Αντικειμενοστρεφής Ανάπτυξη Εφαρμογών

Δρ. Ευθύμιος Αλέπης

Βασικές έννοιες

- Τάξεις και Αντικείμενα
- Κληρονομικότητα
- Πολυμορφισμός
- Ενθυλάκωση
- Υπερφόρτωση
- Αφαιρετικότητα

Κληρονομικότητα

«Η κληρονομικότητα είναι μια από τις σημαντικότερες ιδιότητες του αντικειμενοστρεφούς προγραμματισμού. Σύμφωνα με αυτή, δίνεται η δυνατότητα σε τάξεις να κληρονομούν χαρακτηριστικά και δυνατότητες από άλλες τάξεις. Η κληρονομικότητα καθιστά τον κώδικα επαναχρησιμοποιήσιμο σε μεγάλο βαθμό»

Πολυμορφισμός

«Με τον όρο “πολυμορφισμός” αναφερόμαστε στη δυνατότητα των συναρτήσεων των αντικειμενοστρεφών προγραμμάτων να μπορούν να έχουν διαφορετικές υλοποιήσεις. Έτσι, η ίδια συνάρτηση μπορεί να χρησιμοποιηθεί για πολλούς, διαφορετικούς σκοπούς. Αρκετές φορές η ίδια συνάρτηση μπορεί να οριστεί με διαφορετικά ορίσματα»

Υπερφόρτωση

«Η υπερφόρτωση αποτελεί υποσύνολο του πολυμορφισμού. Κατά την υπερφόρτωση μιας συνάρτησης ή μεθόδου, αυτή λειτουργεί χρησιμοποιώντας διαφορετικό/διαφορετικούς τύπο/τύπους δεδομένων»

Ενθυλάκωση

«Ενθυλάκωση είναι η δυνατότητα που παρέχει ο αντικειμενοστρεφής προγραμματισμός, σύμφωνα με την οποία τόσο τα δεδομένα, όσο και οι συναρτήσεις που αφορούν αυτά τα δεδομένα, να βρίσκονται στο ίδιο τμήμα κώδικα. Κατά την ενθυλάκωση μπορεί να οριστεί και το επίπεδο πρόσβασης στα υπάρχοντα δεδομένα»

Αφαιρετικότητα

«Η αφαιρετικότητα αναφέρεται στη δυνατότητα της αντικειμενοστρεφούς σχεδίασης να παρέχει στον εξωτερικό κόσμο μόνο τις απαραίτητες και σημαντικές πληροφορίες και να αποκρύπτει λεπτομέρειες που δεν ενδιαφέρουν πραγματικά τον εξωτερικό κόσμο»

Αντικειμενοστρεφής Σχεδιασμός

- Εντοπισμός των αντικειμένων που βρίσκονται στο συγκεκριμένο επίπεδο ανάλυσης
- Εντοπισμός της αναμενόμενης συμπεριφοράς των αντικειμένων
- Εντοπισμός των σχέσεων ανάμεσα στα αντικείμενα
- Υλοποίηση των αντικειμένων

Αντικειμενοστρεφής προγραμματισμός σε C++

C++ Class

```
class Box
{
   public:
      double length;    // Length of a box
      double breadth;   // Breadth of a box
      double height;    // Height of a box
};
```

C++ Object

```
Box Box1;        // Declare Box1 of type Box
Box Box2;        // Declare Box2 of type Box
```

C++ Object use

```
int main( )
{
   Box Box1;
   Box Box2; double volume = 0.0;
// box 1 specification
   Box1.height = 5.0;
   Box1.length = 6.0;
   Box1.breadth = 7.0;
// box 2 specification
   Box2.height = 10.0;
   Box2.length = 12.0;
   Box2.breadth = 13.0;
// volume of box 1
   volume = Box1.height * Box1.length * Box1.breadth;
   cout << "Volume of Box1 : " << volume <<endl;
   // volume of box 2
   volume = Box2.height * Box2.length * Box2.breadth;
   cout << "Volume of Box2 : " << volume <<endl;
   return 0;
}
```

C++ Inheritance

```
// Base class
class Shape
{
   public:
      void setWidth(int w)
      {
         width = w;
      }
      void setHeight(int h)
      {
         height = h;
      }
   protected:
      int width;
      int height;
};
```

```
// Derived class
class Rectangle: public Shape
{
   public:
      int getArea()
      {
         return (width * height);
      }
};
```

```
int main(void)
{
   Rectangle Rect;

   Rect.setWidth(5);
   Rect.setHeight(7);

   // Print the area of the object.
   cout << "Total area: " << Rect.getArea() << endl;

   return 0;
}
```

C++ Access control

Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

C++ polymorphism (overloading)

```
class printData
{
   public:
      void print(int i) {
        cout << "Printing int: " << i << endl;
      }

      void print(double  f) {
        cout << "Printing float: " << f << endl;
      }

      void print(char* c) {
        cout << "Printing character: " << c << endl;
      }
};
```

Αντικειμενοστρεφής προγραμματισμός σε PHP

PHP Class

```
<?php
class phpClass{
   var $var1;
   var $var2 = “some string";
   function myfunction ($arg1, $arg2) {
      [code]
   }
   [code for other functions]
}
?>
```

PHP Book Class example

```
<?php
class  Books{
    var $price;
    var $title;
    function setPrice($par){
       $this->price = $par;
    }
    function getPrice(){
       echo $this->price ."<br/>";
    }
    function setTitle($par){
       $this->title = $par;
    }
    function getTitle(){
       echo $this->title ." <br/>";
    }
}
?>
```

PHP Objects

```
$physics = new Books;
$maths = new Books;
$chemistry = new Books;
```

PHP Object Use

```
$physics->setTitle( "Physics for Elementary School" );
$chemistry->setTitle( “Basic Chemistry" );
$maths->setTitle( “Calculus" );

$physics->setPrice( 100 );
$chemistry->setPrice( 150 );
$maths->setPrice( 70 );
```

