ktu informatikos fakultetas 1922

T02. Objektinio programavimo samprata

1 ak. val.

ktu informatikos fakultetas 1922

Temos klausimai

1. Pasaulio atvaizdavimas objektais.
2. Objektinis programavimas, programavimo kalbos.
3. Programos kompiliavimas ir testavimas.
4. Pavyzdžio pradžia.

ktu informatikos fakultetas 1922

Pasaulio atvaizdavimas objektais

Realaus pasaulio atspindėjimas

Spręsdami uždavinius, mes operuojame pasaulio realių objektų modeliais kompiuteryje, atspindinčiais šių objektų savybes ir elgseną įvairiose situacijose.

Realūs objektai pasaulyje – žmonės, automobiliai, prekės ir kt., apibūdinami savybių ir elgsenos visuma.

Savybes ir elgseną nagrinėjame ne visus, o tik susijusius su konkretaus uždavinio sprendimu – abstrakcijos principas.

Savybės aprašomos duomenimis (kintamaisiais), elgsena – metodais (programa).

ktu informatikos fakultetas 1922

Objektų grupės

Programavimo esmė – aprašyti su uždaviniu susijusius realaus pasaulio objektus (jų savybes ir elgseną) bei kas su šiais objektais atliekama.

Kai kurie objektai yra panašūs. Pavyzdžiui:

- ✓ Jonas, Petras, Onutė – tai žmonės
- ✓ Sąsiuvinis, duona, atsuktuvas – tai prekės
- ✓ ...

Tikslinga tokias objektų grupes aprašyti vieną kartą ir aprašą naudoti kuriant objektus (sudaryti klasės aprašą ar tiesiog klasę).

Klasė leidžia neaprašinėti savybių ir elgsenos kiekvienam objektui, tik nurodyti jų priklausomybę klasei.

ktu informatikos fakultetas 1922

Klasės ir objektai

Klasė – tai bendrinis vienatipių objektų grupės savybių ir elgsenos aprašymas. Pavyzdžiui, Žmogus, Prekė, ...

Savybės aprašomos duomenų laukais (kintamaisiais).

Elgseną aprašoma metodais (programa).

Objektas – tai klasės konkretus atvaizdavimas kompiuteryje. Pavyzdžiui, Žmogus Jonas, Prekė sąsiuvinis, ...

Sukurdami objektą, nurodome klasę, kurios aprašymą naudojame objektui sukurti.

Vienai klasei priklausantys objektai skiriasi pavadinimais ir savybėmis (duomenų laukų turiniu). Pavyzdžiui, žmonės – savo vardais bei svoriu, pareigomis, ...

Visi tos pačios klasės objektai pasižymi vienoda elgsena.

ktu informatikos fakultetas 1922

Savybės ir elgsena

Savybės – tai objekto charakteristikos:

- ✓ žmonėms: pareigos, ūgis, svoris, atlyginimas, ...
- ✓ prekėms: pavadinimas, kodas, kaina, kiekis, ...
- ✓ ...

Elgsena – tai objekto atsakas į išorinį poveikį:

- ✓ žmonėms – elgsenos darbe aprašymas (rašyti raštą, skaičiuoti, ...), elgsenos laisvalaikiu aprašymas (eiti į kiną, skaityti knygą, ...), ...
- ✓ prekėms – elgsenos aprašymas (parduoti, pirkti, sandėliuoti, perkainuoti, ...).
- ✓ ...

ktu informatikos fakultetas 1922

Aprašymų forma (C#)

Bendrinis klasės aprašymas:

```
class Klasė {
 Savybės
 Elgsena
}
```

Objekto sukūrimo pavyzdys:

```
Klasė objektas = new Klasė();
```

ktu informatikos fakultetas 1922

Klasių aprašymas

Klasės vardas turi būti unikaliu identifikatoriumi visoje vardų galiojimo srityje.

Klasės aprašomos atskiruose .cs failuose.

Klasės elgseną apraašantys metodai aprašomi tame pačiame faile klasės viduje.

Įtraukus klasę į projektą, jos vardas galioja visoje projekto vardų srityje.

ktu informatikos fakultetas 1922

Objektinis programavimas (OP), programavimo kalbos

ktu informatikos fakultetas 1922

Esmė

Pagrindinė OP idėja – duomenų ir jais operuojančių funkcijų apjungimas į vieną visumą (objektą):

- ✓ objekto duomenys išorėje tiesiogiai neprieinami, tik per juos nuskaitantį objekto sąsajos metodą
- ✓ jei reikia keisti objekto duomenis, kreipiamės į juos įrašantį objekto sąsajos metodą

Duomenų ir juos apdorojančių funkcijų apjungimas į vieną visumą ir paslėpimas nuo išorės vadinamas **inkapsuliacija**.

