ktu informatikos fakultetas 1922

T07. Susietasis sąrašas

2 ak. val.

ktu informatikos fakultetas 1922

Temos klausimai

1. Rekursinės duomenų struktūros.
2. Susietųjų sąrašų tipai.
3. Susietojo sąrašo samprata:
   - Elemento sukūrimas;
   - Sąrašo formavimas;
   - Sąrašo peržiūra.
4. Sąrašo klasė.

Veiksmai su vienkrypčiu sąrašu

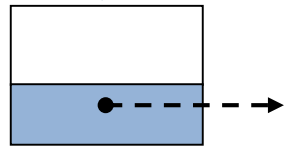
- formavimas,
- peržiūra,
- elementų šalinimas,
- elementų įterpimas,
- paieška,
- tvarkymas,
- rikiavimas.

Rekursinė duomenų struktūra 1/4

ktu informatikos fakultetas 1922

```
public sealed class Mazgas
{
    public int Duomenys { get; private set; }
    public Mazgas Kitas { get; set; }
    public Mazgas()
    {
    }
    public Mazgas(int duomenys, Mazgas adresas)
    {
        this.Duomenys = duomenys;
        this.Kitas = adresas;
    }
}
```

Mazgo (elemento) grafinis vaizdavimas

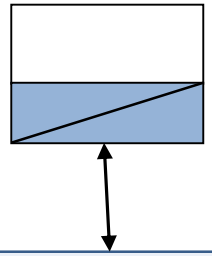


Nuoroda į kitą mazgą (elementą)

Rekursinė duomenų struktūra 2/4

```
. . .
Mazgas p1,  // pirma nuoroda
       p2;  // antra nuoroda
p1 = new Mazgas(9, null);
p2 = new Mazgas(7, null);
p1.Kitas = p2;

Console.Write("{0, 2:d}", p1.Duomenys);
Console.Write("{0, 2:d}", p2.Duomenys);
Console.WriteLine("{0, 2:d}", p1.Kitas.Duomenys);
. . .
```

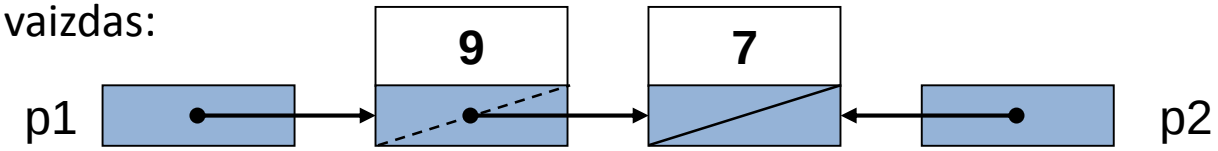


null-
nuorodos reikšmė, kuri į niekur nerodo

Ekrane matysite:

```
 9 7 7
```

Grafinis vaizdas:



Rekursinė duomenų struktūra 3/4

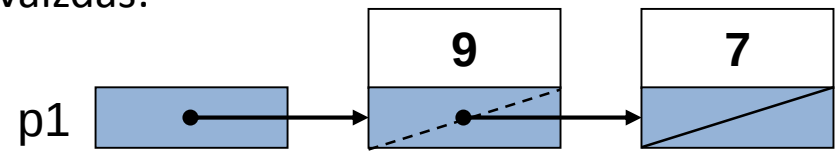
```
. . .
Mazgas p1;      // pirma nuoroda
p1 = new Mazgas(9, null);
p1.Kitas = new Mazgas(7, null);

Console.Write("{0, 2:d}", p1.Duomenys);
Console.WriteLine("{0, 2:d}", p1.Kitas.Duomenys);
. . .
```

Ekrane matysite:

9 7

Grafinis vaizdas:

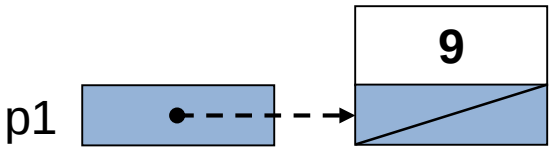


Rekursinė duomenų struktūra 4/4

```
. . .
Mazgas p1;  // pirma nuoroda
p1 = new Mazgas(9, null);
Console.WriteLine("{0, 2:d}", p1.Duomenys);
p1 = null;
Console.WriteLine("{0, 2:d}", p1.Duomenys);
. . .
```

Kuris veiksmas nelogiškas? Kokia bus klaida: **kompiliavimo** ar **vykdymo**?

Grafinis vaizdas:



ktu informatikos fakultetas 1922

Masyvų trūkumai

- Fiksuoto dydžio masyvas paprastai atmintyje užima daugiau vietos, negu reikia jame esančiai informacijai saugoti.
- Šalinant (įterpiant) informaciją masyve, reikia perstumti masyvo elementų reikšmes (jei negalima keisti jame esančios informacijos tvarkos).

Susietosios struktūros

- Tai struktūros, kurių elementuose saugoma ne tik informacija (duomenys), bet ir kito (-ų) elemento (-ų) adresas (ai).
- Šioms struktūroms realizuoti taikomos rekursinės aprašymo struktūros.

ktu informatikos fakultetas 1922

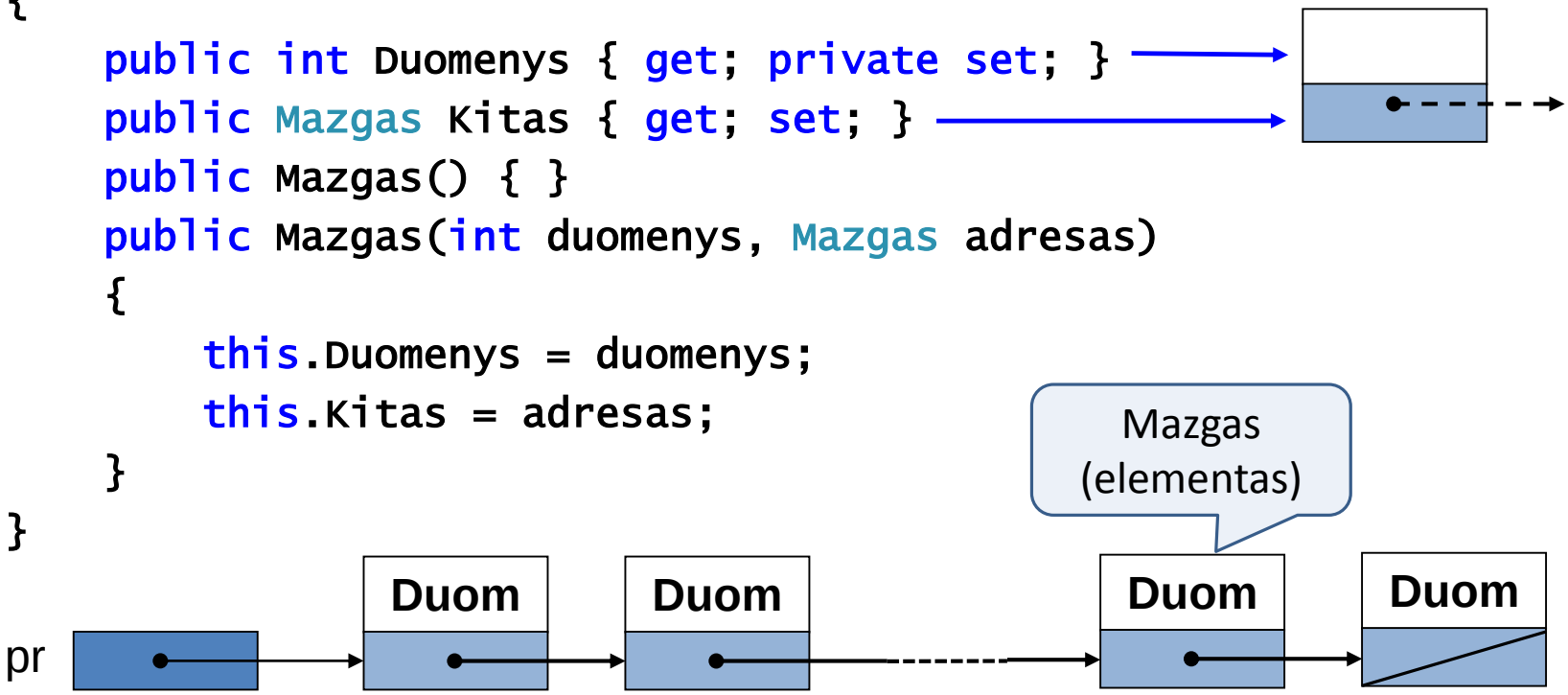
Susietųjų struktūrų tipai

- Tiesiniai vienpusiai (**vienkrypčiai**) sąrašai ir jų modifikacijos.
- Tiesiniai dvipusiai (**dvikrypčiai**) sąrašai ir jų modifikacijos.
- Netiesiniai sąrašai (pvz.: medžiai).

