

# .NET ja C#

## Virtuaalikone

### *Periaate*

- ◆ Virtuaalikone on tapa abstrahoida allaoleva toteutus yhteisen rajapinnan taakse.
- ◆ Virtuaalikoneet muodostavat hierarkian.
- ◆ Hierarkiassa ylempänä oleva käyttää alemman tarjoamaa rajapintaa ja tarjoaa omat palvelunsa ylöspäin.

### *Etuja*

- ◆ Vaikka allaoleva toteutus (esim. laitteisto) muuttuu radikaalisti, hierarkian ylemmät osat (esim. ohjelmat) voivat toimia välissä olevien (esim. käyttöjärjestelmän) kerrosten päällä muutumattomina.
- ◆ Abstraktion vahvistuminen hierarkiassa noustessa helpottaa työtä (esim. sovelluskehitystä)

### *Haittoja*

- ◆ Pääsyä alemman virtuaalikoneen ohi ei välttämättä ole tarjolla.
- ◆ Jokainen kerros kuluttaa tehoa.
- ◆ Jos jokin kerros on tehoton, se heijastuu suoraan kaikkiin ylempiin kerroksiin.

## Common Language Infrastructure (CLI)

### *Mikä on CLI*

- ◆ Microsoftin avoin määrittely, joka kuvaa ajettavan koodin ja ajoympäristön.
- ◆ Muodostaa .NET Frameworkin ytimen.
- ◆ Sallii useiden eri korkean tason ohjelmointikielien käytön eri ympäristöissä.
- ◆ Virtuaalikonehierarkiassa käyttöjärjestelmän ja sovelluksen välissä (kuten Java VM).

### *CLI standardin merkitys (CLS, Ecma)*

- ◆ CLI on Microsoftin kehittämä ja Ecman standardoima (Ecma-335) spesifikaatio.
- ◆ Ecman standardi mahdollistaa avoimen ohjelmistokehityksen CLI:n pohjalta.
- ◆ Microsoftin alkuperäinen ajatusmalli oli luoda käyttöjärjestelmäriippumaton standardi.
- ◆ Tällä hetkellä merkittävin CLI:n toteuttava implementaatio on Microsoftin .NET Framework.
- ◆ .NET tukee vain Microsoft Windows –ympäristöä.

- ◆ Muita CLI:n mukaisia toteutuksia ovat mm. MONO ja, Portable.NET, jotka toimivat mm. Windows, NetBSD, FreeBSD, Solaris, BeOS, Linux ja Mac OS X –ympäristöissä.

## **.NET ja CLI**

### ***Välikieli (CIL)***

- ◆ CLI määrittelee .NET-kääntäjien generoiman välikielen.
- ◆ Ajonaikainen suoritusympäristö kääntää välikielen konekielelle.

### ***Common Language Specification (CLS)***

- ◆ Määrittelee rakenteet, joita CLS-yhteensopivissa ohjelmointikielissä on mahdollista olla
- ◆ Metodit, rajapinnat, taulukot, kentät, enumeraatiot, tapahtumat, arvotyypit, luokat, alustimet, poikkeukset, näkyvyys

### ***Common Type System (CTS)***

- ◆ Määrittää kielestä riippumattoman muuttujatyyppirakenteen eli virtuaalisen suoritusympäristön sisäiset tyypit, jotka kielen on toteutettava ollakseen CLS-yhteensopiva
- ◆ Metadata
- ◆ Sisältää kielestä riippumatonta tietoa ohjelmasta ja sen suorituksesta
- ◆ Voidaan hyödyntää sekä kielissä että työkaluissa
- ◆ Esim. versiotiedot
- ◆ Voidaan laajentaa ohjelmoinnillisesti

### ***Virtual Execution System (VES)***

- ◆ Lataa ja suorittaa CLI-yhteensopivia ohjelmia
- ◆ Yhdistää koodisegmenttejä metadatan avulla
- ◆ Kääntää välikielen konekielelle

### ***Garbage Collection (GC)***

- ◆ Huolehtii muistin siivouksesta
- ◆ Pitää yllä tietoa olioviittauksista

## **.NET ohjelmoijalle**

### ***Historiaa***

- ◆ Delphin kehittäjä Anders Hejlsberg palkattiin vuonna 1996 Borlandilta Microsoftille kehittämään MS:n Java-virtuaalikonetta (JVM) ja Visual Studion J++:aa.
- ◆ Sun ja Microsoft ajautuivat erimielisyyksiin siitä, kuinka paljon MS saa laajentaa JVM:ää.
- ◆ Microsoftilla kehitettiin J++:n ohessa myös omaa kieltä, jonka projektinimi oli Cool