Tai leidžia neprisirišti prie konkrečios objekto realizacijos.

ktu informatikos fakultetas 1922

Duomenų struktūros

Pradžioje – atmintinės ląstelės reikšmėms. Programavimo kalboje – kintamieji.

Daug vienos rūšies duomenų: atmintinės laukai nuosekliai išdėstytoms vieno tipo reikšmėms. Programavimo kalboje – masyvai.

Daug skirtingų rūšių duomenų: atmintinės laukai nuosekliai išdėstytoms skirtingų tipų reikšmėms. Programavimo kalboje – struktūros.

Struktūrų masyvai, masyvai struktūrose, ...

Dinaminiai duomenys, indeksai, katalogai, ryšiai ...

Duomenys išoriniuose informacijos nešėjuose – failų sistemos.

ktu informatikos fakultetas 1922

Objektinis programavimas (1/3)

Objektinis programavimas išpopuliarėjo paskutiniame XX amžiaus dešimtmetyje.

Objektinio programavimo populiarumo priežastis – sąsajos su natūraliais gamtos objektais ir reiškiniais.

Pagrindinė idėja – realaus pasaulio objektų (jų savybių ir elgsenos) bei manipuliavimo objektais procesų aprašymas.

Pagrindinis požymis – savybių (duomenų) ir elgsenos (programų) apjungimas objektuose, galimybė operuoti objektais bei jų dalimis.

Objektinis programavimas (2/3)

Objektiniame programavime informacija slepiama nuo išorės (inkapsuliacija). Tai leidžia kurti universalesnes programas, neprisirišant prie objektų realizacijos būdo, lengviau programą modifikuoti.

Klasės apjungiamos į hierarchijas **paveldėjimo** principu.

Gali būti kuriamos specializuotos klasės.

Kalbos: Smalltalk, C#, C++, Java, Python, ...

Objektinio programavimo privalumai išryškėja dideliuose projektuose, kai dirba daug žmonių.

ktu informatikos fakultetas 1922

Objektinis programavimas (3/3)

Klasė – tai duomenų tipas, nusakantis objekto:

- savybes: tai kintamieji (duomenys).
- elgseną: tai funkcijos (metodai).

Objektas – tai klasės tipo kintamasis. Objektas turi duomenis ir darbo su jais metodus.

ktu informatikos fakultetas 1922

Programos klasė ir `Main()` metodas

Programoje visada yra bent viena klasė.

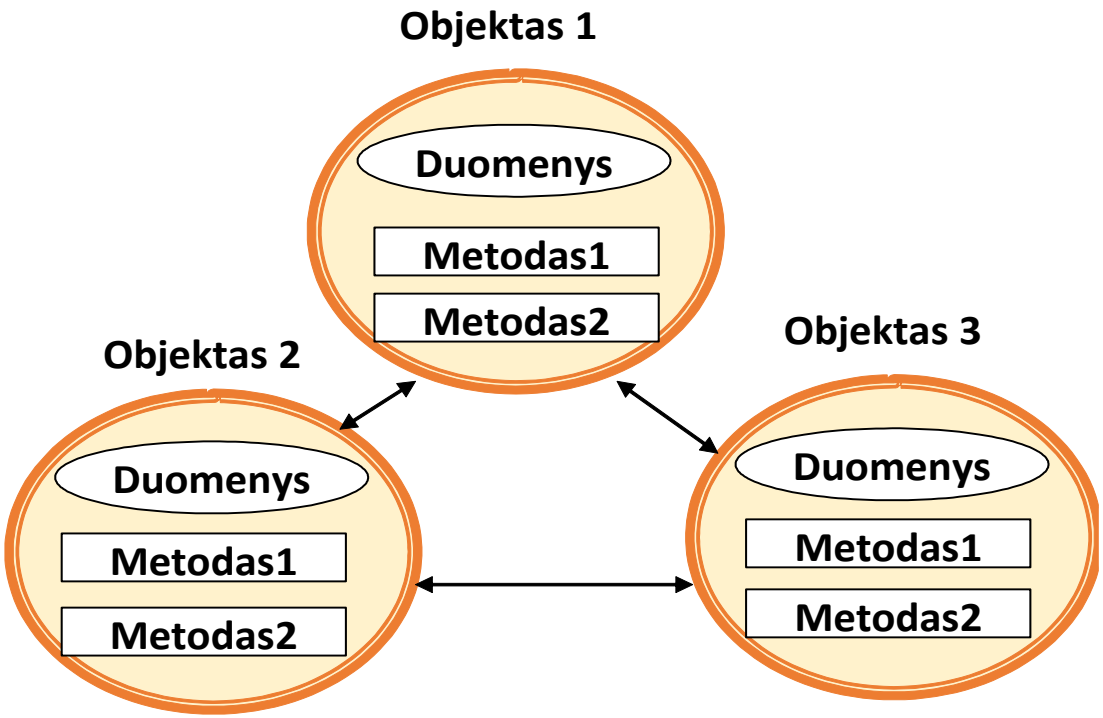
Klasė, turinti `Main()` metodą, yra pagrindinė, o `Main()` metodas yra programos pagrindinis metodas.