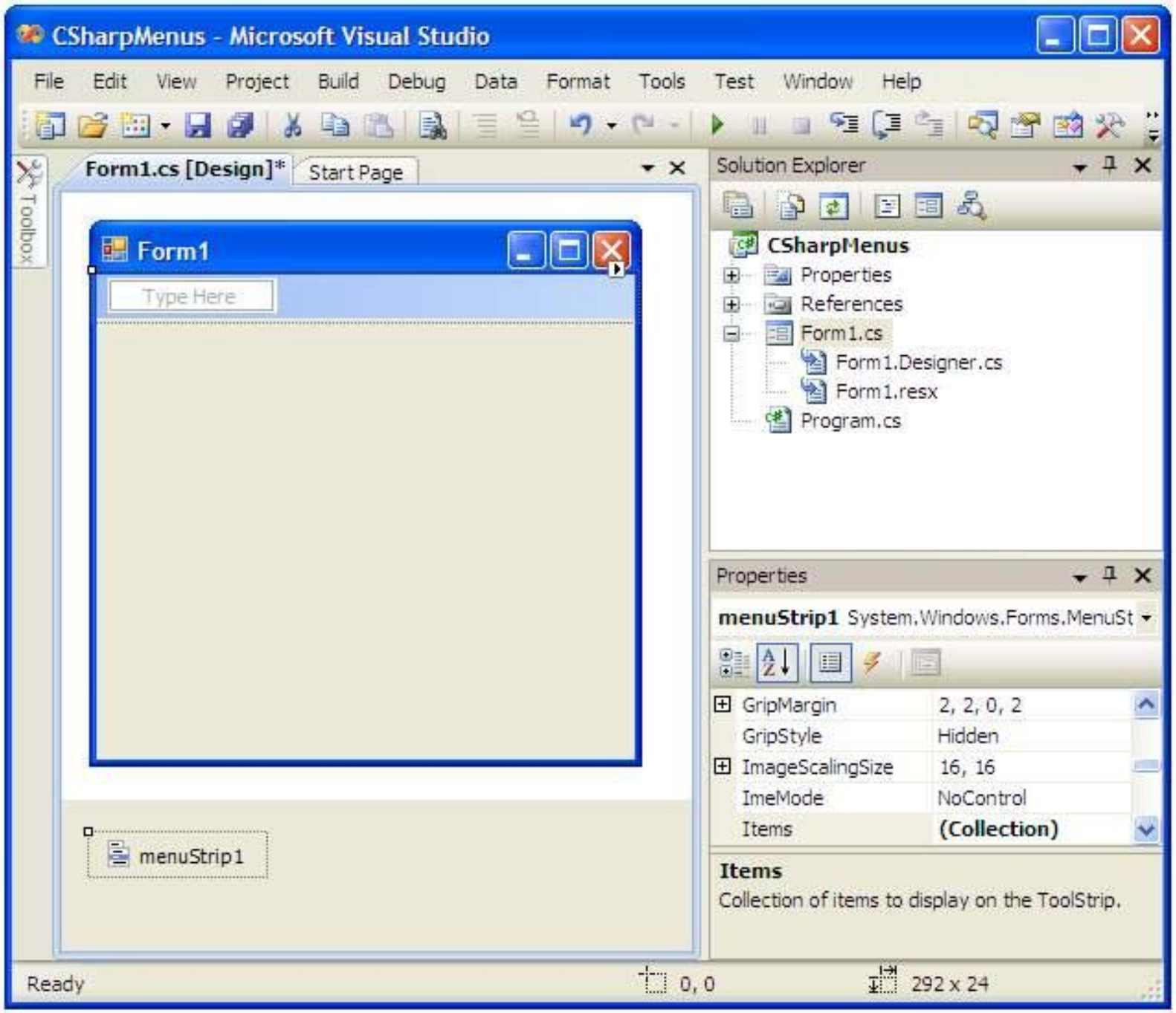
Αντικειμενοστρεφής προγραμματισμός σε C#

