ktu informatikos fakultetas 1922

T05. Paveldėjimas

2 ak. val.

ktu informatikos fakultetas 1922

Temos klausimai

1. Bazinė ir išvestinė klasės.
2. Konstruktorių paveldimumas.
3. protected paveldėjime.
4. „Ar daugiau“ užklojimas.
5. Numatytosios parametrų reikšmės.

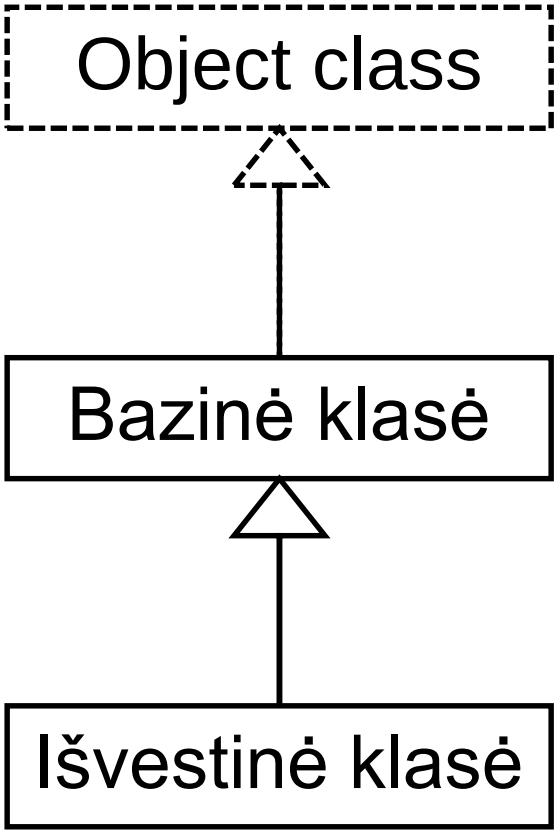
ktu informatikos fakultetas 1922

Sąvokos

- Paveldėjimas – esamo programinio kodo (klasių) panaudojimo forma, kai esamas kodas papildomas naujais duomenimis (kintamaisiais, savybėmis) ir naujais veiksmais (metodais).
- Paveldimumas realizuojamas, išvedant naujas klases iš jau esamų klasių.
- Esama klasė vadinama **bazine** klase (super klase, tėvo klase).
- Nauja klasė, išvedama esamos pagrindu, vadinama **išvestine** klase (poklasiu, vaiko klase).
- Išvestinė klasė gali būti bazine tolimesnei klasei.

ktu informatikos fakultetas 1922

Žymėjimas



ktu informatikos fakultetas 1922

Esminės savybės

Paveldėjimas naudojamas, kai reikia sukurti naują klasę, kuri nedaug skiriasi nuo kitos, anksčiau sukurtos.

Išvestinė klasė paveldi visas bazinės klasės savybes (kintamuosius) ir elgseną (metodus).

ktu informatikos fakultetas 1922

Bazinė ir išvestinės klasės

```
public class BazinėKlasė
{
    // ...
}
//-----------------------------------------
public class IšvestinėKlasė : BazinėKlasė
{
    // ...
}
//-----------------------------------------
public class IšvestinėKlasėAntra : IšvestinėKlasė
{
    // ...
}
```

Bazinės ir išvestinių klasių pavyzdys

ktu informatikos fakultetas 1922

```
class SpausdintasDokumentas
{
    // ...
}
//-----------------------------------------
class Knyga : SpausdintasDokumentas
{
    // ...
}
//-----------------------------------------
class KnygaMinkstaisVirseliais : Knyga
{
    // ...
}
```

ktu informatikos fakultetas 1922

Klasių hierarchija

Klasės, tarpusavyje susietos paveldimumu, sudaro klasių *hierarchiją*.

Klasių hierarchijos viršuje yra abstraktesnės klasės, o einant žemyn klasės darosi konkretesnės.

Klasės, esančios hierarchijos viduryje, yra kartu ir bazinės ir išvestinės.

Dažnai išvestinės klasės turi "*...yra tam tikros rūšies...*" ryšį su bazine klase:

`Knyga` yra tam tikros rūšies **`SpausdintasDokumentas`**,

`KnygaMinkstaisVirseliais` yra tam tikros rūšies **`Knyga`**.