Tiesinis vienkryptis sąrašas

Tai sąrašas, kurio kiekviename elemente (mazge) saugoma informacija (duomenys) ir vieno tolesnio elemento adresas.

```
public sealed class Mazgas
{
    public int Duomenys { get; private set; }
    public Mazgas Kitas { get; set; }
    public Mazgas() { }
    public Mazgas(int duomenys, Mazgas adresas)
    {
        this.Duomenys = duomenys;
        this.Kitas = adresas;
    }
}
```



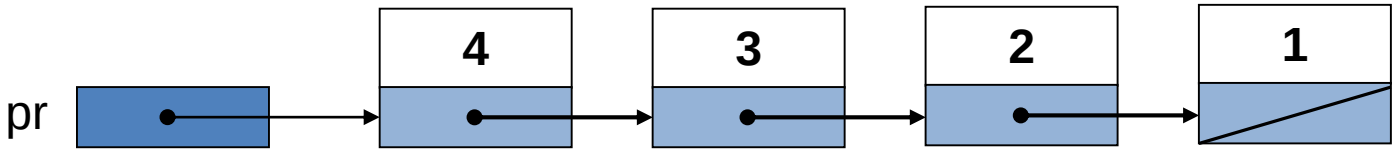
Masyvas ir tiesinis sąrašas 1/2

```
int[] Mas = new int[10];
int n = 4;
// ... reikšmių surašymas į masyvą
```

Masyvas sudarytas iš 10 elementų

1	2	3	4						
0	1	2	3	4	5	6	7	8	9

```
Mazgas pr;  // sąrašo pradžios nuoroda
// ... reikšmių surašymas į sąrašą
```



Masyvas ir tiesinis sąrašas 2/2

```
int[] Mas = new int[10];
int n = 0; // masyvas tuščias
```

Masyvas sudarytas iš 10 elementų

?	?	?	?	?	?	?	?	?	?
0	1	2	3	4	5	6	7	8	9

```
Mazgas pr = null;  // sąrašas tuščias
```

pr 

Sąrašo pradžios nuoroda

```
Mazgas pr;  // sąrašo pradžios nuoroda
```

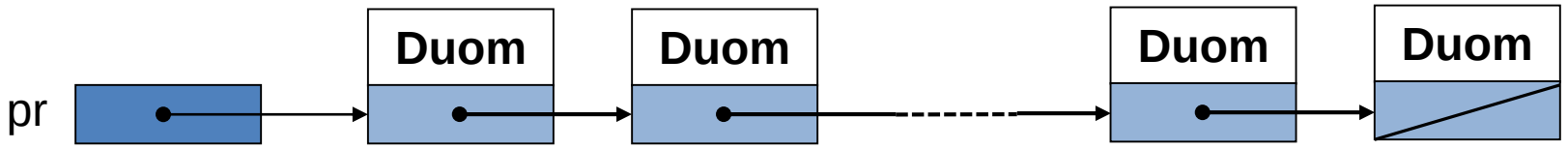
- Nuoroda į susietojo sąrašo pirmąjį elementą vadinama **sąrašo pradžios nuoroda**.
- Sąrašą visuomet žymi sąrašo pradžios nuoroda.
- Nuoroda skelbiama privačiojoje (`private`) klasės aprašo dalyje. (sudarant sąrašo klasės aprašą)
- Klasės (sąrašo) konstruktoriuje sąrašo pradžios nuorodai turi būti priskirta pradinė reikšmė:

```
pr = null;
```

Nulinė (`null`) nuoroda

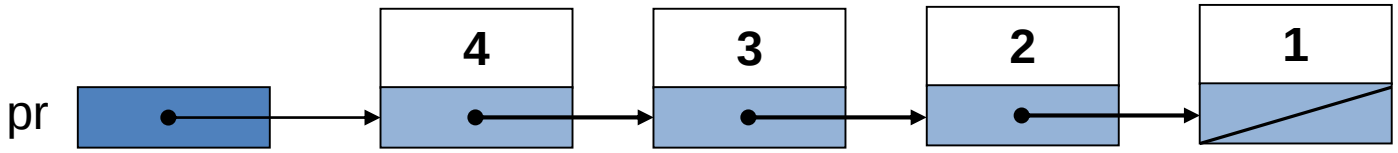
Nulinė nuoroda susietajame sąraše naudojama *dviem atvejais*:

1) Priskiriama sąrašo pradžios nuorodai. Tuomet ji parodo, kad sąrašas yra tuščias. **Visi metodai privalo mokėti dirbti su susietuoju sąrašu, kuris yra tuščias**.
2) Priskiriama paskutiniame sąrašo elemente vietoj ryšio į kitą elementą. Tuomet ji pažymi, kad tai paskutinis sąrašo elementas.

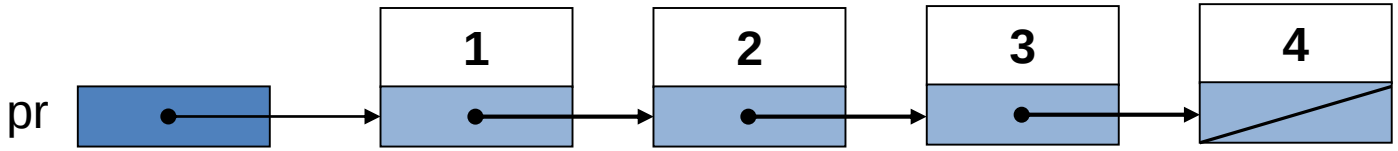


Vienkrypčio sąrašo tipai

Tiesinis vienkryptis **atvirkštinis** sąrašas:



Tiesinis vienkryptis **tiesioginis** sąrašas:

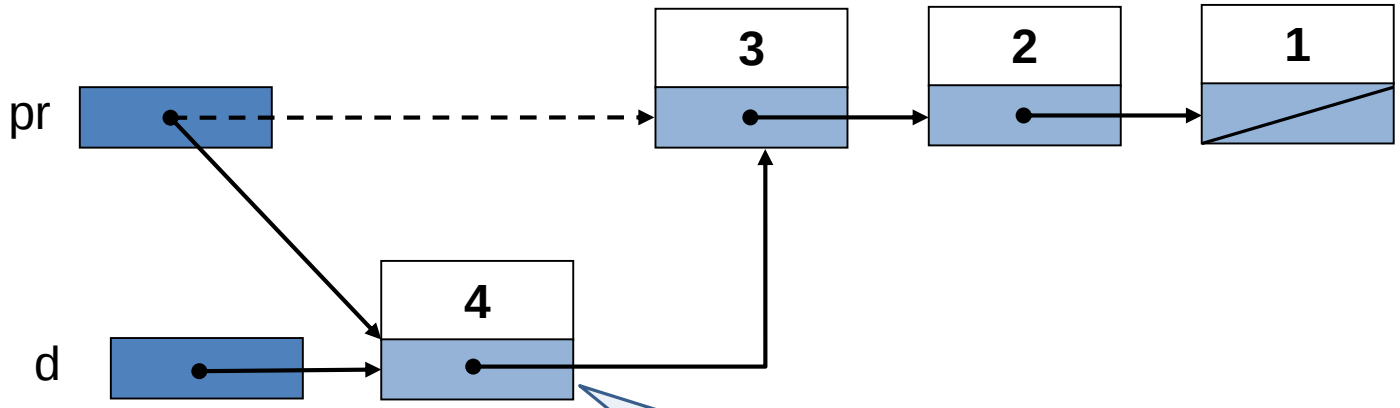


Failas, iš kurio duomenų sudaryti sąrašai:

1 2 3 4

Tiesinis vienkryptis **atvirkštinis** sąrašas

Tiesinio vienkrypčio **atvirkštinio** sąrašo formavimas:



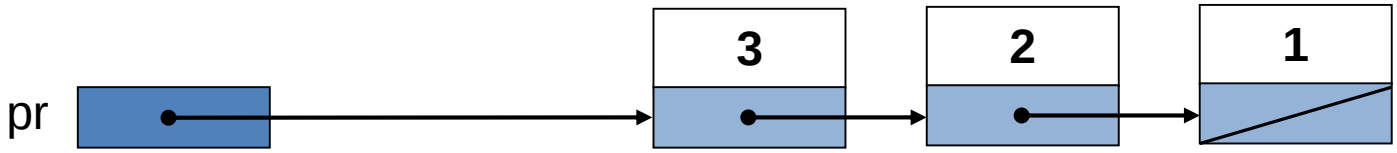
Naujai sukurtas elementas (objektas) prijungiamas sąrašo pradžioje

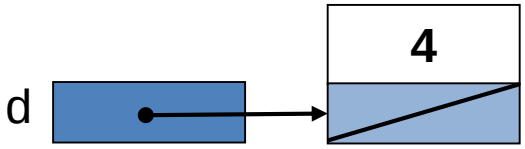
Duomenų failas:

1	2	3	**4**

Tiesinis vienkryptis **atvirkštinis** sąrašas

Tiesinio vienkrypčio **atvirkštinio** sąrašo formavimas:





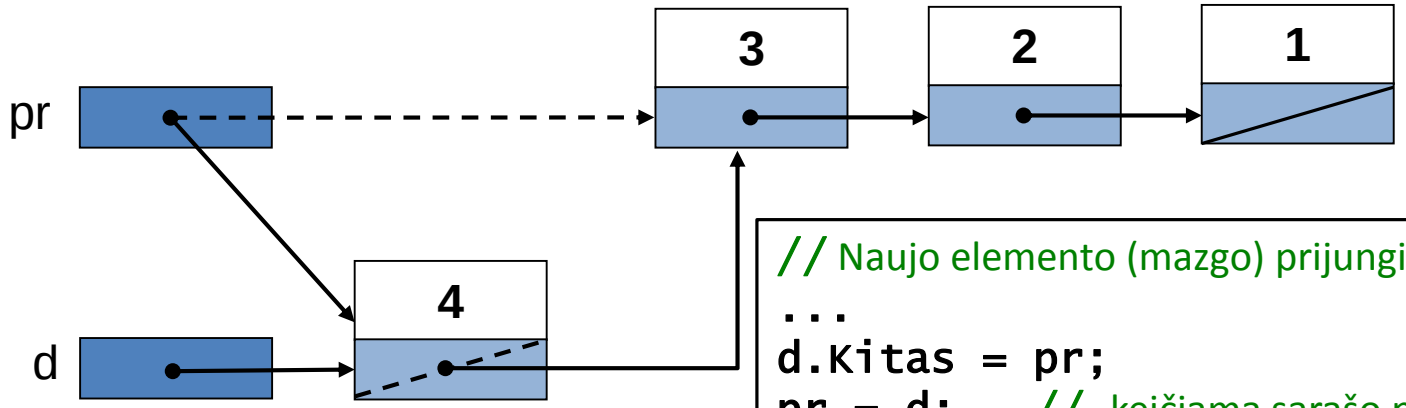

```
// Naujo elemento (mazgo) sukūrimas
int skaicius; // įvedamas skaičius
...
Mazgas d = new Mazgas(skaicius, null);
...
```

Duomenų failas:

```
1  2  3  4
```

Tiesinis vienkryptis **atvirkštinis** sąrašas

Tiesinio vienkrypčio **atvirkštinio** sąrašo formavimas:



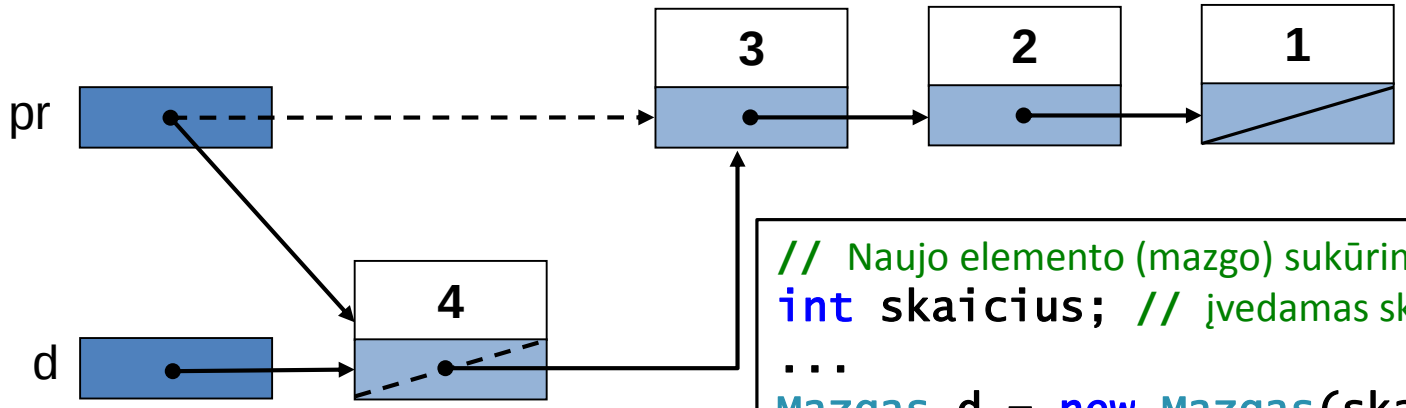

```
// Naujo elemento (mazgo) prijungimas sąrašo pradžioje
...
d.Kitas = pr;
pr = d;    // keičiama sąrašo pradžios nuoroda
```

Duomenų failas:

```
1  2  3  4
```

Tiesinis vienkryptis **atvirkštinis** sąrašas

Tiesinio vienkrypčio **atvirkštinio** sąrašo formavimas:




```
// Naujo elemento (mazgo) sukūrimas
int skaicius; // įvedamas skaičius
...
Mazgas d = new Mazgas(skaicius, null);
// Naujo elemento (mazgo) prijungimas sąrašo pradžioje
d.Kitas = pr;
pr = d;    // keičiama sąrašo pradžios nuoroda
```

Duomenų failas:

```
1  2  3  4
```

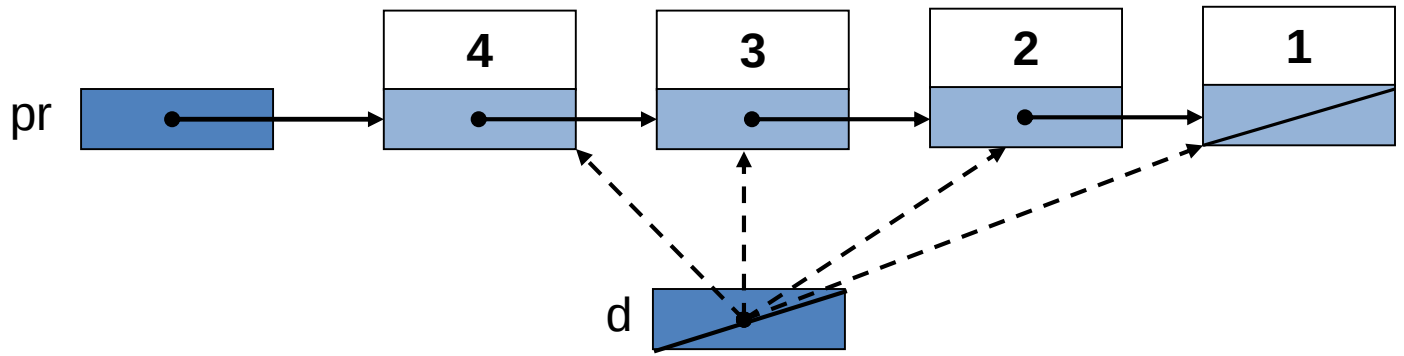
Ar šis būdas tinka tuščiam sąrašui?

Atvirkštinio sąrašo formavimas

```
static void SkaitytiAtv(string fv, out Mazgas pr)
{
    using (var failas = new StreamReader(fv))  {
        int skaicius;
        string eilute;
        pr = null;
        while ((eilute = failas.ReadLine()) != null)
        {
            skaicius = Convert.ToInt32(eilute);
            Mazgas d = new Mazgas(skaicius, null);
            d.Kitas = pr;
            pr = d;
            // arba
            //pr = new Mazgas(skaicius, pr);
        }
    }
}
```