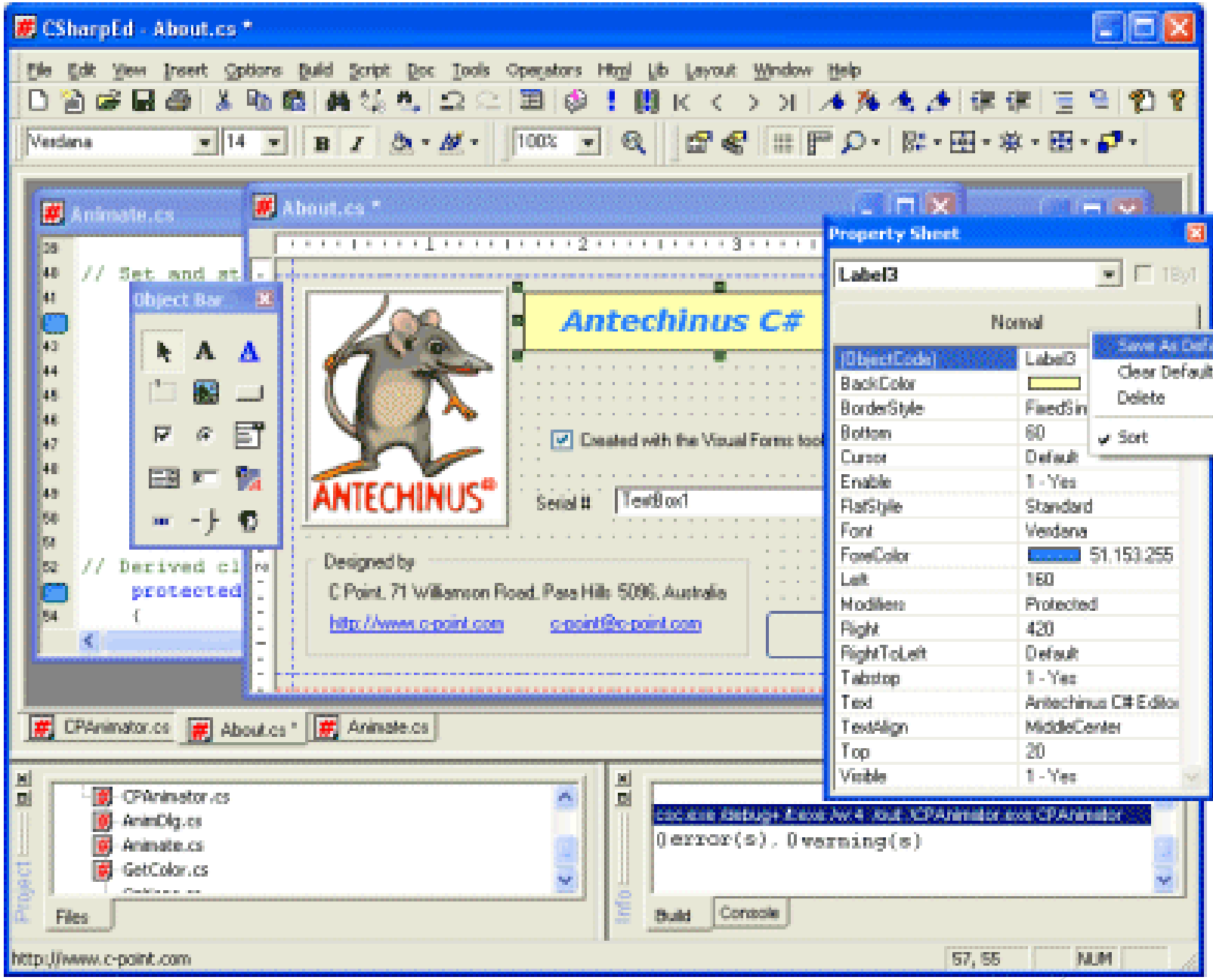
C# General Class Definition

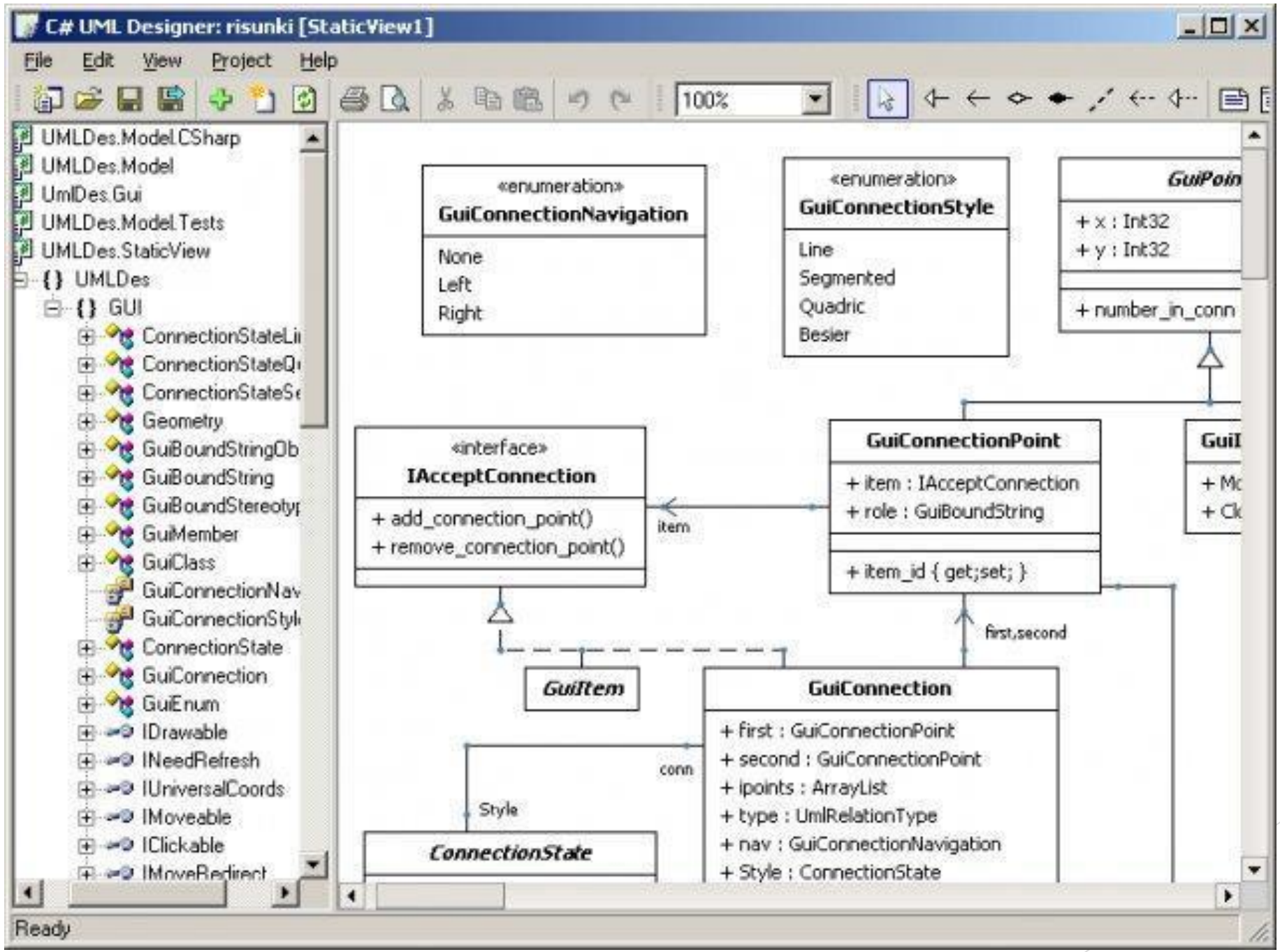
```
<access specifier> class  class_name
{
   // member variables
   <access specifier> <data type> variable1;
   <access specifier> <data type> variable2;
   ...
   <access specifier> <data type> variableN;
   // member methods
   <access specifier> <return type> method1(parameter_list)
   {
      // method1 body
   }
   <access specifier> <return type> method2(parameter_list)
   {
      // method2 body
   }
   ...
   <access specifier> <return type> methodN(parameter_list)
   {
      // method3 body
   }
}
```