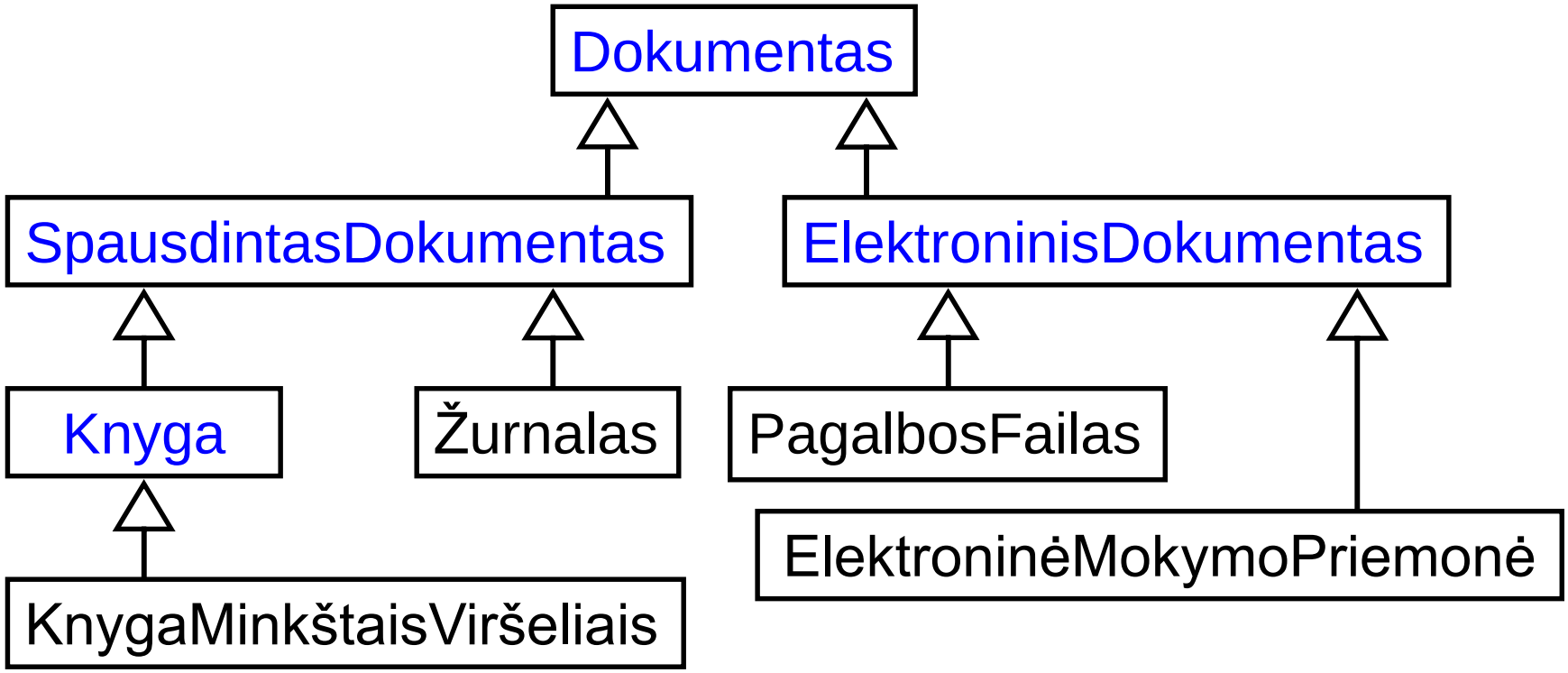
ktu informatikos fakultetas 1922

Paveldėjimo nauda

Klasių hierarchijos mechanizmas leidžia bendrąsias panašių klasių savybes ir funkcionalumą apjungti bazinėse klasėse, o išvestinėse klasėse palikti tik specifines:

- palengvina uždavinio struktūrizavimą;
- leidžia sumažinti kodo pasikartojimą, sutaupyti atminties ir laiko sąnaudas.

Klasių hierarchijos pavyzdys



Klasė (pvz.: Dokumentas) gali būti bazinė daugeliui klasių.

Klasė (pvz.: Knyga) gali būti vienu metu bazine ir išvestine.

ktu informatikos fakultetas 1922

Papildomos sąvokos

Tiesioginė bazinė klasė – tai klasė, iš kurios išreikštai išvestinė klasė paveldi.

Netiesioginė bazinė klasė – tai klasė, iš kurios paveldima per klasių hierarchiją.

Klasių hierarchijos pradžioje yra klasė `Object` (iš `System`), kurią kiekviena klasė tiesiogiai ar netiesiogiai paveldi.

Vienetinis paveldėjimas – tiesiogiai paveldima tik iš vienos klasės.

Neišreikštas daugybinis paveldėjimas įmanomas, paveldint sąsajas.

ktu informatikos fakultetas 1922

Paveldėjimo pavyzdžiai

- Gyvūnas – šuo, katinas, arklys.
- Paukštis – žvirblis, erelis, pingvinas.
- Forma – apskritimas, trikampis, stačiakampis.
- Paskola – paskolaAutomobiliui, paskolaButui.
- Tarnautojas – pagrindinis, valandinis, kviestinis.
- Sąskaita – einamoji, terminuota.

Konstruktorių paveldimumas

- Išvestinė klasė nepaveldi bazinės klasės konstruktorių.
- Bazinės klasės konstruktorius visada įvykdomas, kai sukuriamas naujas išvestinės klasės objektas.

Svarbu suprasti kaip kuriamas išvestinės klasės objektas!

Išvestinės klasės objekto sukūrimas

ktu informatikos fakultetas 1922

- Objektui yra skiriama atmintis.
- Vykdomi klasių konstruktoriai:

  pirmasis vykdomas bazinės klasės konstruktorius;

  po to vykdomas išvestinės klasės konstruktorius.
- Kiekvienos išvestinės klasės konstruktorius gali pasikliauti pilnu savo bazinės klasės sukonstravimu.

Kuris bazinės klasės konstruktorius vykdomas, jeigu jų yra keli?