Sąrašo peržiūros ciklas 1/6

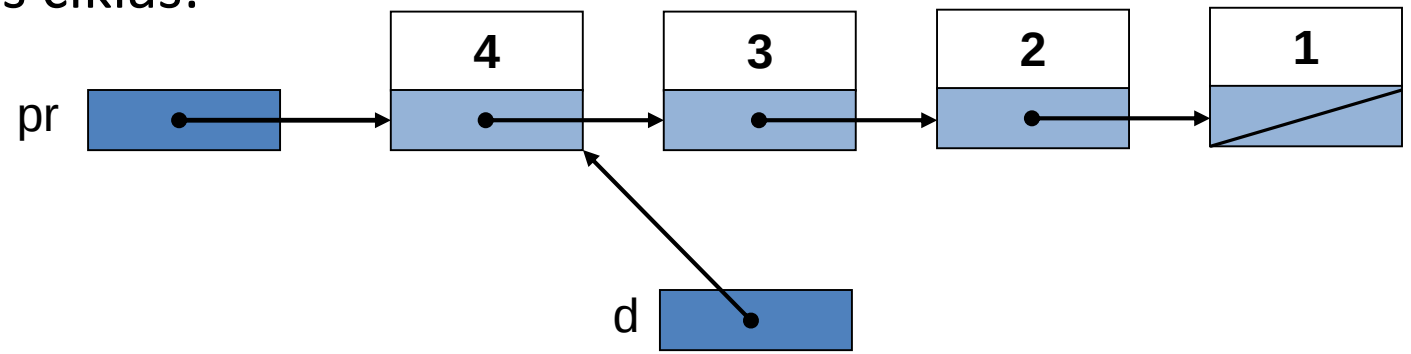
```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```



Sąrašo peržiūros ciklas 2/6

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```

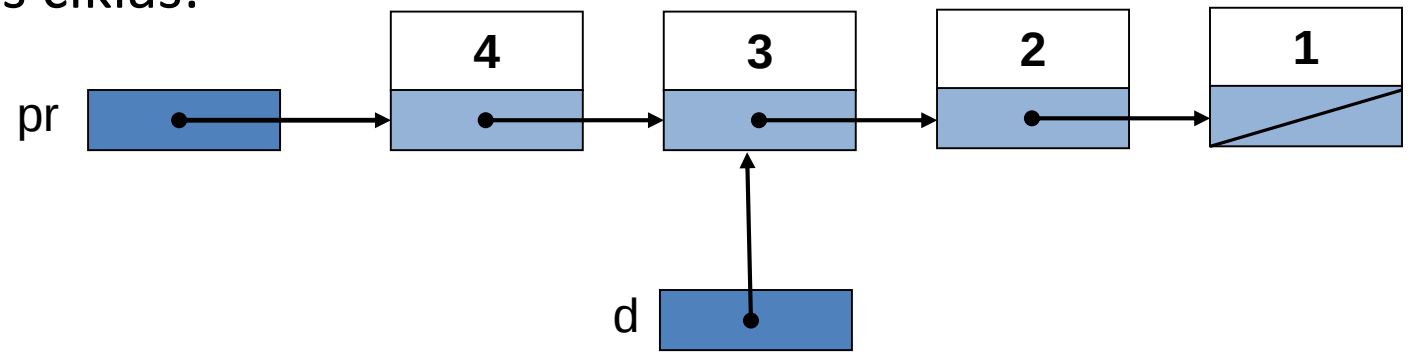
1-as ciklas:



Sąrašo peržiūros ciklas 3/6

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```

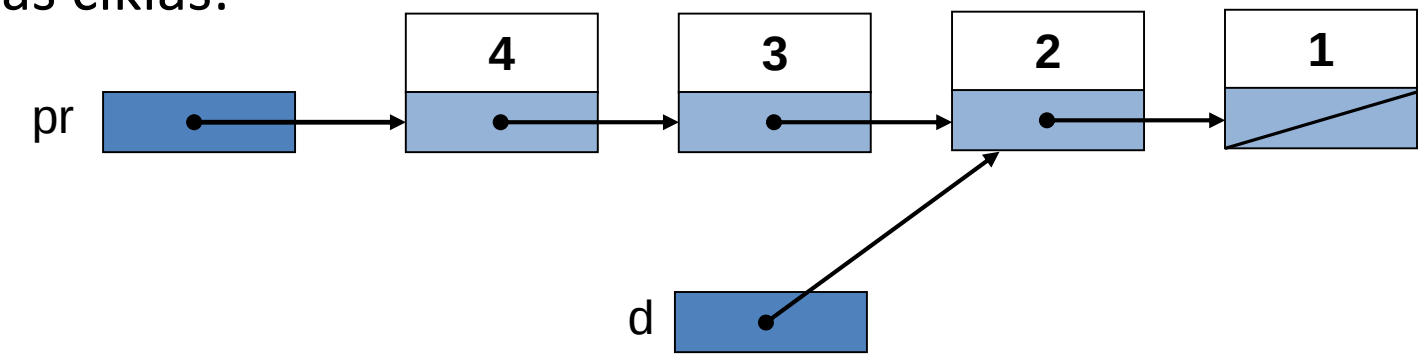
2-as ciklas:



Sąrašo peržiūros ciklas 4/6

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```

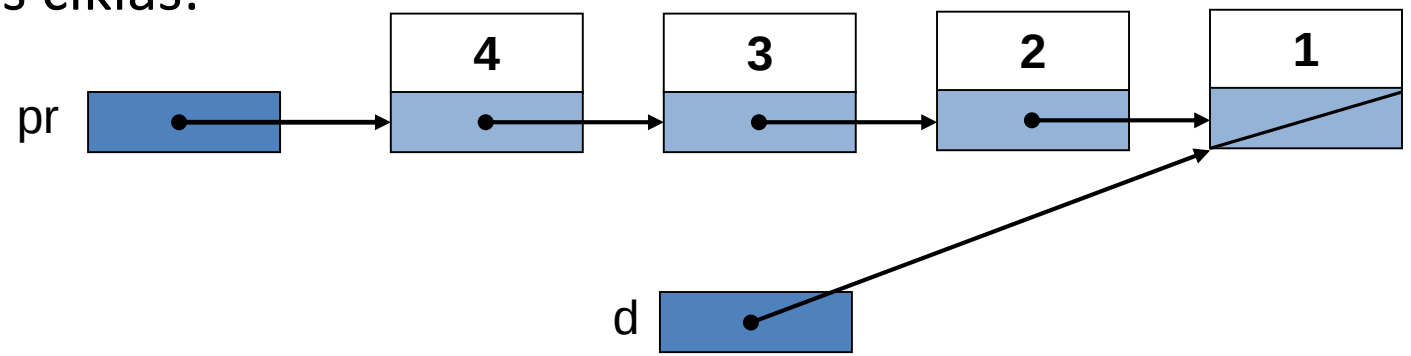
3-ias ciklas:



Sąrašo peržiūros ciklas 5/6

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```

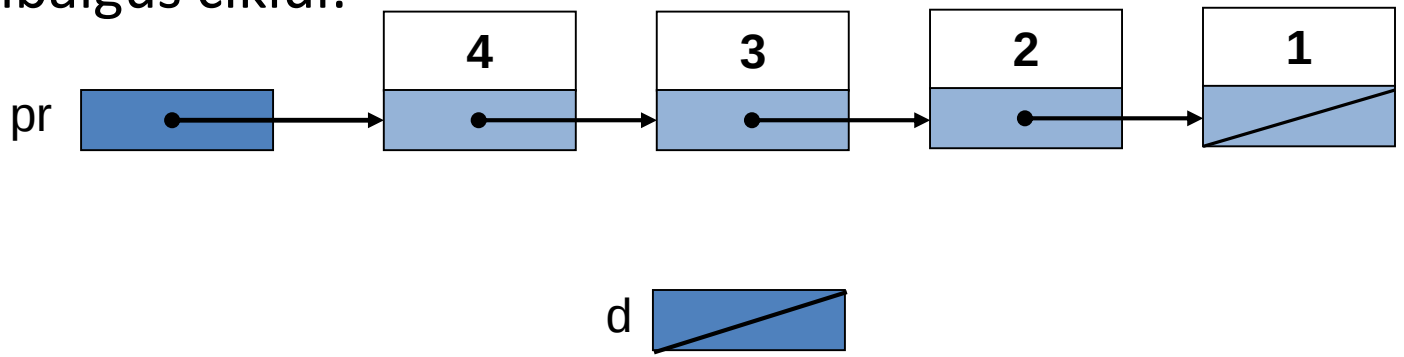
4-as ciklas:



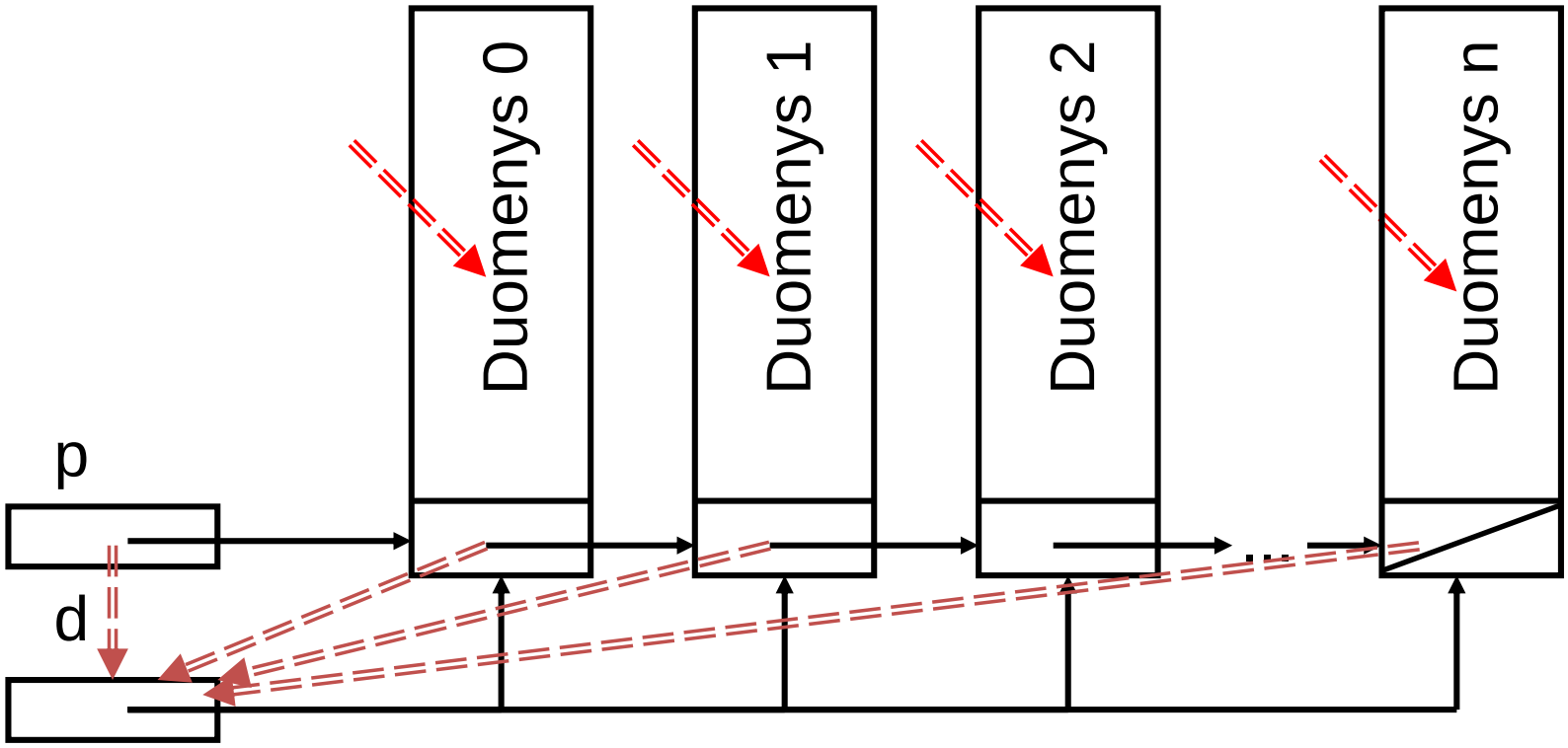
Sąrašo peržiūros ciklas 6/6

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
    ... //   Veiksmai su d.Duomenys (spausdinimas, skaičiavimai)
}
```

Pasibaigus ciklui:



Sąrašo peržiūros ciklas



Kai bus išnagrinėtas paskutinis elementas, ciklą reikia užbaigti.

Atvirkštinio sąrašo spausdinimas

```
static void Spausdinti(string fv, Mazgas pr, string koment)
{
    using (var failas = new StreamWriter(fv, true))
    {
        failas.WriteLine(koment);
        for (Mazgas d = pr; d != null; d = d.Kitas)
        {
            failas.WriteLine("{0, 3:d}", d.Duomenys);
        }
        failas.WriteLine();
    }
}
```

Sąrašo formavimas ir peržiūra

```
. . .
const string CFd = @"..\..\Duomenys.txt";
const string CFr = @"..\..\Rezultatai.txt";
. . .
Mazgas prA;  // atvirkštinio sąrašo pradžios rodyklė
SkaitytiAtv(CFd, out prA);
Spausdinti(CFr, prA, "Atvirkštinis sąrašas");
. . .
```