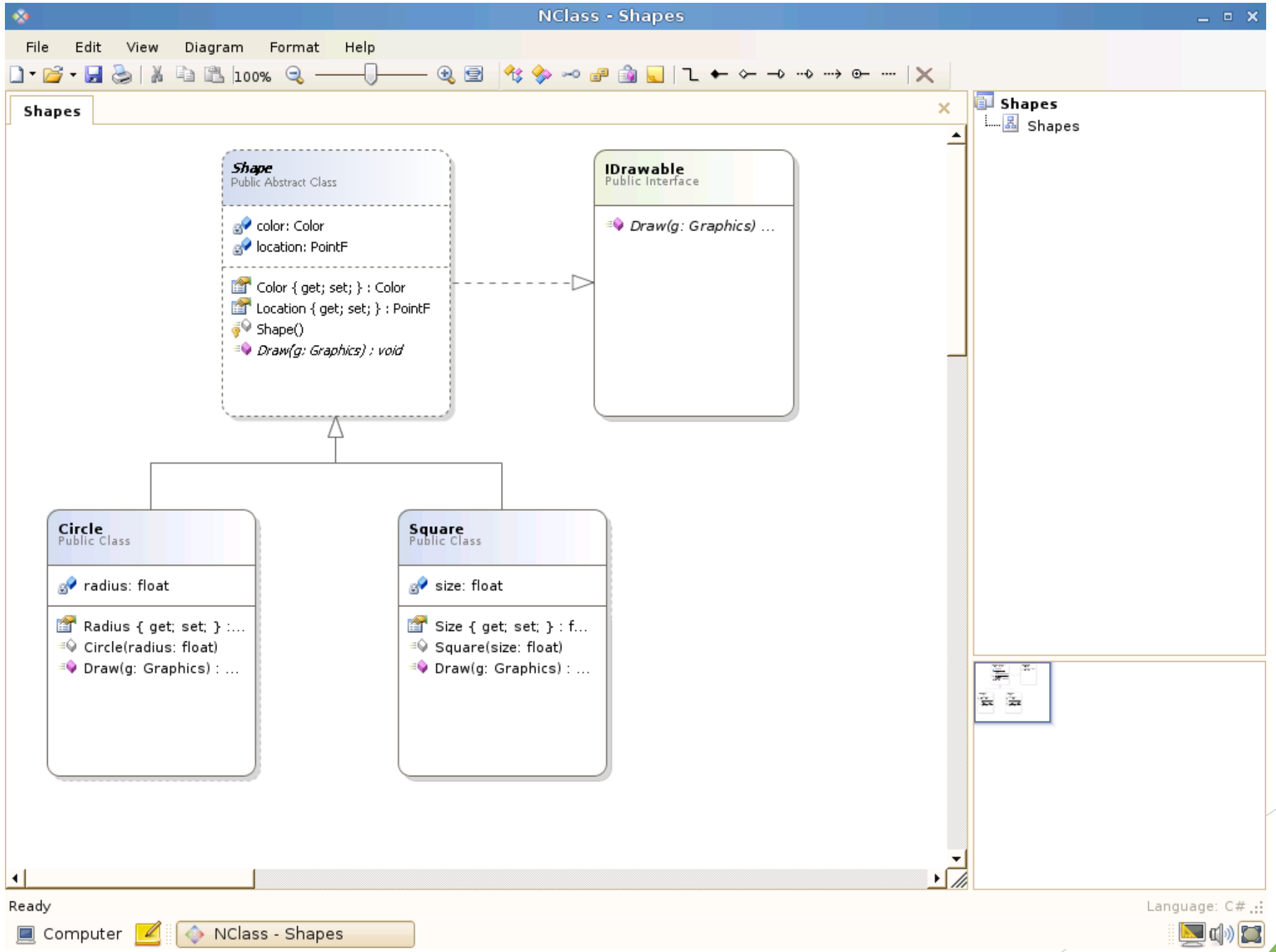
C# example

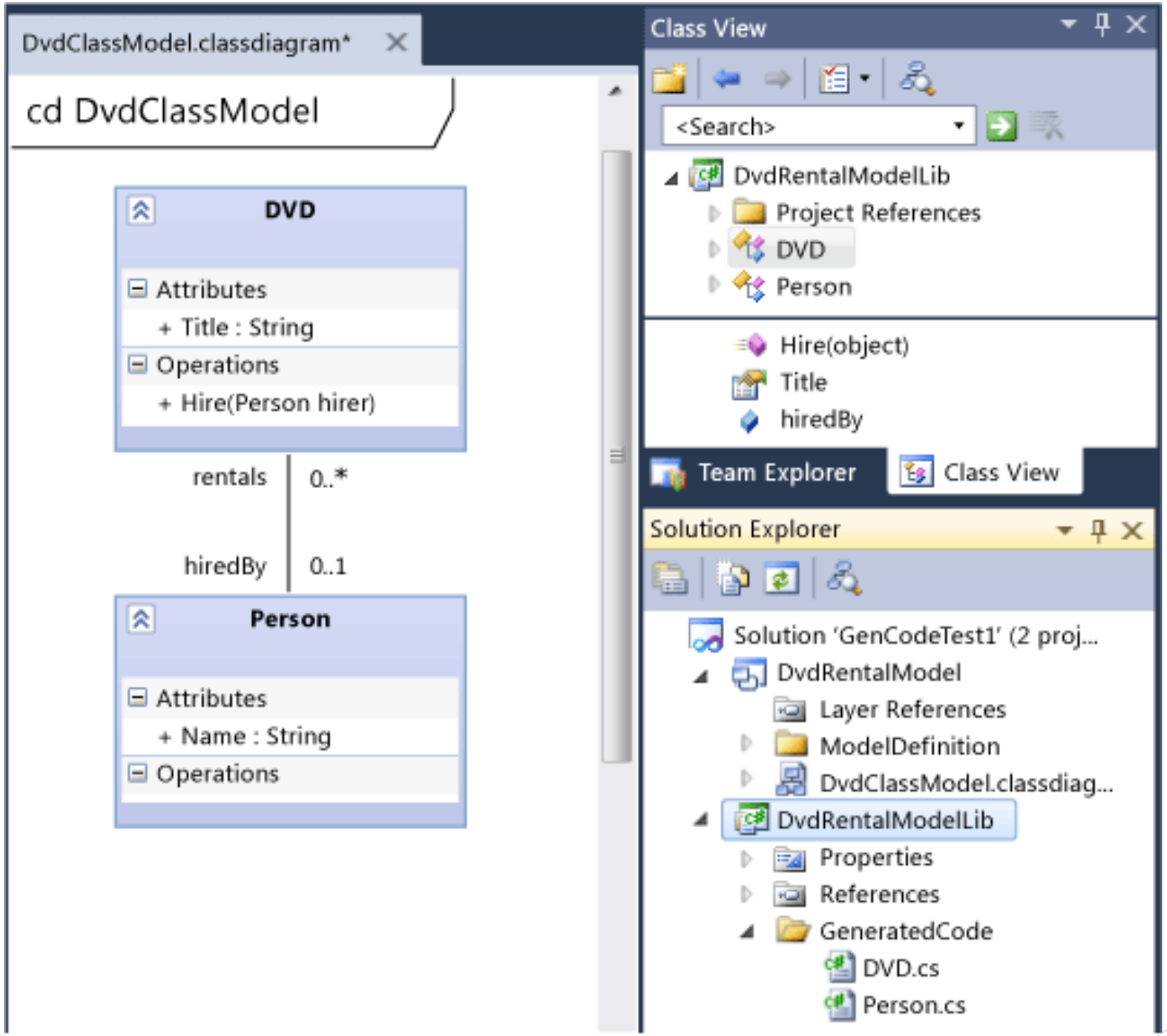
```
using System;
namespace BoxApplication
{
    class Box
    {
       public double length;   // Length of a box
       public double breadth;  // Breadth of a box
       public double height;   // Height of a box
    }
    class Boxtester
    {
        static void Main(string[] args)
        {
            Box Box1 = new Box();        // Declare Box1 of type Box
            double volume = 0.0;         // Store the volume of a box here
            // box 1 specification
            Box1.height = 5.0;
            Box1.length = 6.0;
            Box1.breadth = 7.0;
            // volume of box 1
            volume = Box1.height * Box1.length * Box1.breadth;
            Console.WriteLine("Volume of Box1 : {0}",  volume);
        }
    }
}
```