Bazinės klasės konstruktoriaus parinkimas

ktu informatikos fakultetas 1922

- Nenurodžius, objekto konstravimo metu vykdomas numatytasis bazinės klasės konstruktorius (*prisiminkite*: tai konstruktorius be parametrų).
- Kai bazinėje klasėje nėra konstruktoriaus be parametrų, būtina nurodyti, kurį bazinės klasės konstruktorių vykdyti.
- Konstruktoriaus iškvietimo formatas:

public Išvestinės_klasės_konstruktorius(parametrai) **:**

base(argumentai)

{ ... }

Kreipinys į bazinės klasės konstruktorių

Tai ypač svarbu, jei turim perduoti parametrą (-us) bazinei klasei.

Konstruktoriai paveldėjime (1/9)

ktu informatikos fakultetas 1922

Visos bazinės klasės **tiesiogiai** paveldi `Object` klasę, esančią **System** vardų erdvėje.

```
public class BazinėKlasė : Object
{
    public BazinėKlasė()            // konstruktorius be parametrų
    {
        Console.WriteLine("Dirba bazinės klasės konstruktorius.");
    }
    public void Spausdinti()
    {
        Console.WriteLine("Dirba bazinės klasės metodas Spausdinti().");
    }
}
public class IšvestinėKlasė : BazinėKlasė
{
    public IšvestinėKlasė()   // arba public IšvestinėKlasė() : base()
    {
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
    }
    public void Spausdinti1()
    {
        Console.WriteLine("Dirba išvestinės klasė metodas Spausdinti1().");
    }
}
```

Kreipinys į bazinės klasės konstruktorių

Konstruktoriai paveldėjime (2/9)

```
...
IšvestinėKlasė objIšvKlasės = new IšvestinėKlasė();
objIšvKlasės.Spausdinti();
objIšvKlasės.Spausdinti1();
...
```

Ekrane matysite:

```
Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
Dirba bazinės klasės metodas Spausdinti().
Dirba išvestinės klasė metodas Spausdinti1().
```

Konstruktoriai paveldėjime (3/9)

ktu informatikos fakultetas 1922

Tokiu būdu savybę uždraudžiama keisti išorėje.

```
public class BazinėKlasė
{
    public string eiluteBaz { get; set; } // arba { get; private set; }

    public BazinėKlasė()    // konstruktorius be parametrų
    {
        Console.WriteLine("Dirba bazinės klasės konstruktorius.");
    }

    public BazinėKlasė(string eilute)   // konstruktorius su parametrais
    {
        Console.WriteLine("Dirba bazinės klasės konstruktorius.");
        eiluteBaz = eilute;
    }

    public void Spausdinti()
    {
        Console.WriteLine("Dirba bazinės klasės metodas Spausdinti():");
        Console.WriteLine("{0}", eiluteBaz);
    }
}
```

Konstruktoriai paveldėjime (4/9)

```
public class IšvestinėKlasė : BazinėKlasė
{
    public IšvestinėKlasė() : base()
    {
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
    }
    public IšvestinėKlasė(string eil) : base(eil)
    {
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
        Console.WriteLine(" Jis kreipiasi į bazinės klasės konstuktorių.");
    }
    public new void Spausdinti()  // su new paslepiam bazinės klasės metodą Spausdinti()
    {
        Console.WriteLine("Dirba išvestinės klasės metodas Spausdinti():");
        Console.WriteLine(eiluteBaz);
        Console.WriteLine(" Jis kreipiasi į baz. klasės metodą Spausdinti().");
        base.Spausdinti();
    }
}
```

Konstruktoriai paveldėjime (5/9)

```
...
IšvestinėKlasė objIšvKlasės = new IšvestinėKlasė(“DUOMENYS");
objIšvKlasės.Spausdinti();
Console.WriteLine("Dirba klasės Program metodas Main().");
Console.WriteLine("Jis parodo bazinės klasės savybės reikšmę: {0}.", objIšvKlasės.eiluteBaz);
...
// Jei būtų bazinėje klasėje būtų public string eiluteBaz { get; private set; }, tai toks priskyrimas nebūtų galimas !
objIšvKlasės.eiluteBaz = “KITI DUOMENYS";  // Klaida!
...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Konstruktoriai paveldėjime (6/9)

```
Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
  Jis kreipiasi į bazinės klasės konstuktorių.
Dirba išvestinės klasės metodas Spausdinti():
DUOMENYS
  Jis kreipiasi į bazinės klasės metodą Spausdinti().
Dirba bazinės klasės metodas Spausdinti():
DUOMENYS
Dirba klasės Program metodas Main().
Jis parodo bazinės klasės savybės reikšmę: DUOMENYS.
```

Konstruktoriai paveldėjime (7/9)

ktu informatikos fakultetas 1922

Išvestinės klasės savybė

```
public class IšvestinėKlasė : BazinėKlasė
{
    public int skaiciusIsvest { get; private set; }

    public IšvestinėKlasė() //: base() - galima ir nerašyti, vis tiek kreipsis
    {
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
        Console.WriteLine(" Jis kreipiasi į baz. klasės konstruktorių base().");
    }
    // Reikia užrašyti kreipinį bazinės klasės konstruktorių
    public IšvestinėKlasė(string eil, int skaicius) : base(eil)
    {
        skaiciusIsvest = skaicius;
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
        Console.WriteLine(" Jis kreipiasi į baz. klasės konstruktorių.");
    }
    public new void Spausdinti()  // new paslepia bazinės klasės metodą
    {
        Console.WriteLine("Dirba išvestinės klasės metodas Spausdinti().");
        Console.WriteLine(" Jis kreipiasi į bazinės klasės metodą Spausd.().");
        base.Spausdinti();
        Console.WriteLine("{0}", skaiciusIsvest);
    }
}
```

Konstruktoriai paveldėjime (8/9)

ktu informatikos fakultetas 1922

