

C#

Esitlus aines Programmeerimiskeeled

Autor: Mihkel Pajusalu

Sissejuhatus

- C# on objektorienteeritud keel, mis algupäraselt töötati välja Microsoftis.
 - Esimene ECMA spetsifikatsioon (ECMA-334) 2001
 - ECMA 2. spetsifikatsioon 2002 (ISO standard 2003)
 - ECMA 3. spetsifikatsioon 2005
- Eesmärgiks oli luua lihtne, modernne, üldotstarbeline objektorienteeritud programmeerimiskeel
- Ühendada C++ ja RAD keeled
- Osa .NET initsiatiivist

Keele nimi

- C# loetakse nagu ing. k. see-sharp
 - # tähistab algupäraselt dieesi (#) (ing. k. sharp), kuid tehniliste piirangute tõttu võeti kasutusele numbri märk

Keele eesmärgid

- Lihtsus
- Üldotstarbelisus
- Vigadevastased meetodid
 - Tüübikontroll
 - Massiivi piiride kontroll
 - Algväärtustamata muutujate kontroll
 - Automaatne garbage collector
- Koodi ja programmeerijate taaskasutus ja portaablus
- Ökonoomne ressursikasutus
 - Eesmärgiks polnud siiski C-ga konkureerimine

Mõjud

- Mõjud C#-le
 - C++
 - Delphi
 - Java
 - Modula-3
 - Eiffel
- Mõju teistele keeltele
 - Nemerle
 - D
 - Java

IDE-d ja kompilaatorid

- Microsoft Visual C#
 - Express versioon on tasuta allalaetav
 - Kompilaatorit kasutavad lisaks Borland C# builder ja SharpDevelop
- Mono C# kompilaator
- Microsofti projekt Rotor
- Dot GNU
- C#-is JavaScripti

Realisatsioon

- C# on ehitatud Common Language Infrastructure'ile (.NET)
 - Suurem osa C# tüüpidest vastab CLI tüüpidele
- C# kompilaatori väljund pole spetsifitseeritud
 - Kompilaator võib väljastada masinkoodi
 - Valdavalt suunatud Common Language Runtime'ile ja Common Intermediate Language'i väljastamisele (just-in-time kompileerimine)
- C# programmikoodi laiend on .cs

Üldine

- Pole globaalseid muutujaid ja funktsioone
 - Kõik peab olema klassides
 - Muutujate “varjutamine” keelatud
- Peaklassis Main funktsioon
- Klassid kuuluvad namespace'idesse
 - Neid saab kaasata using käsuga
 - Kaks namespace'i võivad komponente jagada
 - Üks klass võib olla defineeritud teise sees
- Ühene pärilus
 - Siiski võib liideseid olla mitu

Hello, World

```
using System;
```

```
class Hello
```

```
{
```

```
    static void Main() {
```

```
        Console.WriteLine("hello, world");
```

```
    }
```

```
}
```

- Kompileerimine: `csc hello.cs`
- Väljund: `hello, world`

Muutujatüübid

- Väärtus (value)
 - Lihtsamad muutujad (int, float, char)
 - Enumeratsioonid (enum)
 - Struktuurid (struct)
- Viit (reference)
 - Klassid (class)
 - Liidesed (interface)
 - Massiivid (array)
 - Delegaadid (delegate)
- Viit (pointer)
- boxing ja unboxing
- Unifitseeritud tüüpide süsteem

Muutujatüübid

- Lokaalsed muutujad
- Väljad (klassi muutujad)
 - staatilised
 - instantsieeritud
- Parameetrid
 - väärtusparameeter
 - viitparameeter
 - väljundparameeter
 - parameetrite massiiv

Eeldefineeritud andmetüübid

- object : Kõigi teiste tüüpide alud
- string : Unicode sümbolite järjend
- sbyte : 8-bitine märgiga täisarv
- short : 16-bitine märgiga täisarv
- int : 32-bitine märgiga täisarv
- long : 64-bitine märgiga täisarv
- byte: 8-bitine märgita täisarv
- ushort : 16-bitine märgita täisarv
- uint : 32-bitine märgita täisarv
- ulong: 64-bitine märgita täisarv

Eeldefineeritud andmetüübid

- float: ühekordse täpsusega ujukomaarv
- double: topelitäpsusega ujukomaarv
- bool: tõeväärtus, true või false
- char: üks Unicode'i sümbol
- decimal: täpne kümnendsüsteemis arv vähemalt 28 tüvenumbriga