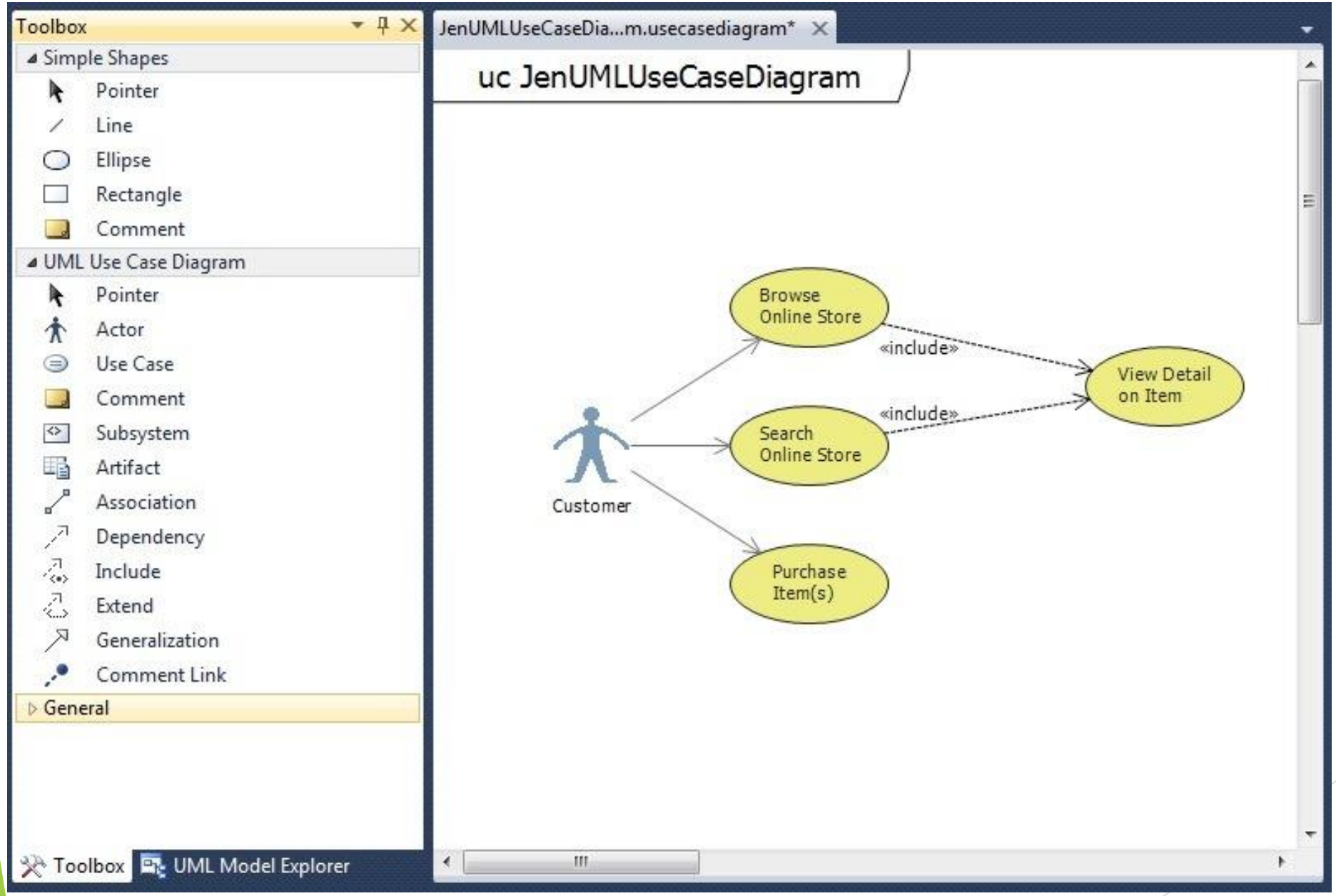

CSharpMenus - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Toolbox
Form1.cs [Design]*
Start Page
Form1
Type Here
menuStrip1
Solution Explorer
CSharpMenus
Properties
References
Form1.cs
Form1.Designer.cs
Form1.resx
Program.cs
Properties
menuStrip1 System.Windows.Forms.MenuSt
GripMargin 2, 2, 0, 2
GripStyle Hidden
ImageScalingSize 16, 16
ImeMode NoControl
Items (Collection)
Items
Collection of items to display on the ToolStrip.
Ready
0, 0
292 x 24