Visos C# programos pradedamos vykdyti nuo šio metodo.

Metodo tekstą sudaro visos žemiau antraštės riestiniuose skliaustuose { } įrašytos eilutės.

ktu informatikos fakultetas 1922

Objektinė programa



ktu informatikos fakultetas 1922

Programavimo samprata

Programavimas – tai objektų bei uždavinio sprendimo eigos aprašymas programavimo kalboje.

Sudėtingi uždaviniai reikalauja papildomų etapų:

- ✓ uždavinio analizės,
- ✓ sprendimo būdo ir kelio parinkimo,
- ✓ suderinimo su užsakovu,
- ✓ sprendimo algoritmo aprašymo,
- ✓ jo pavertimo programa,
- ✓ programos testavimo ir derinimo,
- ✓ bandomosios eksploatacijos, ...

ktu informatikos fakultetas 1922

Programavimo etapai

1. Problemos aprašymas.
2. Algoritmo parinkimas (sudarymas).
3. Kodavimas pasirinkta programavimo kalba.
4. Programos kompiliavimas, derinimas ir testavimas.
5. Programos diegimas ir eksploatavimas.

ktu informatikos fakultetas 1922

Programos kodo kompiliavimas

- Programos sintaksės ir semantikos tikrinimas.
- Transliavimas į žemo lygio kalbą.
- Vykdomojo programos failo sukūrimas.

ktu informatikos fakultetas 1922

Programos derinimas ir testavimas

- Klaidų aptikimas ir ištaisymas.
- Programos vykdymas su įvairiais duomenų rinkiniais ir rezultatų analize.

ktu informatikos fakultetas 1922

C#

Programavimo kalba – tai priemonė algoritmo užrašymui kompiuteriui suprantama forma ir jo įvykdymui kompiuteryje.

C# – tai viena iš naujausių objektinio programavimo kalbų (naujesnė už C++, Java).

C# – galinga ir efektyvi kalba (nenusileidžia C++).

C# – lengvai suprantama (lyginant su C++) ir gerai tinka programavimo mokymui (mokymuisi).

ktu informatikos fakultetas 1922

Programos kompiliavimas ir testavimas

ktu informatikos fakultetas 1922

Programos kūrimas

Programos kūrimui naudojama Visual Studio aplinka, skirta kurti objektines programas C#, C++ ir kt. kalbomis.

Visual Studio aplinka padeda programuotojui, parengdama dalį programos teksto ir programos rašymo metu nurodydama klaidas.

Raudonas banguotas pabraukimas informuoja apie klaidą: užveskite ant jo pelytės žymeklį ir Jums pateiks klaidos paaiškinimą.

Ištaisius klaidas, programą galima kompiliuoti ir testuoti.

Detaliau su aplinka susipažinsite laboratorinių darbų metu.

ktu informatikos fakultetas 1922

Programavimo terpė

- Microsoft Visual Studio (MVS) 2013.
- Kuri nors ankstesnė MVS versija (2012, 2010, 2008).
- Kuri nors vėlesnė MVS versija (2015).

ktu informatikos fakultetas 1922

Programos kompiliavimas

Nesant klaidų, programą galima kompiliuoti – specialia programa (kompiliatoriumi) paversti vykdomu mašininiu kodu.

Kompiliatorius kviečiamas klavišu F5 arba Start mygtuku, arba meniu Debug -> Start Debuging.

Programos kompiliavimo metu gali būti aptiktos naujos, dar nepastebėtos klaidos. Apie jas informacija išvedama programos lango apačioje.

Kai visos klaidos bus ištaisytos, Jūsų programa pradės darbą naujame lange.

ktu informatikos fakultetas 1922

Programos testavimas

Jei ekrane matote naują langą tamsiu fonu su Jūsų pranešimais, sveikiname – programa pradėjo darbą. Įveskite duomenis ir atlikite kitus reikiamus veiksmus.

Šiame etape galimos klaidos – stebėkite skaičiavimų eigą ir rezultatus. Gali tekti grįžti ir taisyti programą.