```
...
IšvestinėKlasė objIšvKlasės = new IšvestinėKlasė("DUOMENYS",
                                                  99999);
objIšvKlasės.Spausdinti();
IšvestinėKlasė objIšvKlasės1 = new IšvestinėKlasė();
objIšvKlasės1.Spausdinti();
...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Konstruktoriai paveldėjime (9/9)

ktu informatikos fakultetas 1922

```
Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
  Jis kreipiasi į bazinės klasės konstuktorių base(eil).
Dirba išvestinės klasės metodas Spausdinti().
  Jis kreipiasi į bazinės klasės metodą Spausdinti().
Dirba bazinės klasės metodas Spausdinti():
DUOMENYS
99999
Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
  Jis kreipiasi į bazinės klasės konstuktorių base().
Dirba išvestinės klasės metodas Spausdinti().
  Jis kreipiasi į bazinės klasės metodą Spausdinti().
Dirba bazinės klasės metodas Spausdinti():

0
```

Paveldėjimo pavyzdys 1 (1/5)

Sukurkite bazinę klasę Apskritimas su savybe *Spindulys* bei reikalingus konstruktorius ir metodus.

Sukurkite išvestinę klasę Cilindras, kuri turėtų savybę *Aukštinė*, reikalingus konstruktorius ir metodus.

Pademonstruokite sukurtų klasių konstruktorių ir metodų darbą.

Paveldėjimo pavyzdys 1 (2/5)

```
class Apskritimas : Object
{
    public double Spindulys { get; private set; }

    public Apskritimas()           // konstruktorius be parametrų
    {
        Spindulys = 1;
    }
    public Apskritimas(double r)  // konstruktorius su parametru
    {
        Spindulys = r;
    }
    public double Plotas()
    {
        return 3.141592653589 * Spindulys * Spindulys;
    }
    public override string ToString()    // vietoj Metodo spausdinti()
    {
        return string.Format(" spindulys = {0, 6:f2} ", Spindulys);
    }
}
```

Paveldėjimo pavyzdys 1 (3/5)

```
class Cilindras : Apskritimas
{
    public double Aukstine { get; private set; }

    public Cilindras() : base()   // konstruktorius be parametrų
    {
        Aukstine = 1;
    }
    public Cilindras(double r, double h) : base(r)  // konstruktorius su parametrais
    {
        Aukstine = h;
    }
    public double Turis()
    {
        return Plotas() * Aukstine;
    }
    public override string ToString()     // vietoj Metodo spausdinti()
    {
        return string.Format("{0} aukštinė = {1, 6:f2}", base.ToString(),
                                                         Aukstine);
    }
}
```

Paveldėjimo pavyzdys 1 (4/5)

```
...
Apskritimas apskritimasA = new Apskritimas(10);
Console.WriteLine("Apskritimo A duomenys: {0}", apskritimasA.ToString());
Console.WriteLine("Apskritimo A ribojamas plotas: {0, 10:f3}",
                                                  apskritimasA.Plotas());
Apskritimas apskritimasB = new Apskritimas();
Console.WriteLine("Apskritimo B duomenys: {0}", apskritimasB.ToString());
Console.WriteLine("Apskritimo B ribojamas plotas: {0, 10:f3}",
                                                  apskritimasB.Plotas());

Cilindras cilindrasA = new Cilindras();
Console.WriteLine("Cilindro A duomenys: {0}", cilindrasA.ToString());
Console.WriteLine("Cilindro A tūris ={0, 10:f3}", cilindrasA.Turis());
Cilindras cilindrasB = new Cilindras(10, 50);
Console.WriteLine("Cilindro B duomenys: {0}", cilindrasB.ToString());
Console.WriteLine("Cilindro B tūris ={0, 10:f3}", cilindrasB.Turis());

...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Paveldėjimo pavyzdys 1 (5/5)

```
Apskritimo A duomenys:  spindulys =  10,00
Apskritimo A ribojamas plotas:    314,159

Apskritimo B duomenys:  spindulys =   1,00
Apskritimo B ribojamas plotas:      3,142

Cilindro A duomenys:  spindulys =   1,00  aukštinė =   1,00
Cilindro A tūris =     3,142

Cilindro B duomenys:  spindulys =  10,00  aukštinė =  50,00
Cilindro B tūris = 15707,963
```

Konstruktoriai su numatytosiomis parametrų reikšmėmis (1/6)

ktu informatikos fakultetas 1922

```
public class BazinėKlasė
{
    public int    ABaz { get; private set; }
    public double BBaz { get; private set; }
    public string CBaz { get; private set; }