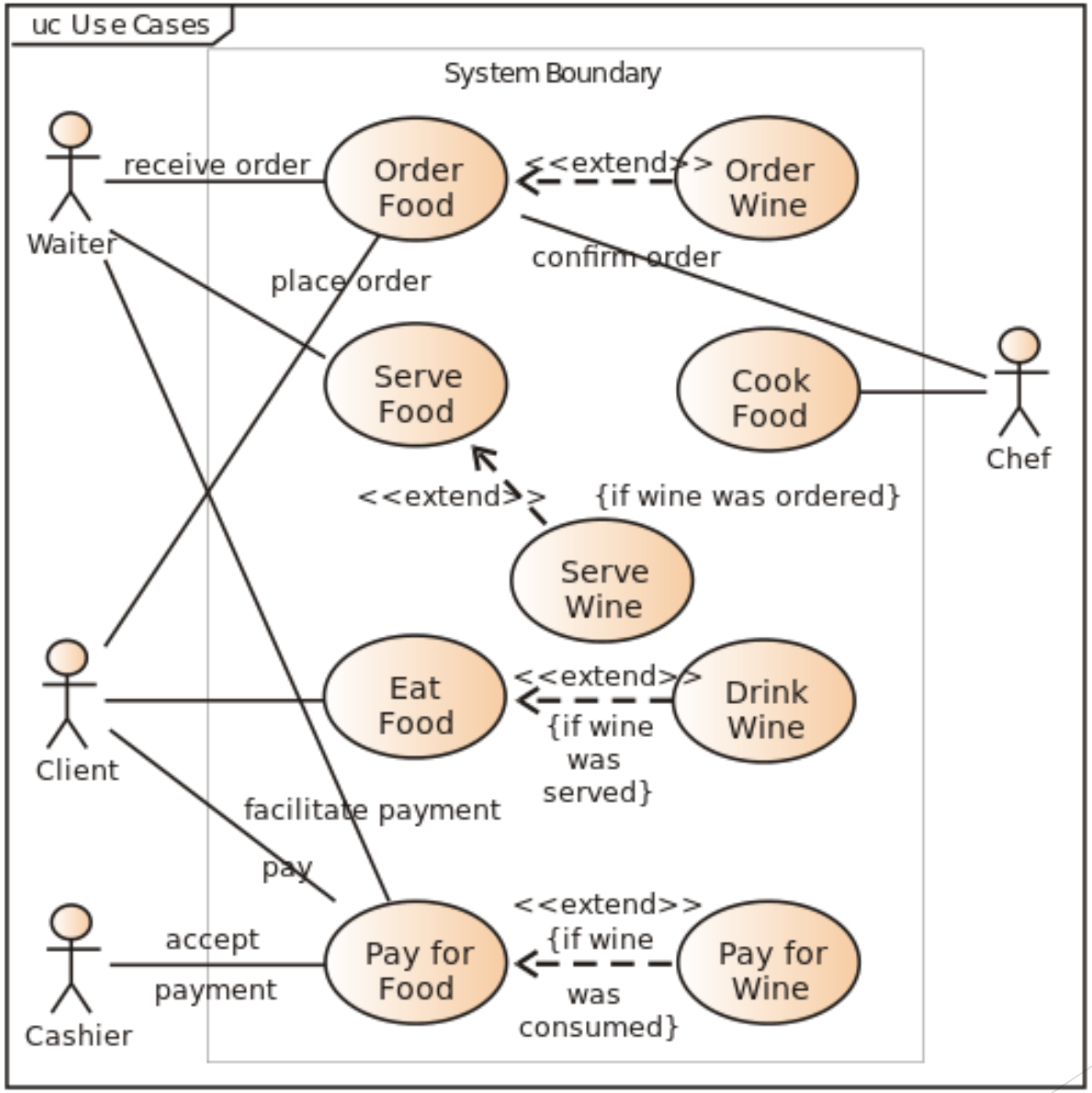

CSharpEd - About.cs *
File Edit View Insert Options Build Script Doc Tools Operators Html Lib Layout Window Help
Animate.cs
About.cs *
Object Bar
Antechinus C#
ANTECHINUS
Created with the Visual Forms tool
Serial #
TextBox1
Designed by
C Point, 71 Williamson Road, Pass Hills 5096, Australia
http://www.c-point.com
c-point@c-point.com
Property Sheet
Label3
Normal
(ObjectCode) Label3
BackColor
BorderStyle
Bottom 60
Cursor Default
Enable 1 - Yes
FlatStyle Standard
Font Verdana
ForeColor 51, 153, 255
Left 160
Modifiers Protected
Right 420
RightToLeft Default
Tabstop 1 - Yes
Text Antechinus C# Editor
TextAlign MiddleCenter
Top 20
Visible 1 - Yes
Save As Default
Clear Default
Delete
Sort
CPAnimator.cs
About.cs *
Animate.cs
CPAnimator.cs
AnimDlg.cs
Animate.cs
GetColor.cs
Files
Project
0 error(s), 0 warning(s)
Build
Console
Info
http://www.c-point.com
57, 55
NUM

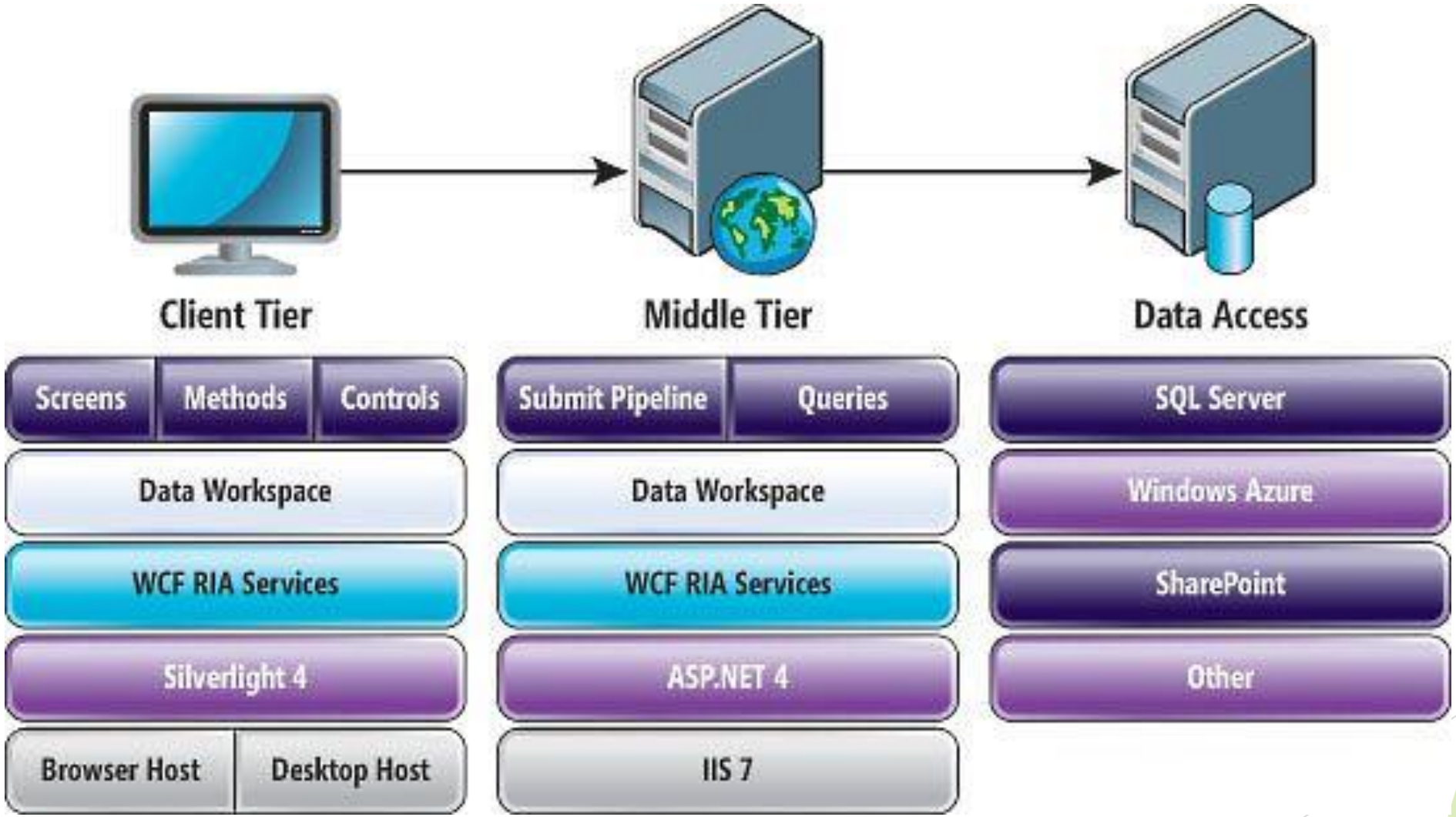

C# UML Designer: risunki [StaticView1]
File
Edit
View
Project
Help
100%
UMLDes.ModelCSharp
UMLDes.Model
UmlDes.Gui
UMLDes.ModelTests
UMLDes.StaticView
UMLDes
GUI
ConnectionStateLi
ConnectionStateQ
ConnectionStateSe
Geometry
GuiBoundStringOb
GuiBoundString
GuiBoundStereotyp
GuiMember
GuiClass
GuiConnectionNav
GuiConnectionStyl
ConnectionState
GuiConnection
GuiEnum
IDrawable
INeedRefresh
IUniversalCoords
IMoveable
IClickable
IMoveRedirect
«enumeration»
GuiConnectionNavigation
None
Left
Right
«enumeration»
GuiConnectionStyle
Line
Segmented
Quadric
Besier
GuiPoin
+ x : Int32
+ y : Int32
+ number_in_conn
«interface»
IAcceptConnection
+ add_connection_point()
+ remove_connection_point()
item
GuiConnectionPoint
+ item : IAcceptConnection
+ role : GuiBoundString
+ item_id { get;set; }
GuiI
+ Mc
+ Cl
first,second
GuiItem
GuiConnection
+ first : GuiConnectionPoint
+ second : GuiConnectionPoint
+ ipoints : ArrayList
+ type : UmlRelationType
+ nav : GuiConnectionNavigation
+ Style : ConnectionState
conn
Style
ConnectionState
Ready

