.ReadLine()) != null)
    {
      string[] eilDalis = eilute.Split(';');  // eilutė dalys
      string pavVrd = eilDalis[0];
      int pazym = int.Parse(eilDalis[1]);
      Studentas studentas = new Studentas(pavVrd, pazym);
      StudentaiAL.Add(studentas);
    }
  }
}
```

Kreipinys į **ArrayList** metodą **Add**.

ArrayList pavyzdys 2 (5/8)

```
static void SpausdintiStudAL(string fv, ArrayList StudentaiAL, string a)
{
    const string virsus =
              "-----------------------------------\n"
            + " Nr.  Pavardė ir vardas     Pažymys \n"
            + "-----------------------------------";
    using (var fr = File.AppendText(fv))
    {
        fr.WriteLine("\n      " + a);
        fr.WriteLine(virsus);
        int i = 0;
        foreach (Studentas st in StudentaiAL)
                fr.WriteLine("{0, 3:d}   {1, -20}   {2, 2:d}",
                             ++i, st.ImtiPavv(), st.ImtiPazym());
        fr.WriteLine("-----------------------------------\n");
    }
}
```

ArrayList pavyzdys 2 (6/8)

```
static Studentas MaxStudentasAL(ArrayList studentaiAL)
{
    Studentas max = (Studentas) studentaiAL[0];
    for (int i = 1; i < studentaiAL.Count; i++)
    {
        Studentas stud = (Studentas)studentaiAL[i];
        if (stud.ImtiPazym() > max.ImtiPazym())
        {
            max = stud;
        }
    }
    return max;
}
```

ArrayList pavyzdys 2 (7/8)

```
using System.Collections;

...
ArrayList StudentaiAL = new ArrayList();
...
SkaitytiStudAL(CFd1, StudentaiAL);
SpausdintiStudAL(CFr, StudentaiAL, "Studentų sąrašas");
Console.WriteLine("Masyvo dydis: {0}",
                                  StudentaiAL.Capacity);
Console.WriteLine("Studentų skaičius: {0}",
                                  StudentaiAL.Count);
Studentas studMax = MaxStudentasAL(StudentaiAL);
Console.WriteLine("Geriausias studentas: {0} {1}",
        studMax.ImtiPavv(), studMax.ImtiPazym());
```

ArrayList pavyzdys 2 (8/8)

Duomenys faile:

```
Jonaitis Jonas;          8;
Petraitis Petras;        7;
Antanaitis Antanas;      10;
Giedraitis Giedrius;     5;
Onaitytė Ona;            8;
Juozaitis Juozas;        4;
Ramunaitė Ramunė;        5;
```

Rezultatai faile:

```
   Studentų sąrašas
-----------------------------------
 Nr.  Pavardė ir vardas     Pažymys
-----------------------------------
  1   Jonaitis Jonas            8
  2   Petraitis Petras          7
  3   Antanaitis Antanas       10
  4   Giedraitis Giedrius       5
  5   Onaitytė Ona              8
  6   Juozaitis Juozas          4
  7   Ramunaitė Ramunė          5
-----------------------------------
```

Rezultatai ekrane:

```
Masyvo dydis: 8
Studentų skaičius: 7
Geriausias studentas: Antanaitis Antanas 10
```

ktu informatikos fakultetas 1922

Demo

Demonstracinė programa MVS: ArrayList sudaryto iš objektų formavimas, duomenų skaitymas ir spausdinimas, skaičiavimai

ktu informatikos fakultetas 1922

Šioje temoje susipažinome su:

1. Problemos atvaizdavimu objektais
2. Maksimumo (minimumo) – reikšmės, indekso, objekto paieška
3. Maksimumo su apribojimais paieška
4. Dviejų maksimumų paieška
5. `ArrayList` klase

Klausimai?