Massiivid

- C# toetab nii ristikülikulisi kui ka “sakilisi” massiive
- Näide:

```
int[] a1 = new int[] {1, 2, 3};
```

```
int[,] a2 = new int[,] {{1, 2, 3}, {4, 5, 6}};
```

```
int[, ] a3 = new int[10, 20, 30];
```

```
int[][] j2 = new int[3][];
```

```
j2[0] = new int[] {1, 2, 3};
```

```
j2[1] = new int[] {1, 2, 3, 4, 5, 6};
```

```
j2[2] = new int[] {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Mäluhaldus

- C# kasutab üldiselt automaatset mäluhaldust
 - garbage collector
- Otsene ligipääs mälule kasutades “ebaturvalist” (unsafe) koodi
 - Objektid tuleb ajutiselt fikseerida
 - Saab kasutada C/C++ sarnaseid viitu (pointer)
 - Tuleb kompileerida /unsafe direktiiviga

Otsese mäluhalduse näide

```
using System;
class Test
{
    static void writeLocations(byte[] arr) {
        unsafe {
            fixed (byte* pArray = arr) {
                byte* pElem = pArray;
                for (int i = 0; i < arr.Length; i++) {
                    byte value = *pElem;
                    Console.WriteLine("arr[{0}] at 0x{1:X}
                        is {2}", i, (uint)pElem, value);
                    pElem++;
                }
            }
        }
    }
    static void Main() {
        byte[] arr = new byte[] {1, 2, 3, 4, 5};
        writeLocations(arr);
    }
}
```


Operaatorid

- Primaarsed: `x.y f(x) a[x] x++ x-- new typeof checked unchecked`
- Unaarsed: `+ - ! ~ ++x --x (T)x`
- Multiplikatiivsed: `* / %`
- Additiivsed: `+ -`
- Nihked: `<< >>`
- Võrdlus ja tüübikontroll: `< > <= >= is as`
- Võrdsus: `== !=`
- Loogiline AND: `&`
- Loogiline XOR: `^`
- Loogiline OR: `|`
- Tingimuslik AND: `&&`
- Tingimuslik OR: `||`
- Tingimus: `?:`
- Omistamine: `= *= /= %= += -= ' <<= >>= &= ^= |=`

Keele elemendid

- Märgendid ja goto'd
- if
- switch
- while
- do..while
- for
- foreach
- break
- continue
- return

Keele elemendid

- throw ja try
- checked ja unchecked
- lock
- using
- yield

Lisavõimalused

- C# võimaldab operaatorite ülelaadimist
- C# võimaldab kasutaja poolt defineeritud teisendusi (cast)
- Preprotsessor
- unsafe
- Teistes keeltes kirjutatud programmide esilekutsumine
- Erindid

Preprotsessori direktiivid

- #if
- #else
- #elif
- #endif
- #define
- #undef
- #warning
- #error
- #line
- #region
- #endregion

Preprotsessori näide

```
#define DEBUG
#define VC_V6
using System;
public class MyClass
{
    public static void Main()
    {
#if (DEBUG && !VC_V6)
        Console.WriteLine("DEBUG is defined");
#elif (!DEBUG && VC_V6)
        Console.WriteLine("VC_V6 is defined");
#elif (DEBUG && VC_V6)
        Console.WriteLine("DEBUG and VC_V6 are defined");
#else
        Console.WriteLine("DEBUG and VC_V6 are not
            defined");
#endif
    }
}
```

Lisavõimalused 2.0

- Osaklassid (partial classes)
- Geneerikud (generics)
- Uus iteraatoritüüp
 - yield return
- Anonüümsed delegaadid

yield return näide

// Method that takes an iterable input (possibly an array) and returns all even numbers.

```
public static IEnumerable<int> GetEven(IEnumerable<int>
    numbers) {
    foreach (int i in numbers){
        if (i % 2 == 0) yield return i;
    }
}
```


Anonüümsed delegaadid

```
public void Foo(object parameter) {  
    // ...  
  
    ThreadPool.QueueUserWorkItem(delegate  
    {  
        // anonymous delegates have full access to local  
        variables of the enclosing method  
        if (parameter == ...)  
        {  
            // ...  
        }  
  
        // ...  
    });  
}
```

Standardklassid

- C# kasutab .NET klassiteeki (Class Library)
 - Sisaldab erinevaid klasse erinevateks funktsioonideks
 - Üldelhituselt sarnane Java klasside teegiga

Viited

- <http://msdn2.microsoft.com/en-us/vcsharp/aa336809.aspx>
- <http://www.ecma-international.org/publications/files/ecma-st/ECMA-334.pdf>
- http://en.wikipedia.org/wiki/C_Sharp
- <http://www.c-sharpcorner.com/UploadFile/vgupta/PreprocessorDirectivesCS1>