    // Konstruktorius
    public BazinėKlasė(int a = 1, double b = 0.001, string c = "###")
    {
        Console.WriteLine("Dirba bazinės klasės konstruktorius.");
        this.ABaz = a;
        this.BBaz = b;
        this.CBaz = c;
    }
    public void Spausdinti()
    {
        Console.WriteLine("{0, 3:d}   {1, 12:f5}   {2, -10}",
                          ABaz, BBaz, CBaz);
    }
}
```

Konstruktorius su trimis numatytosiomis reikšmėmis

Konstruktoriai su numatytosiomis parametrų reikšmėmis (2/6)

ktu informatikos fakultetas 1922

```
...
BazinėKlasė objBaz1 = new BazinėKlasė();
objBaz1.Spausdinti();
BazinėKlasė objBaz2 = new BazinėKlasė(200, 2.1, "@@@");
objBaz2.Spausdinti();
BazinėKlasė objBaz3 = new BazinėKlasė(300, 3.1);
objBaz3.Spausdinti();
BazinėKlasė objBaz4 = new BazinėKlasė(400);
objBaz4.Spausdinti();
//BazinėKlasė objBaz5 = new BazinėKlasė(500, "$$$$$$$$$$");   // KLaida!
//objBaz5.Spausdinti();
...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Kreipiantis į konstruktorių su numatytosiomis parametrų reikšmėmis galima nenaudoti argumentų iš kairės į dešinę. Kitokie kreipinių variantai yra negalimi!

Konstruktoriai su numatytosiomis parametrų reikšmėmis (3/6)

```
Dirba bazinės klasės konstruktorius.
  1         0,00100   ###

Dirba bazinės klasės konstruktorius.
200         2,10000   @@@

Dirba bazinės klasės konstruktorius.
300         3,10000   ###

Dirba bazinės klasės konstruktorius.
400         0,00100   ###
```

Konstruktoriai su numatytosiomis parametrų reikšmėmis (4/6)

ktu informatikos fakultetas 1922

```
public class IšvestinėKlasė : BazinėKlasė
{
    public int    AIsv { get; private set; }
    public string BIsv { get; private set; }
    // Konstruktorius
    public IšvestinėKlasė(int aB, double bB, string cB,
                          int aI = 11, string bI = "***") : base(aB, bB, cB)
    {
        this.AIsv = aI;
        this.BIsv = bI;
        Console.WriteLine("Dirba išvestinės klasės konstruktorius.");
    }
    public new void Spausdinti()
    {
        Console.WriteLine("{0, 3:d}  {1, -10}", AIsv, BIsv);
        base.Spausdinti();
    }
}
```

Konstruktorius su dviem numatytosiomis reikšmėmis

Konstruktoriai su numatytosiomis parametrų reikšmėmis (5/6)

```
...
IšvestinėKlasė objIsv1 = new IšvestinėKlasė(100, 1.1, "-----");
objIsv1.Spausdinti();
IšvestinėKlasė objIsv2 = new IšvestinėKlasė(100, 1.1, "-----",
                                            200, "+++++");
objIsv2.Spausdinti();

// Klaida!
//IšvestinėKlasė objIsv3 = new IšvestinėKlasė(100, 1.1, "-----", "+++++");
//objIsv3.Spausdinti();
...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Konstruktoriai su numatytosiomis parametrų reikšmėmis (6/6)

```
Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
 11  ***
100        1,10000   -----

Dirba bazinės klasės konstruktorius.
Dirba išvestinės klasės konstruktorius.
200  +++++
100        1,10000   -----
```

Konstruktoriaus su numatytaja reikšme pavyzdys 2 (1/4)

ktu informatikos fakultetas 1922

```
class Apskritimas : Object
{
    public double Spindulys { get; private set; }

    // Konstruktorius su numatytaja reikšme
    public Apskritimas(double r = 1)
    {
        Spindulys = r;
    }
    public double Plotas()
    {
        return 3.141592653589 * Spindulys * Spindulys;
    }
    public override string ToString()    // vietoj Metodo spausdinti()
    {
        return string.Format(" spindulys = {0, 6:f2} ", Spindulys);
    }
}
```

Vietoje dviejų konstruktorių - vienas konstruktorius su numatytaja reikšme

Konstruktoriaus su numatytaja reikšme pavyzdys 2 (2/4)

ktu informatikos fakultetas 1922

```
class Cilindras : Apskritimas
{
    public double Aukstine { get; private set; }

    public Cilindras() : base()   // konstruktorius be parametrų
    {
        Aukstine = 1;
    }
    // Konstruktorius su numatytaja reikšme
    public Cilindras(double r, double h = 1) : base(r)
    {
        Aukstine = h;
    }
    public double Turis()
    {
        return Plotas() * Aukstine;
    }
    public override string ToString()     // vietoj Metodo spausdinti()
    {
        return string.Format("{0} aukštinė = {1, 6:f2}", base.ToString(),
                                                         Aukstine);
    }
}
```

Konstruktoriaus su numatytaja reikšme pavyzdys 2 (3/4)

ktu informatikos fakultetas 1922

```
...
Apskritimas apskritimasA = new Apskritimas(10);
Console.WriteLine("Apskritimo A duomenys: {0}", apskritimasA.ToString());
Console.WriteLine("Apskritimo A ribojamas plotas: {0, 10:f3}",
                                                  apskritimasA.Plotas());
Apskritimas apskritimasB = new Apskritimas();
Console.WriteLine("Apskritimo B duomenys: {0}", apskritimasB.ToString());
Console.WriteLine("Apskritimo B ribojamas plotas: {0, 10:f3}",
                                                  apskritimasB.Plotas());