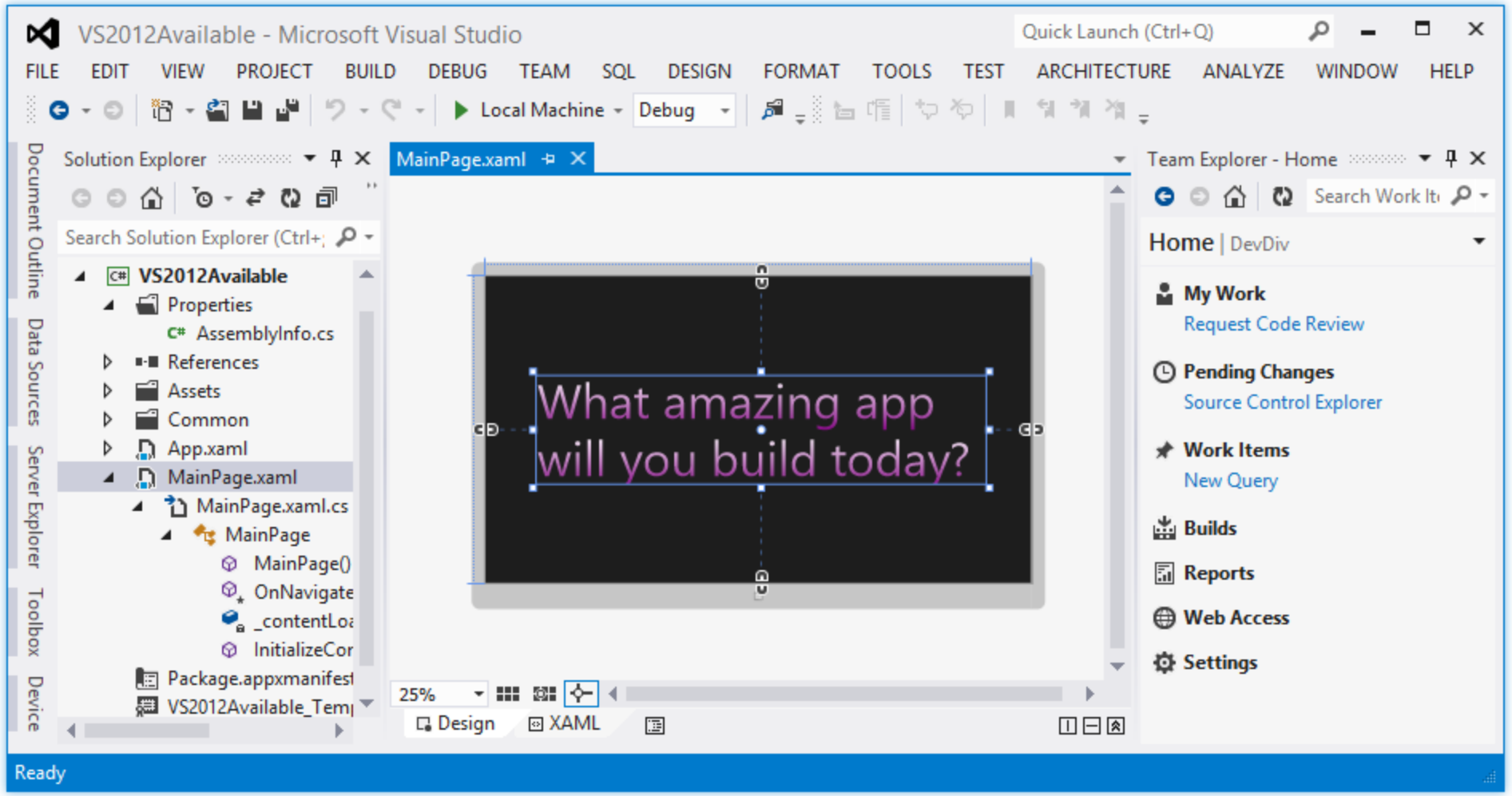

NClass - Shapes
File
Edit
View
Diagram
Format
Help
100%
Shapes
Shape
Public Abstract Class
color: Color
location: PointF
Color { get; set; } : Color
Location { get; set; } : PointF
Shape()
Draw(g: Graphics) : void
IDrawable
Public Interface
Draw(g: Graphics) ...
Circle
Public Class
radius: float
Radius { get; set; } :...
Circle(radius: float)
Draw(g: Graphics) : ...
Square
Public Class
size: float
Size { get; set; } : f...
Square(size: float)
Draw(g: Graphics) : ...
Shapes
Shapes
Ready
Language: C#
Computer
NClass - Shapes