Net programai pateikus gerus rezultatus, skaičiavimus reikia kartoti su skirtingais duomenimis visiems galimiems programos panaudojimo atvejams.

Tai vadinama programos testavimu.

Ištestuotą programą galima rodyti dėstytojui ir atsiskaityti.

ktu informatikos fakultetas 1922

Pirmoji programa

ktu informatikos fakultetas 1922

Pirmoji programa (1/3)

```
using System;

    // Pirmoji programa C# kalba
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Labas. Tai programa C# kalba.");
            Console.WriteLine("Galiu rašyti lietuviškai.");

            Console.ReadKey();
        }
    }
}
```

Pirmoji programa (2/3)

Įtraukiama sistemos vardų erdvė

Komentarų eilutė

Sukuriama programos vardų erdvė

```
using System;

    // Pirmoji programa C# kalba
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Labas. Tai programa C# kalba.");
            Console.WriteLine("Galiu rašyti lietuviškai.");

            Console.ReadKey();
        }
    }
}
```

Pirmoji programa (3/3)

```
using System;

    // Pirmoji programa C# kalba
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Labas. Tai programa C# kalba.");
            Console.WriteLine("Galiu rašyti lietuviškai.");

            Console.ReadKey();
        }
    }
}
```

Aprašoma programos klasė

Aprašomas programos `Main()` metodas

Į konsolės langą išvedamos dvi eilutės

Konsolės langas uždaromas nuspaudus kurį nors klavišą

ktu informatikos fakultetas 1922

Pavyzdys

Pratęskime susipažinimą su klasėmis ir objektais nagrinėdami pavyzdį:

- duota informacija apie prekę: pavadinimas, kodas, vieneto kaina;
- ši prekė buvo pirkta du kartus, kiekvieną kartą skirtingus jos kiekius pirko skirtingi pirkėjai;
- reikia paskaičiuoti, už kokią sumą nupirko prekės kiekvienas pirkėjas, bei už kokią sumą nupirkta prekės iš viso.

Elementorius K-C001 13.27
Jonaitis 1
Rasienė 3

ktu informatikos fakultetas 1922

Savybių duomenų tipai

Pradžioje naudosime duomenų tipus:

- `int` – sveikieji skaičiai;
- `double` – realūs skaičiai;
- `char` – vienas simbolis;
- `string` – simbolių eilutė.

Duomenų tipų yra daugiau – susipažinsime su jais vėliau.

ktu informatikos fakultetas 1922

Savybių aprašymas

Savybės aprašomos nusirodant duomenų tipą ir išvardinant duomenų laukų pavadinimus:

```
int      a, b, kiekis;
double   t, suma, kaina;
char     simbolis;
string   eilutė, pavadinimas;
```

Norint, kad savybė išlaikytų pastovią reikšmę, reikią ją aprašyti kaip konstanta, pažymint const ir nurodant jos reikšmę:

```
const double pi = 3.14;
```

ktu informatikos fakultetas 1922

Pavyzdys: prekės klasė

```
class Prekė {
  private string  pavadinimas;
  private string  kodas;
  private double  kaina;
  ...
}
```

ktu informatikos fakultetas 1922

Pavyzdys: pirkimo klasė

```
class Pirkimas {
  private string pavardė;
  private double kiekis;
  ...
}
```

ktu informatikos fakultetas 1922

Pavyzdys: objektų sukūrimas

```
class Prekė {
  private string pavadinimas;
  private string kodas;
  private double kaina;
  ...
}
```

```
class Pirkimas {
 private string pavardė;
 private double kiekis;
 ...
}
```

```
Prekė    knyga = new Prekė();
Pirkimas Jono  = new Pirkimas(),
         Rasos = new Pirkimas();
```

ktu informatikos fakultetas 1922

Pavyzdys: problemos

Tęsti pavyzdžio negalime, nes dar nežinome kaip:

- atliekamos aritmetinės ir kitos operacijos
- dirbti su objektų duomenimis
- aprašyti objektų elgseną
- užtikrinti sąsajas tarp objektų
- nurodyti objektams pradines jų savybių reikšmes
- skaičiavimų metu keistis duomenimis su vartotoju – priimti jo nurodytus duomenis, išvesti jam skaičiavimų rezultatus.

ktu informatikos fakultetas 1922

Šioje temoje:

1. Susipažinome su realaus pasaulio atvaizdavimu kompiuteryje.
2. Susipažinome su programavimo sąvoka ir objektiniu programavimu.
3. Aptarėme programos kūrimo, kompiliavimo ir testavimo etapus.
4. Pradėjome pirmąjį pavyzdį, tačiau mums pritrūko žinių. Pratęsime kitoje temoje.

ktu informatikos fakultetas 1922

Klausimai?