Cilindras cilindrasA = new Cilindras();
Console.WriteLine("Cilindro A duomenys: {0}", cilindrasA.ToString());
Console.WriteLine("Cilindro A tūris ={0, 10:f3}", cilindrasA.Turis());
Cilindras cilindrasB = new Cilindras(10, 50);
Console.WriteLine("Cilindro B duomenys: {0}", cilindrasB.ToString());
Console.WriteLine("Cilindro B tūris ={0, 10:f3}", cilindrasB.Turis());
Cilindras cilindrasC = new Cilindras(10);
Console.WriteLine("Cilindro C duomenys: {0}", cilindrasC.ToString());
Console.WriteLine("Cilindro C tūris ={0, 10:f3}", cilindrasC.Turis());

...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Konstruktoriaus su numatytaja reikšme pavyzdys 2 (3/4)

```
Apskritimo A duomenys:  spindulys =  10,00
Apskritimo A ribojamas plotas:    314,159

Apskritimo B duomenys:  spindulys =   1,00
Apskritimo B ribojamas plotas:      3,142

Cilindro A duomenys:  spindulys =   1,00  aukštinė =   1,00
Cilindro A tūris =     3,142

Cilindro B duomenys:  spindulys =  10,00  aukštinė =  50,00
Cilindro B tūris = 15707,963

Cilindro C duomenys:  spindulys =  10,00  aukštinė =   1,00
Cilindro C tūris =   314,159
```

Užklotas operatorius paveldėjime (1/4)

ktu informatikos fakultetas 1922

```
public class Auto : Object
{
    public int Metai { get; private set; }  // pagaminimo metai
    public int Rida  { get; private set; }  // kiek pravažiuota (tūkst. km.)
    public Auto(int metai = 2016, int rida = 0)
    {
        this.Metai = metai;
        this.Rida = rida;
    }
    public override string ToString()    // vietoj Metodo spausdinti()
    {
     return string.Format(" pagaminimo metai =  {0, 4:d};  rida = {1, 7:d} km;",
                                                    Metai, Rida);
    }
}
```

Užklotas operatorius paveldėjime (2/4)

ktu informatikos fakultetas 1922

```
public class KrovAuto : Auto
{
    public string Modelis { get; private set; }
    public double KGalia  { get; private set; }   // keliamoji galia (tonomis)
    public KrovAuto(int metai, int rida, string modelis, double galia)
             : base(metai, rida)
    {
        this.Modelis = modelis;
        this.KGalia = galia;
    }
    public static bool operator >(KrovAuto pirmas, KrovAuto antras)
    {
        return pirmas.Metai > antras.Metai ||
               pirmas.Metai == antras.Metai && pirmas.KGalia > antras.KGalia;
    }
    public static bool operator <(KrovAuto pirmas, KrovAuto antras)
    {
        return pirmas.Metai < antras.Metai ||
               pirmas.Metai == antras.Metai && pirmas.KGalia < antras.KGalia;
    }
    public override string ToString()     // vietoj Metodo spausdinti()
    {
        return string.Format(" modelis: {0};  kel. galia = {1, 6:f2}; \n{2} ",
                             Modelis, KGalia, base.ToString());
    }
}
```

Užklotas operatorius paveldėjime (3/4)

ktu informatikos fakultetas 1922

```
...
Auto AutoMob = new Auto();
Console.WriteLine("AutoMob duomenys:\n{0}", AutoMob.ToString());
KrovAuto Volvo = new KrovAuto(2012, 125000, "Volvo FH12", 24.5);
Console.WriteLine("Volvo duomenys:\n{0}", Volvo.ToString());
KrovAuto Scania = new KrovAuto(2002, 450000, "Scania FH12", 19.5);
Console.WriteLine("Scania duomenys:\n{0}", Scania.ToString());
Console.WriteLine();
if (Volvo > Scania)
    Console.WriteLine("Volvo > už Scania.");
else
    Console.WriteLine("Volvo < už Scania.");
...
```

Naudojamas užklotas operatorius **>**

Ekrane matysite (žiūr. kitoje skaidrėje):

Užklotas operatorius paveldėjime (4/4)

```
AutoMob duomenys:
 pagaminimo metai =  2016;  rida =       0 km;

Volvo duomenys:
 modelis: Volvo FH12;  keliamoji galia =  24,50;
 pagaminimo metai =  2012;  rida =  125000 km;

Scania duomenys:
 modelis: Scania FH12;  keliamoji galia =  19,50;
 pagaminimo metai =  2002;  rida =  450000 km;

Volvo > už Scania.
```

Klasės elementų matomumas

Prisiminkite, klasės elementai (nariai) gali turėti matomumo požymius:

- `private`
- `public`
- `protected` (*dar nenaudotas*)

Matomumo požymis protected

- Klasės elementų matomumas toks pats, kaip ir **`private`**, išskyrus išvestines klases.
- Išvestinė klasė bazinės klasės **`protected`** elementus „mato“ taip pat kaip **`public`**.
- Šiuo požymiu piktnaudžiauti nereikia.