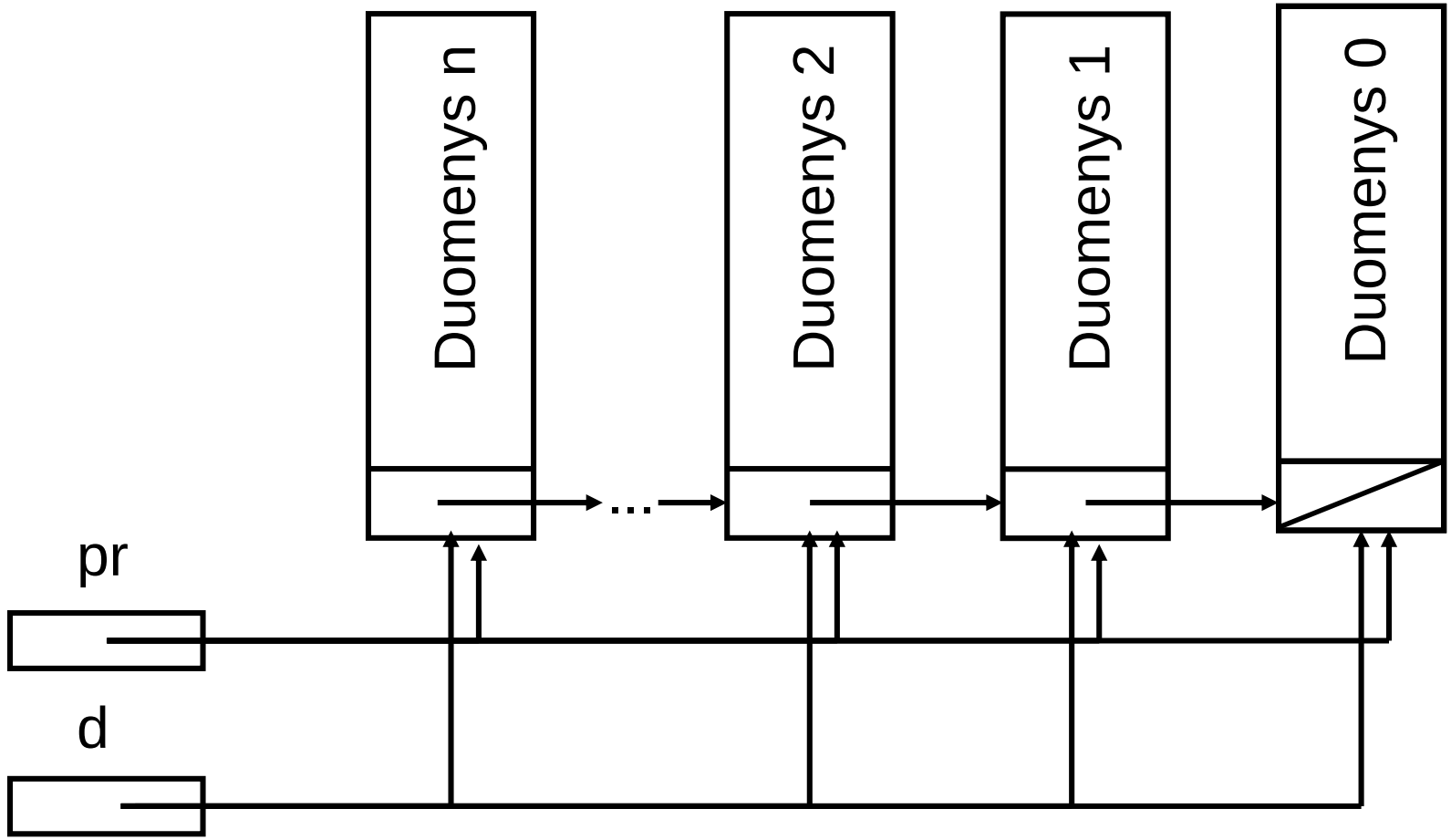
Duomenų failas:

```
1  2  3  4
```

Rezultatų failas:

```
Atvirkštinis sąrašas
  4
  3
  2
  1
```

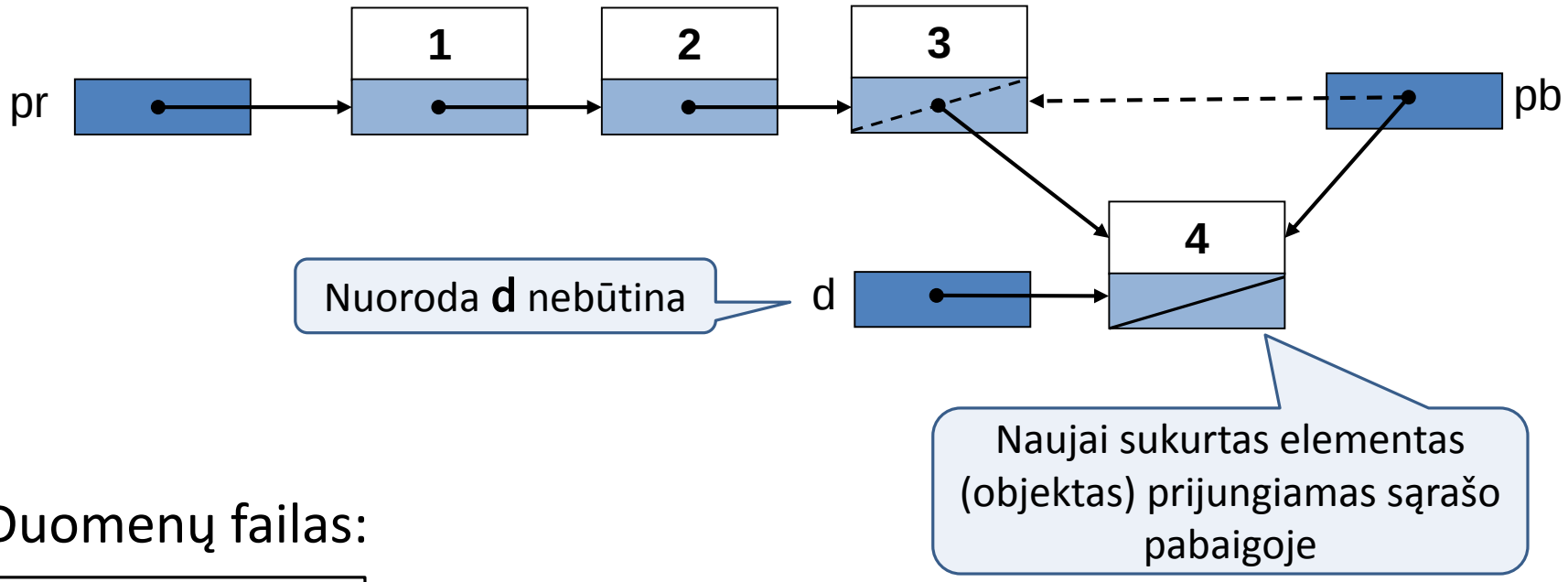
Atvirkštinio sąrašo formavimas (su animacija)



Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Tiesinio vienkrypčio **tiesioginio** sąrašo formavimas:



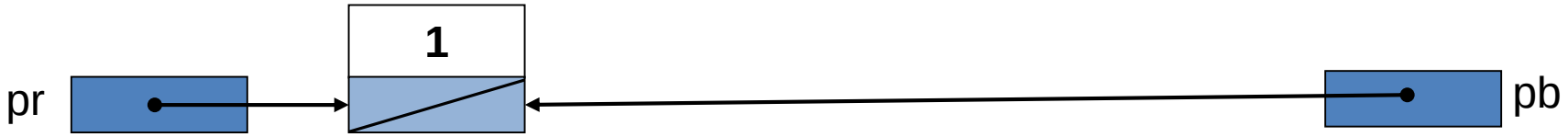
Duomenų failas:

1	2	3	**4**

Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Pirmojo sąrašo elemento (mazgo) sukūrimas (atskiras atvejis):




```
// Pirmo elemento (mazgo) sukūrimas
int skaicius; // įvedamas skaičius
...
pr = new Mazgas(skaicius, null);
pb = pr;
...
```

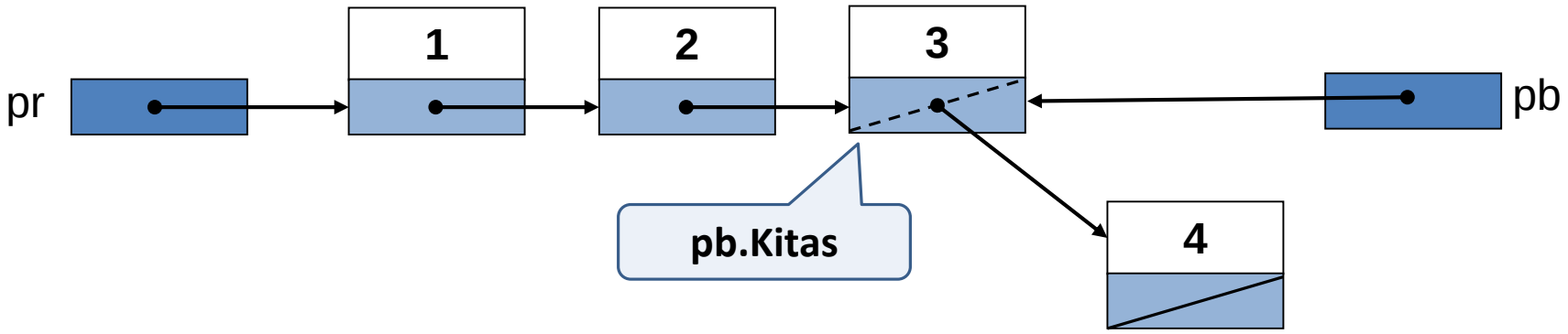
Duomenų failas:

```
1 2 3 4
```

Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Kitų sąrašo elementų (mazgų) sukūrimas:



Duomenų failas:

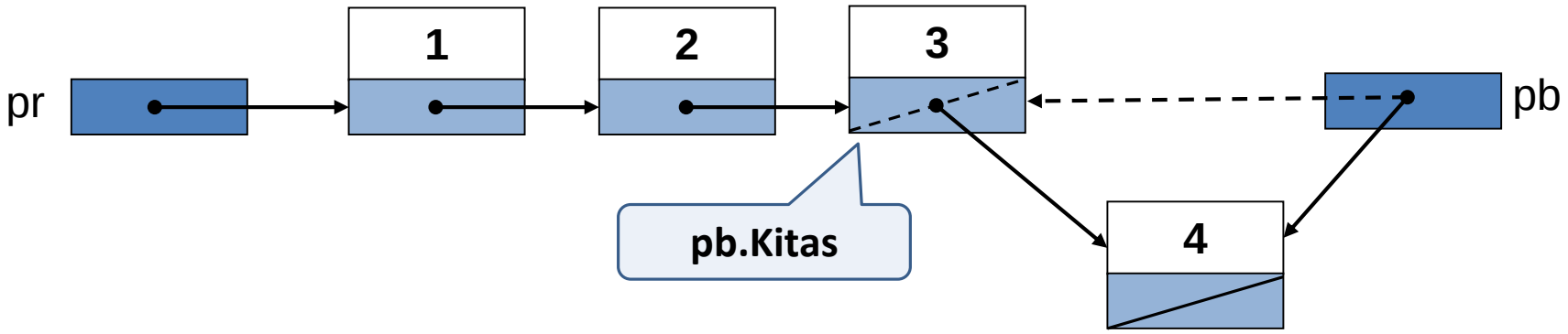
```
1  2  3  4
```

```
// Kitų elementų (mazgų) sukūrimas
int skaicius; // įvedamas skaičius
...
pb.Kitas = new Mazgas(skaicius, null);
...
```

Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Kitų sąrašo elementų (mazgų) prijungimas:




```
// Kitų elementų (mazgų) prijungimas
pb = pb.Kitas;
```

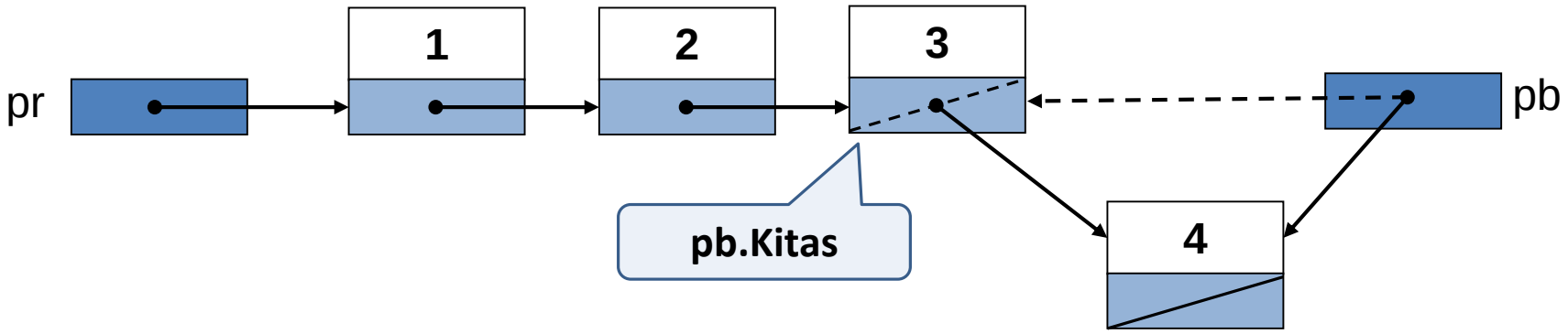
Duomenų failas:

```
1 2 3 4
```

Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Kitų sąrašo elementų (mazgų) sukūrimas ir prijungimas:



Duomenų failas:

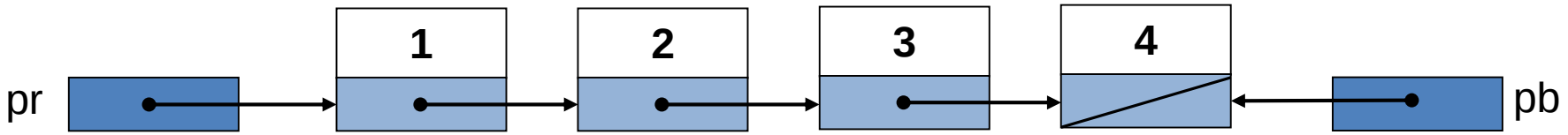
```
1  2  3  4
```

```
// Kitų elementų (mazgų) sukūrimas
int skaicius; // įvedamas skaičius
...
pb.Kitas = new Mazgas(skaicius, null);
pb = pb.Kitas;
```

Tiesinis vienkryptis **tiesioginis** sąrašas

ktu informatikos fakultetas 1922

Paskutiniojo sąrašo elemento paieška, kai žinoma sąrašo pradžia:




```
// Sąrašo paskutiniojo elemento paieškos ciklas
Mazgas pb = pr;
while (pb.Kitas != null)
    pb = pb.Kitas;
```

Ar šis ciklas visuomet veiks?

Failas, iš kurio duomenų sudarytas sąrašas:

```
1  2  3  4
```

Tiesioginio sąrašo formavimas

```
static void SkaitytiTiesiog(string fv, out Mazgas pr)
{
    using (var failas = new StreamReader(fv))
    {
        int skaicius;
        string eilute;
        pr = null;           // sąrašo pradžios nuoroda
        Mazgas pb = null;    // sąrašo pabaigos nuoroda
        // Pirmo sąrašo elemento sukūrimas
        if ((eilute = failas.ReadLine()) != null)
        {
            skaicius = Convert.ToInt32(eilute);
            pr = new Mazgas(skaicius, null);
            pb = pr;
        }
        // Kitų (likusių) sąrašo elementų sukūrimas
        ...
    }
}
```

Tiesioginio sąrašo formavimas

```
// Kitų (likusių) sąrašo elementų (mazgų) sukūrimas
...
while ((eilute = failas.ReadLine()) != null)
{
    skaicius = Convert.ToInt32(eilute);
    pb.Kitas = new Mazgas(skaicius, null);
    pb = pb.Kitas;
}
```

Sąrašo formavimas ir peržiūra

```
. . .
const string CFd = @"..\..\Duomenys.txt";
const string CFr = @"..\..\Rezultatai.txt";
. . .
Mazgas prT;  // tiesioginio sąrašo pradžios rodyklė
SkaitytiTiesiog(CFd, out prT);
Spausdinti(CFr, prT, "Tiesioginis sąrašas");
. . .
```

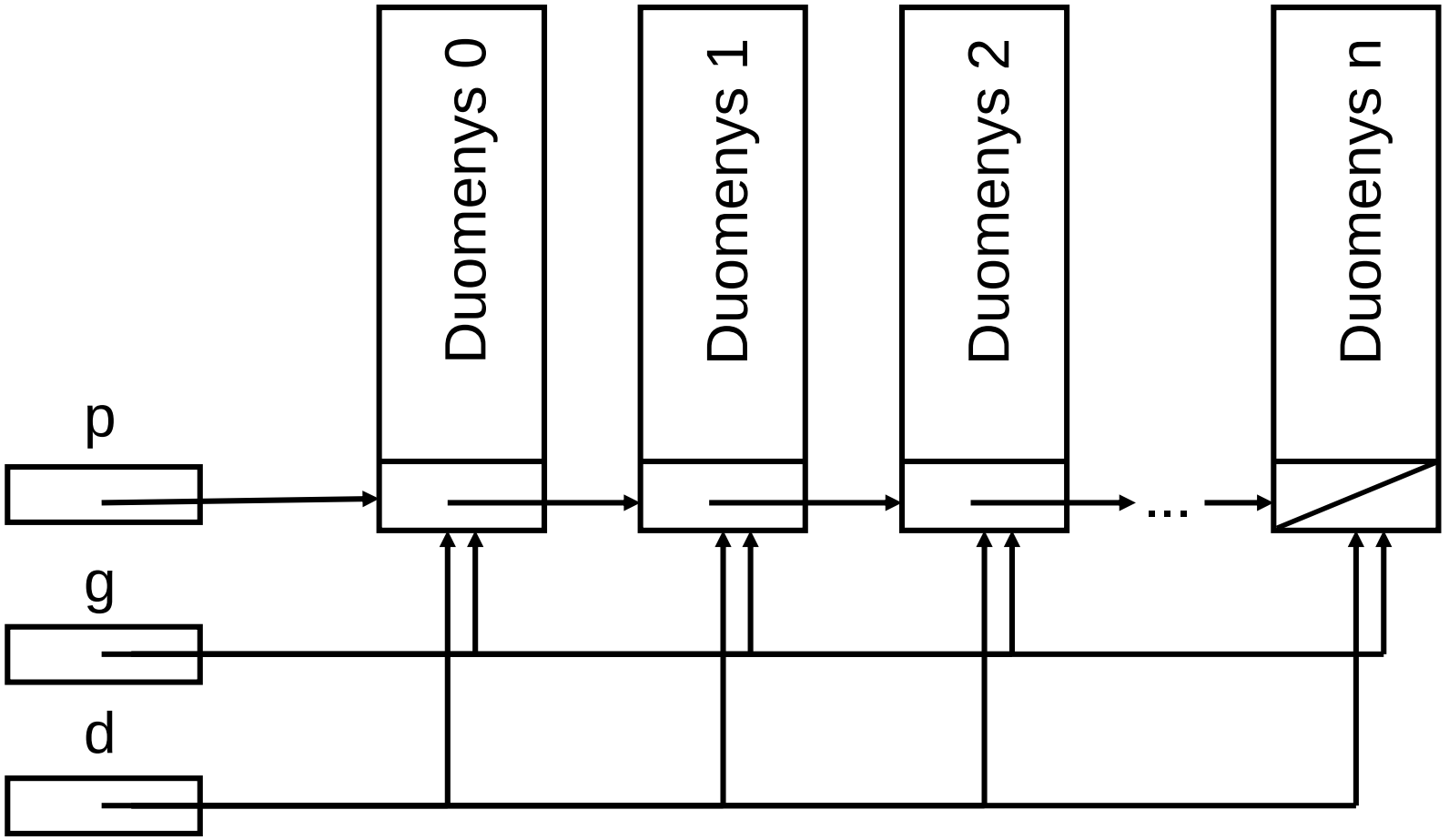
Duomenų failas:

```
1  2  3  4
```

Rezultatų failas:

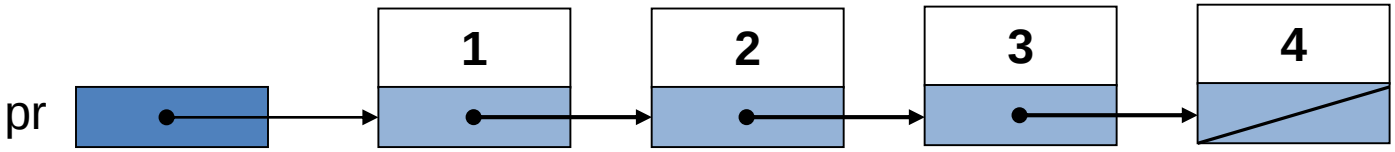
```
Tiesioginis sąrašas
  1
  2
  3
  4
```

Tiesioginio sąrašo formavimas (su animacija)



Vidurkio skaičiavimas sąraše

```
static double Vidurkis(Mazgas pr)
{
    int suma = 0;
    int kiekis = 0;
    for (Mazgas d = pr; d != null; d = d.Kitas)
    {
        suma += d.Duomenys;
        kiekis++;
    }
    if (kiekis != 0)
        return (double)suma / kiekis;
    else
        return 0.0;
}
...
Console.WriteLine("Reikšmių vidurkis {0, 7:f2}", Vidurkis(prT));
```



Sąrašo klasė

Sąrašo vieta klasėje

Sąrašo pradžios ir pabaigos nuorodos klasėje turi būti aprašytos `private`:

```
private Mazgas pr;   // sąrašo pradžia
private Mazgas pb;   // sąrašo pabaiga
```

Klasėje atviroje srityje `public` neturi būti metodo, kuris per savo vardą ar per parametrus grąžintų kurio tai sąrašo elemento adresą, nes tai pažeidžia OP principus, pvz.:

```
public Mazgas MaxElementas() // Draudžiama!
{
   ...
}
```

Sąrašo duomenų perdavimas į klasės išorę

Į klasės išorę gali būti perduoti tik sąrašo elementų mazgo dalyje esantys duomenys.

Duomenų perdavimui iš klasės ir į klasę naudojami **sąsajos metodai**.

Kaip ir konteinerinės klasės su objektų masyvu atveju, duomenims paimti, pakeisti ir naujiems patalpinti sukuriami atskiri sąsajos metodai.

Prisiminkime: masyvo elemento išrinkimas

Masyvo atveju viskas paprasta:

klasės išorėje žinome, kiek elementų yra (šį skaičių pateikia atitinkamas sąsajos metodas), ir pagal eilės numerį (indeksą) juos galime išrinkti.