- ◆ Ilmeisesti Java-kielen kehittämisen mahdottomuus ajoi MS:n julkaisemaan Coolin, josta tuli C#
- ◆ Julkaisunsa jälkeen vuonna 2000 C# ja CLI annettiin Ecmalle (European Computer Manufacturers Association) standardoitavaksi ja se standardoitiin vuonna 2002

## **.NET tänään**

- ◆ Nykypäivänä lukuisista kielistä on olemassa .NET toteutus
- ◆ Sisäänrakennettuina kielinä
  - ◆ C#
  - ◆ JScript .NET
  - ◆ J#
  - ◆ C++ .NET (Managed C++)
  - ◆ Visual Basic .NET
- ◆ Lisäksi kolmannen osapuolen toteuttamia .NET yhteensopivia kieliä on useita, esimerkiksi:
 

◆ A# (Ada)	◆ IronPython (Python)
◆ APL	◆ Lexico
◆ Boo, based on Python	◆ Lisp
◆ COBOL	◆ Mercury
◆ Component Pascal	◆ Mondrian
◆ Delphi 8 ja Delphi 2005	◆ Nemerle
◆ Eiffel	◆ Oberon
◆ F#	◆ Perl
◆ Forth	◆ RPG
◆ FORTRAN	◆ Smalltalk
◆ IKVM (Java)	
- ◆ Lisäksi .NETiin kuuluvat ASP.NET, joka on .NETiin perustuva kehittyneempi versio Active Server Pages -teknologiasta sekä ADO.NET, joka on .NETin toteutus Microsoftin ADO (ActiveX Data Objects) -teknologiasta

## **Ohjelmistokehitys .NETillä**

- ◆ Useiden kielten toimiminen yhteen helpottaa ohjelmistokehitystä
- ◆ Eri kielten osaajat voivat toimia suoraan samassa ympäristössä
- ◆ Voit periä C#-luokan C++:ssa, instantioida sen Eiffelissä ja kutsua sen metodia COBOLissa
- ◆ Kieli ja sen .NET-versio saattavat erota toisistaan huomattavasti

- ◆ Kielten suorituskyky ei myöskään välttämättä vastaa alkuperäistä
- ◆ C++.NET
  - ◆ Tuo C++:aan .NET:in vaatimat ominaisuudet
  - ◆ Pystyy sekoittamaan hallittua ja hallitsematonta koodia
  - ◆ Nopein .NETin sisäänrakennetuista kielistä

## **Hallittu koodi**

- ◆ CLI-standardin mukaista koodia, jonka suoritusta hallinnoidaan erilaisin turvallisuusmenettelyin, esimerkiksi
  - ◆ Rajoittamalla suoritusoikeuksia
  - ◆ Tarkastamalla taulukoiden indeksit ajonaikaisesti, mikä estää puskuriylikuodot
  - ◆ Ilmoittamalla ylikuodoista aritmeettisissa operaatioissa
- ◆ Hallitusta koodista ei ole mahdollista päästä tekemään suoria muistiosoitteita eikä käsittelemään prosessorin rekistereitä. Hallitussa koodissa ei ole myöskään perinteisiä funktio-osoittimia
- ◆ Ajoympäristön suorittamat turvallisuustarkastukset sekä oikeuksien hallinta toimii ainoastaan silloin kun on kyse hallitusta koodista
- ◆ CLI-standardi mahdollistaa hallitusta koodista tehtävät kutsut hallitsemattomaan koodiin eli perinteiseen käyttöjärjestelmässä ajettavaan ohjelmakoodiin
- ◆ Helpottaa ohjelmointia ja ylläpitoa

## **C# kielenä**

### **C# vs Java vs C++**

- ◆ Keskimäärin C# on suorituskyvyltään C++:n ja Javan välillä
- ◆ C#:ssa on enemmän ominaisuuksia kuin Javassa
- ◆ Syntaktisesti enemmän Javan kuin C++:n kaltainen
- ◆ C# on näistä kolmesta nopeiten kehittyvä

### **C# 2.0**

- ◆ Julkaistiin 7.11.2005
- ◆ Uusia ominaisuuksia esimerkiksi
  - ◆ Generics
  - ◆ Iteraattorit
  - ◆ Osittaiset luokat
  - ◆ Nimettömät metodit
  - ◆ Nollattavat tyypit
  - ◆ Closures

## **C# 3.0**

- ◆ Lambda expressions
- ◆ Kieleen rakennetut tietokantakyselyt
- ◆ Laajennusmetodit