Klasės elementų matomumas

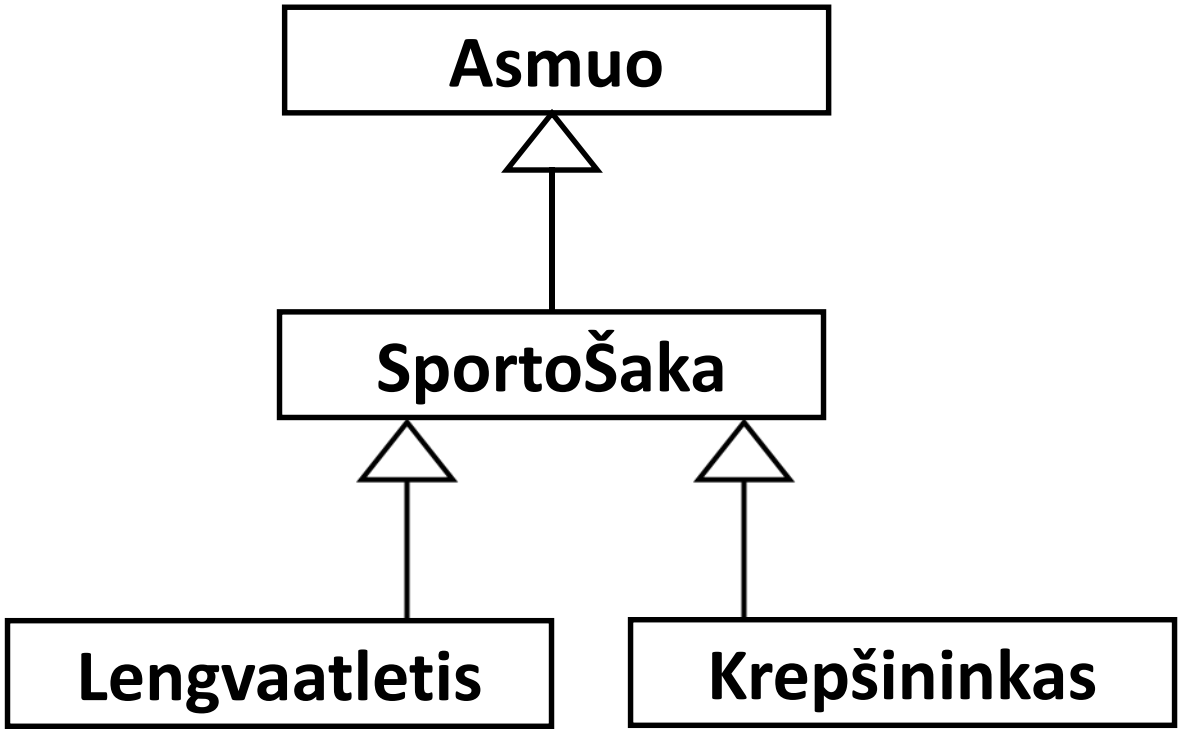
Kas kreipiasi \ Matomumo požymis	`public`	`protected`	`private`
Tos pačios klasės nariai	**taip**	**taip**	**taip**
Išvestinės klasės nariai	**taip**	**taip**	**ne**
Ne nariai *	**taip**	**ne**	**ne**

* Ne nariai – visi klasės atžvilgiu išoriniai metodai

Klasės savybės `protected` paveldėjime pavyzdys (1/8)

- Sukurkite reikalingas klases sportininkų (lengvaatlečių ir krepšininkų) duomenims saugoti.
- Klases tarpusavyje susiekite paveldėjimu.
- Pademonstruokite šių klasių veikimą.

Klasės savybės `protected` paveldėjime pavyzdys (2/8)



Klasės savybės protected paveldėjime pavyzdys (3/8)

ktu informatikos fakultetas 1922

```
class Asmuo : Object
{
    protected string Vardas { get; private set; }
    protected int    Amzius { get; private set; }

    public Asmuo(string vardas, int amzius) : base()
    {
        Vardas = vardas;
        Amzius = amzius;
    }
    public override string ToString()
    {
        return string.Format(" vardas = {0};\n amžius = {1, 2:d};\n",
                             Vardas, Amzius);
    }
}
```

Klasės savybės protected paveldėjime pavyzdys (4/8)

```
class SportoSaka : Asmuo
{
    public string SSPavadinimas { get; private set; }

    public SportoSaka(string vardas, int amzius, string sspavad)
           : base(vardas, amzius)
    {
        SSPavadinimas = sspavad;
    }
    public override string ToString()    // vietoj Metodo spausdinti()
    {
        return string.Format("{0} sporto šaka = {1};\n",
                             base.ToString(), SSPavadinimas);
    }
}
```

Klasės savybės protected paveldėjime pavyzdys (5/8)

```
class Lengvaatletis : SportoSaka
{
    public string Rungtis  { get; private set; }
    public double Rekordas { get; private set; }
    public string Matas    { get; private set; }

    public Lengvaatletis(string vardas, int amzius, string sspavad,
                         string rungtis, double rekordas, string matas)
           : base(vardas, amzius, sspavad)
    {
        this.Rungtis = rungtis;
        this.Rekordas = rekordas;
        this.Matas = matas;
    }
    public override string ToString()
    {
        return string.Format("{0} rungtis = {1};\n rekordas = {2, 7:f3} {3};",
                             base.ToString(), Rungtis, Rekordas, Matas);
    }
}
```