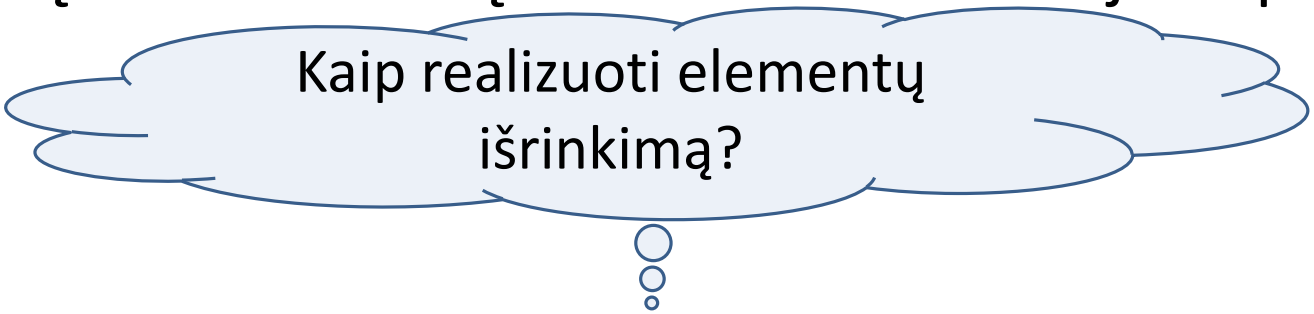
Sąsajos metodui perduodamas elemento eilės numeris (indeksas).

Prisiminkime: sąrašo elemento išrinkimas

Tiesinio sąrašo elementai išrenkami kintamojo-nuorodos pagalba „keliaujant“ per sąrašą nuo vieno elemento prie kito.

```
for (Mazgas d = pr; d != null; d = d.Kitas)
{
   ... //   Veiksmai su d.Duomenys
}
```

Tačiau sąrašo elementų adresai klasės išorėje neprieinami.



Sąrašo elemento (mazgo) išrinkimas

```
for (Mazgas d = pr; d != null; d = d.Kitas) {}
```

Kiekvieną iš trijų `for` ciklo dalių:

```
d = pr
d != null
d = d.Kitas
```

realizuojame atskiru sąsajos metodu klasėje.

`for` cikle vartojamą nuorodą **`d`** aprašome klasėje ir pavadiname **`ss`**:

```
private Mazgas ss;   // sąrašo sąsaja
```

Šiai nuorodai klasės konstruktoriuje suteikiame reikšmę `null`.
Sąsajos metoduose, duomenims paimti ar keisti, naudosime šią sąsajos nuorodą **`ss`**.

Sąrašo pildymas naujais elementais

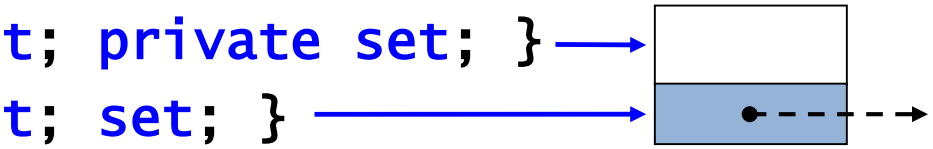
Prisiminkime: tiesinį vienkryptį sąrašą nauju elementu galime papildyti:

- elementą prijungdami sąrašo pabaigoje (**tiesioginis** sąrašo formavimas);
- elementą prijungdami sąrašo pradžioje (**atvirkštinis** sąrašo formavimas).

Taigi, reikalingi dar du sąsajos metodai sąrašo elementų papildymu pradžioje ir pabaigoje.

Sąrašo elemento (mazgo) klasė

```
public sealed class Mazgas
{
  public int Duomenys { get; private set; }
  public Mazgas Kitas { get; set; }
  public Mazgas() { }
  public Mazgas(int duomenys, Mazgas adresas)
  {
    this.Duomenys = duomenys;
    this.Kitas = adresas;
  }
}
```



ktu informatikos fakultetas 1922

Sąrašo klasė

```
public sealed class Sąrašas
{
    private Mazgas pr;   // sąrašo pradžia
    private Mazgas pb;   // sąrašo pabaiga
    private Mazgas ss;   // sąrašo sąsaja
    // Konstruktorius: suteikiamos pradinės reikšmės
    public Sąrašas()
    {
        this.pr = null;
        this.pb = null;
        this.ss = null;
    }
    // Sąsajos metodai
    ...
}
```

Sąrašo klasės sąsajos metodai 1/5

```
public sealed class Sąrašas
{
    ...
    // Sąsajos metodai
    // Sukuriamas sąrašo elementas ir prijungiamas prie sąrašo PRADŽIOS
    // skaicius – naujo elemento reikšmė (duomenys)
    public void DėtiDuomenisA(int skaicius)
    {
        var d = new Mazgas(skaicius, null);
        d.Kitas = pr;
        pr = d;
        // arba
        //pr = new Mazgas(skaicius, pr);
    }
    ...
}
```

Sąrašo klasės sąsajos metodai 2/5

```
public sealed class Sąrašas
{     ...
      // Sukuriamas sąrašo elementas ir prijungiamas prie sąrašo PABAIGOS
      // skaicius – naujo elemento reikšmė (duomenys)
      public void DėtiDuomenisT(int skaicius)
      {
          var d = new Mazgas(skaicius, null);
          if (pr != null)
          {
              pb.Kitas = d;
              pb = d;
          }
          else   // jei sąrašas tuščias
          {
              pr = d;
              pb = d;
          }
      }
     ...
}
```

Sąrašo klasės sąsajos metodai 3/5

ktu informatikos fakultetas 1922

```
public sealed class Sąrašas
{
    ...
    // Grąžina sąrašo sąsajos elemento reikšmę
    public int ImtiDuomenis()
    {
        return ss.Duomenys;
    }
    ...
}
```

Sąrašo klasės sąsajos metodai 4/5

```
public sealed class Sąrašas
{
    ...
    // Sąsajai priskiriama sąrašo pradžia
    public void Pradžia()
    {
        ss = pr;
    }
    // Sąsajai priskiriamas sąrašo sekantis elementas
    public void Kitas()
    {
         ss = ss.Kitas;
    }
    // Grąžina true, jeigu sąsaja netuščia; false - priešingu atveju
    public bool Yra()
    {
        return ss != null;
    }

    ...
}
```

Sąrašo klasės sąsajos metodai 5/5

```
public sealed class Sąrašas
{
    ...
     // Sunaikinamas sąrašas
     public void Naikinti()
     {
         while (pr != null)
         {
             ss = pr;
             pr = pr.Kitas;
             ss.Kitas = null;
         }
         pb = ss = pr;   // pb = ss = null;
     }
}
```

Sąrašo klasės sąrašo sudarymas 1/2

ktu informatikos fakultetas 1922

```
// Skaitomi skaičiai iš failo ir sudedami į sąrašą ATVIRKŠTINE tvarka
// fv – duomenų failo vardas
static Sąrašas SkaitytiAtv(string fv)
{
    var A = new Sąrašas();
    using (var failas = new StreamReader(fv))
    {
        int skaicius;
        string eilute;
        while ((eilute = failas.ReadLine()) != null)
        {
            skaicius = Convert.ToInt32(eilute);
            A.DėtiDuomenisA(skaicius);
        }
    }
    return A;
}
```

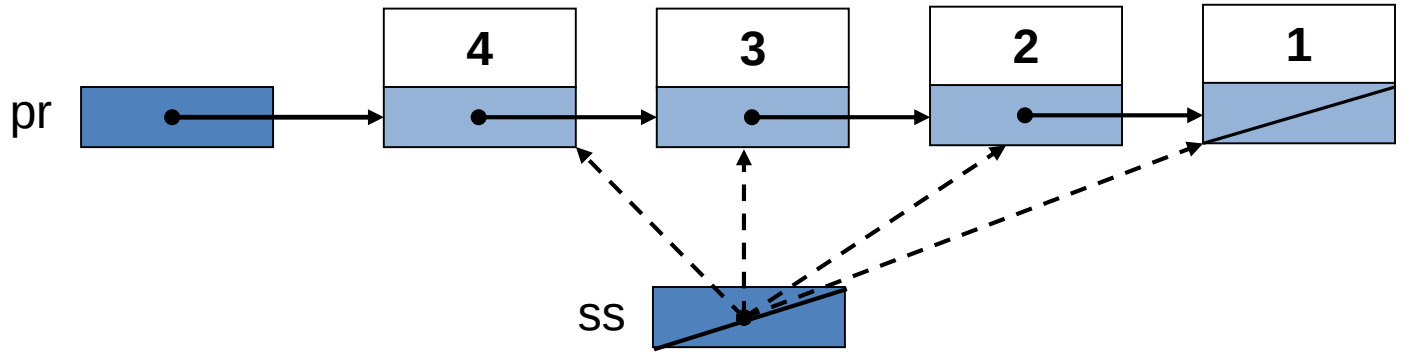
Sąrašo klasės sąrašo sudarymas 2/2

ktu informatikos fakultetas 1922

```
// Skaitomi skaičiai iš failo ir sudedami į sąrašą TIESIOGINE tvarka
// fv – duomenų failo vardas
static Sąrašas SkaitytiTiesiog(string fv)
{
    var A = new Sąrašas();
    using (var failas = new StreamReader(fv))
    {
        int skaicius;
        string eilute;
        while ((eilute = failas.ReadLine()) != null)
        {
            skaicius = Convert.ToInt32(eilute);
            A.DėtiDuomenisT(skaicius);
        }
    }
    return A;
}
```

Sąrašo klasės sąrašo peržiūros ciklas 1/6

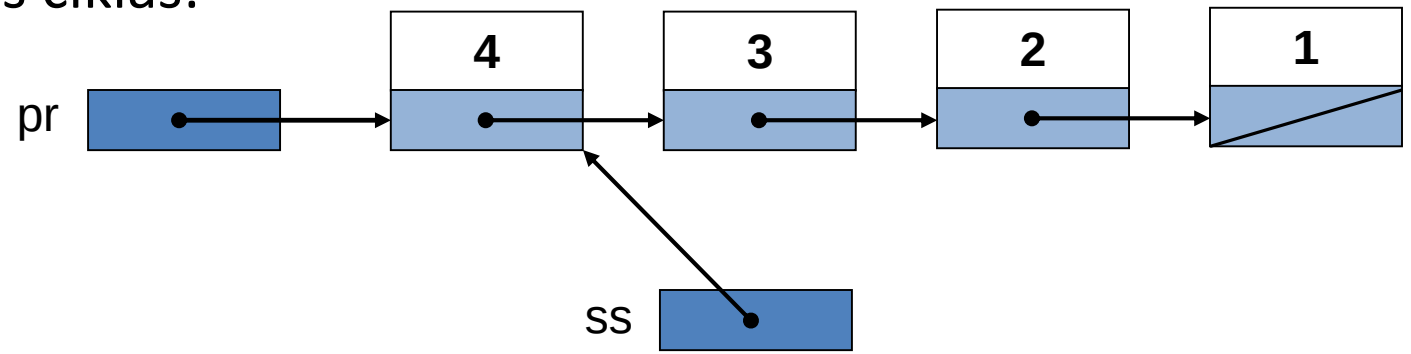
```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```



Sąrašo klasės sąrašo peržiūros ciklas 2/6

```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```

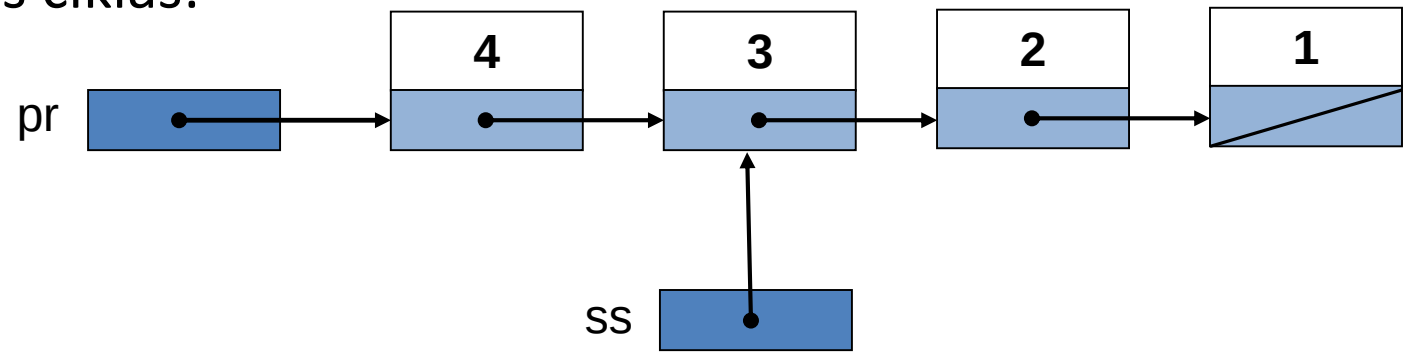
1-as ciklas:



Sąrašo klasės sąrašo peržiūros ciklas 3/6

```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```

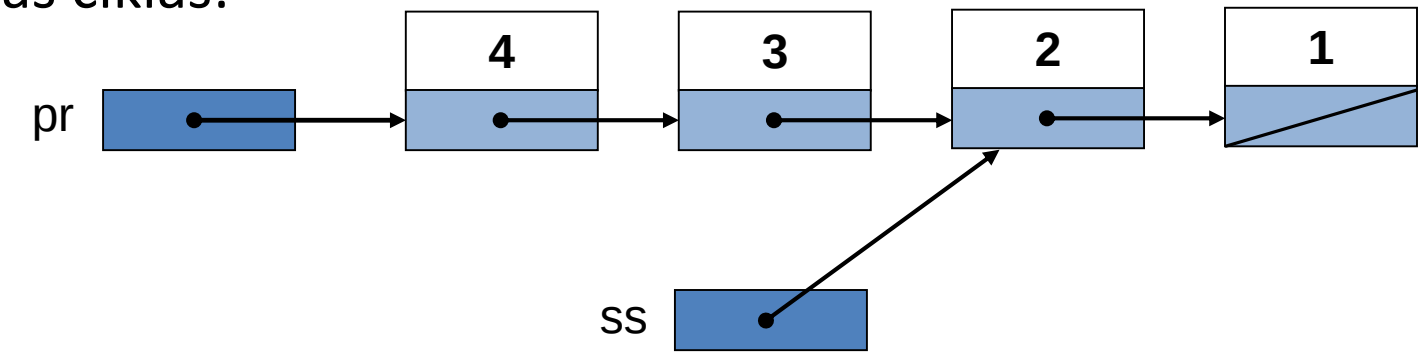
2-as ciklas:



Sąrašo klasės sąrašo peržiūros ciklas 4/6

```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```

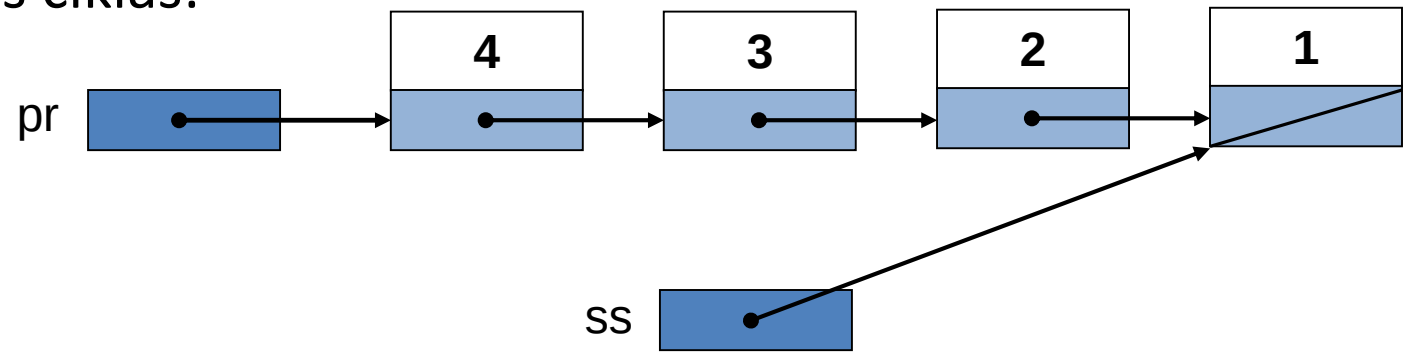
3-ias ciklas:



Sąrašo klasės sąrašo peržiūros ciklas 5/6

```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```

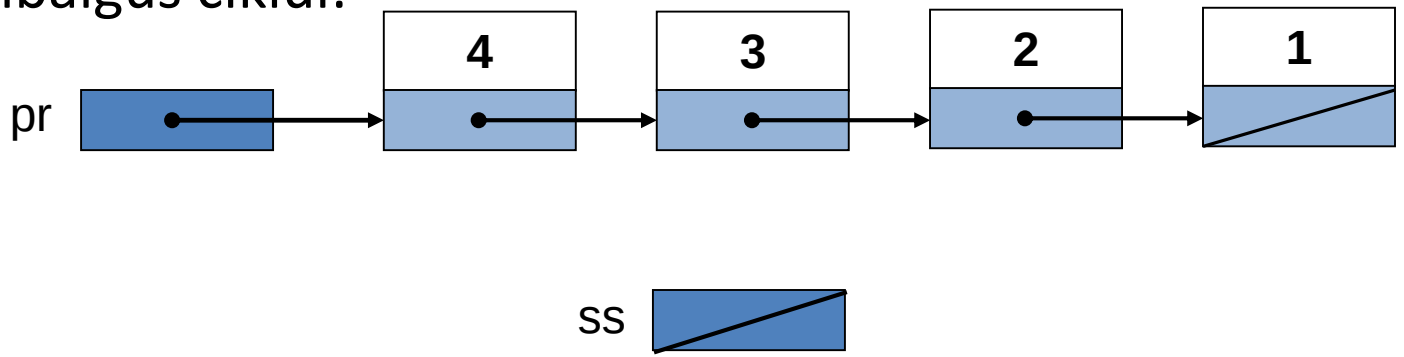
4-as ciklas:



Sąrašo klasės sąrašo peržiūros ciklas 6/6

```
for (A.Pradžia(); A.Yra(); A.Kitas())
{
    ... // Veiksmai su A.ImtiDuomenis() (spausdinimas, skaičiavimai)
}
```

Pasibaigus ciklui:



Sąrašo klasės sąrašo spausdinimas

ktu informatikos fakultetas 1922

```
// Sąrašo duomenys spausdinami faile
// fv – duomenų failo vardas
// A - sąrašo objekto nuoroda
// koment - komentaras
static void Spausdinti(string fv, Sąrašas A, string koment)
{
    using (var failas = new StreamWriter(fv, true))
    {
        failas.WriteLine(koment);
        // Sąrašo peržiūra, panaudojant sąsajos metodus
        for (A.Pradžia(); A.Yra(); A.Kitas())
        {
            failas.WriteLine("{0, 3:d}", A.ImtiDuomenis());
        }
        failas.WriteLine();
    }
}
```

Sąrašo klasės sąrašo vidurkio skaičiavimas

ktu informatikos fakultetas 1922

```
// Suskaičiuojamas ir grąžinamas sąrašo A elementų reikšmių vidurkis
// A - sąrašo objekto nuoroda
static double Vidurkis(Sąrašas A)
{
    int suma = 0;
    int kiekis = 0;
    for (A.Pradžia(); A.Yra(); A.Kitas())
    {
        suma += A.ImtiDuomenis();
        kiekis++;
    }
    if (kiekis != 0)
        return (double)suma / kiekis;
    else
        return 0.0;
}
```

Sąrašo klasės objektas 1/2

```
. . .
const string CFd = @"..\..\Duomenys.txt";
const string CFr = @"..\..\Rezultatai.txt";
. . .
Sąrašas A = SkaitytiAtv(CFd);
Spausdinti(CFr, A, "Atvirkštinis sąrašas");
Console.WriteLine("Reikšmių vidurkis {0, 7:f2}", Vidurkis(A));

Sąrašas T = SkaitytiTiesiog(CFd);
Spausdinti(CFr, T, "Tiesioginis sąrašas");
Console.WriteLine("Reikšmių vidurkis {0, 7:f2}", Vidurkis(T));

. . .
```

Sąrašo klasės objektas 2/2

Duomenų failas:

```
1  2  3  4
```

Rezultatų failas:

```
Atvirkštinis sąrašas
  4
  3
  2
  1
Reikšmių vidurkis   2,50
Tiesioginis sąrašas
  4
  3
  2
  1
Reikšmių vidurkis   2,50
```

ktu informatikos fakultetas 1922

Šioje temoje susipažinote:

1. Rekursine duomenų struktūra
2. Susietųjų sąrašų tipais
3. Susietojo sąrašo samprata
4. Elemento sukūrimu
5. Sąrašo formavimu
6. Sąrašo peržiūra
7. Sąrašo klase

Klausimai?