Klasės savybės protected paveldėjime pavyzdys (6/8)

```
class Krepsininkas : SportoSaka
{
    public string Pozicija { get; set; }
    public int    Ugis     { get; set; }  // centimetrais

    public Krepsininkas(string vardas, int amzius, string sspavad,
                        string pozicija, int ugis)
           : base(vardas, amzius, sspavad)
    {
        this.Pozicija = pozicija;
        this.Ugis = ugis;
    }
    public override string ToString()    // vietoj Metodo spausdinti()
    {
        return string.Format("{0} pozicija = {1};\n ugis = {2, 3:d};",
                             base.ToString(), Pozicija, Ugis);
    }
}
```

Klasės savybės protected paveldėjime pavyzdys (7/8)

```
...
Lengvaatletis Lengv = new Lengvaatletis("Greta Greitoji", 21,
                "Lengvoji atletika", "100 metrų bėgimas", 9.99, "s");
Console.WriteLine("Sportininkė:\n{0}", Lengv.ToString());

Krepsininkas Kreps = new Krepsininkas("Tomas Taiklusis", 19,
                                      "Krepšinis", "Centras", 211);
Console.WriteLine("Sportininkas:\n{0}", Kreps.ToString());

Kreps.Amzius = Kreps.Amzius + 1;
Kreps.Ugis = Kreps.Ugis + 4;
...
```

Šioje eilutėje kompiliatorius rodys klaidą, o sekančioje eilutėje nerodys. **Kodėl?**

Ekrane matysite (žiūr. kitoje skaidrėje):

Klasės savybės protected paveldėjime pavyzdys (8/8)

```
Sportininkė:
 vardas = Greta Greitoji;
 amžius = 21;
 sporto šaka = Lengvoji atletika;
 rungtis = 100 metrų bėgimas;
 rekordas =   9,990 s;

Sportininkas:
 vardas = Tomas Taiklusis;
 amžius = 19;
 sporto šaka = Krepšinis;
 pozicija = Centras;
 ugis = 211;
```

Paveldėjimo pavyzdys (1/6)

Asmuo identifikuojamas pagal jo pavardę ir vardą.

Mama gali turėti keletą vaikų.

Panaudodami paveldėjimą sukurkite klases, šios rūšies informacijai saugoti.

Pademonstruokite sukurtų klasių objektų sukūrimą.

Paveldėjimo pavyzdys (2/6)

```
class Asmuo : Object
{
    public string Vardas  { get; private set; }
    public string Pavarde { get; private set; }
    public Asmuo() : base()
    {
    }
    public Asmuo(string vardas, string pavarde) : base()
    {
        this.Vardas = vardas;
        this.Pavarde = pavarde;
    }
    public override string ToString()
    {
        return string.Format(" vardas = {0}; pavardė = {1};",
                             Vardas, Pavarde);
    }
}
```

Paveldėjimo pavyzdys (3/6)

```
class Mama : Asmuo
{
    const int CMax = 10;
    private Asmuo[] Vaikai;
    public int Kiek { get; private set; }
    public Mama() : base()
    {
        Vaikai = new Asmuo[CMax];
        Kiek = 0;
    }
    public Mama(string mamosVardas, string mamosPavarde)
         : base(mamosVardas, mamosPavarde)
    {
        Vaikai = new Asmuo[CMax];
        Kiek = 0;
    }
    // ... tęsinys kitoje skaidrėje
}
```

Paveldėjimo pavyzdys (4/6)

```
class Mama : Asmuo
{
    // ... pradžia ankstesnėje skaidrėje
    public Asmuo Naujagimis(string vardas)
    {
        Vaikai[Kiek++] = new Asmuo(vardas, Pavarde);
        return Vaikai[Kiek - 1];
    }
    public override string ToString()
    {
        string eil = string.Format("{0}\n", base.ToString());
        eil = eil + "Ir jos vaikai:\n";
        for (int i = 0; i < Kiek; i++)
            eil = eil + Vaikai[i].ToString() + "\n";
        return eil;
    }
}
```

Paveldėjimo pavyzdys (5/6)

```
...
Mama mama = new Mama(„Aldona", „Adams");
Asmuo sunus1 = mama.Naujagimis("Jonas");
Asmuo dukra1 = mama.Naujagimis("Rasa");
Asmuo sunus2 = mama.Naujagimis("Titas");
Console.WriteLine("Mama:\n{0}", mama.ToString());
...
```

Ekrane matysite (žiūr. kitoje skaidrėje):

Paveldėjimo pavyzdys (6/6)

```
Mama:
 vardas = Aldona; pavardė = Adams;
Ir jos vaikai:
 vardas = Jonas; pavardė = Adams;
 vardas = Rasa; pavardė = Adams;
 vardas = Titas; pavardė = Adams;
```

ktu informatikos fakultetas 1922

Šioje temoje susipažinote:

1. Bazine ir išvestine klasėmis.
2. Konstruktorių paveldimumu.
3. protected paveldėjime.
4. „Ar daugiau“ užklojimu.
5. Numatytosiomis parametrų reikšmėmis.

Klausimai?