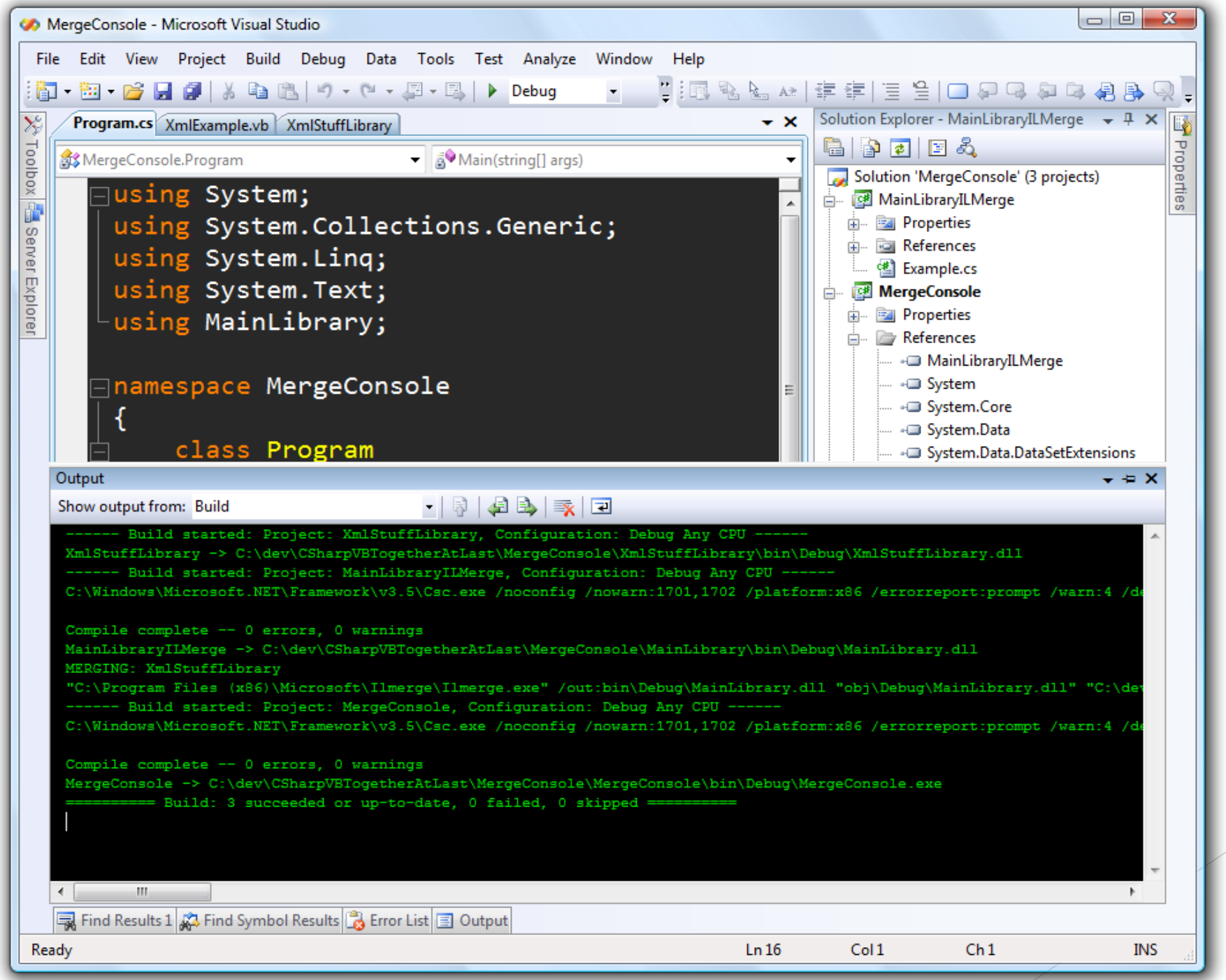

DvdClassModel.classdiagram*
cd DvdClassModel
DVD
Attributes
+ Title : String
Operations
+ Hire(Person hirer)
rentals
0..*
hiredBy
0..1
Person
Attributes
+ Name : String
Operations
Class View
<Search>
DvdRentalModelLib
Project References
DVD
Person
Hire(object)
Title
hiredBy
Team Explorer
Class View
Solution Explorer
Solution 'GenCodeTest1' (2 proj...
DvdRentalModel
Layer References
ModelDefinition
DvdClassModel.classdiag...
DvdRentalModelLib
Properties
References
GeneratedCode
DVD.cs
Person.cs

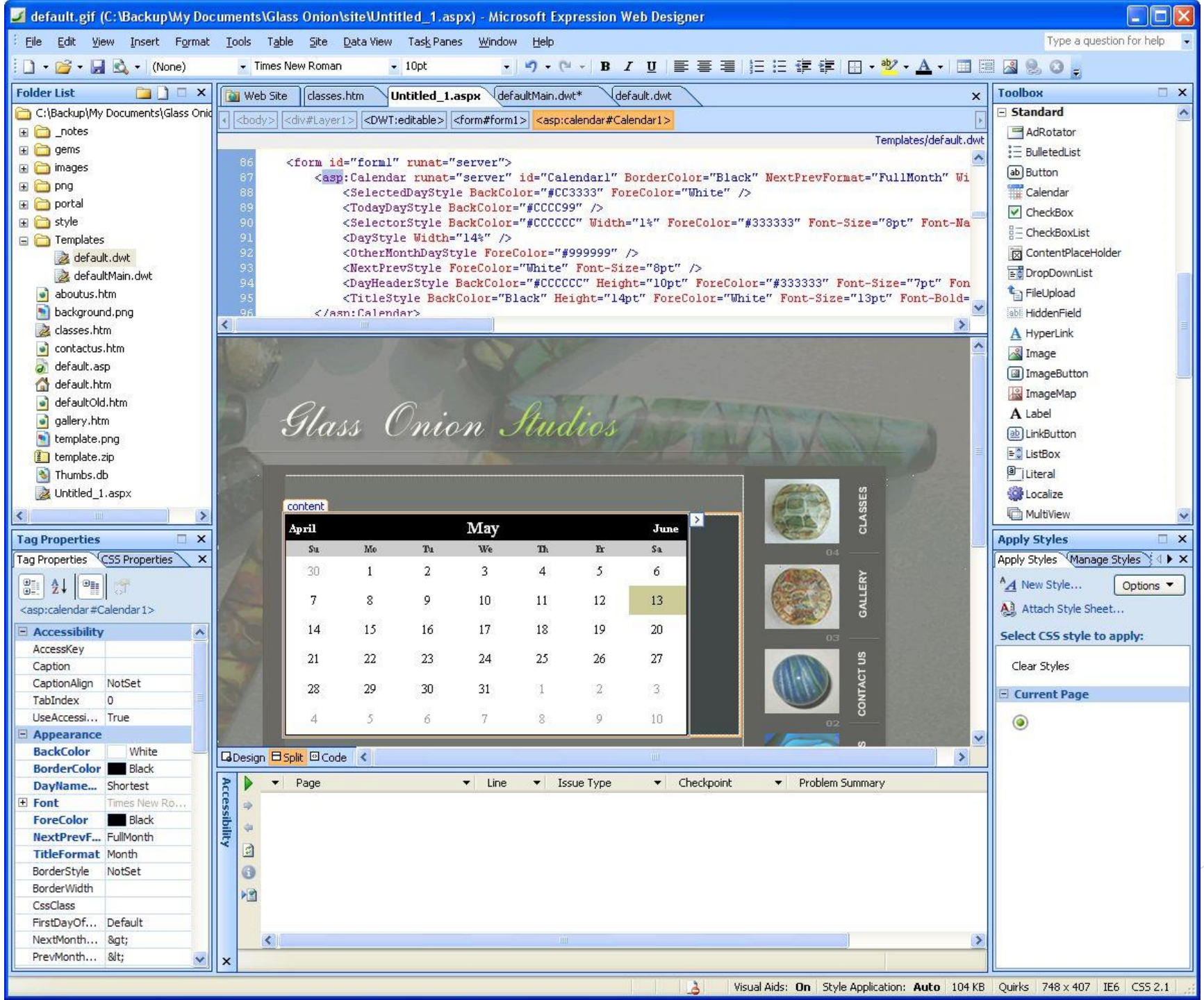

Toolbox
Simple Shapes
Pointer
Line
Ellipse
Rectangle
Comment
UML Use Case Diagram
Pointer
Actor
Use Case
Comment
Subsystem
Artifact
Association
Dependency
Include
Extend
Generalization
Comment Link
General
Toolbox
UML Model Explorer
JenUMLUseCaseDia...m.usecasediagram*
uc JenUMLUseCaseDiagram
Browse Online Store
«include»
View Detail on Item
Search Online Store
«include»
Customer
Purchase Item(s)

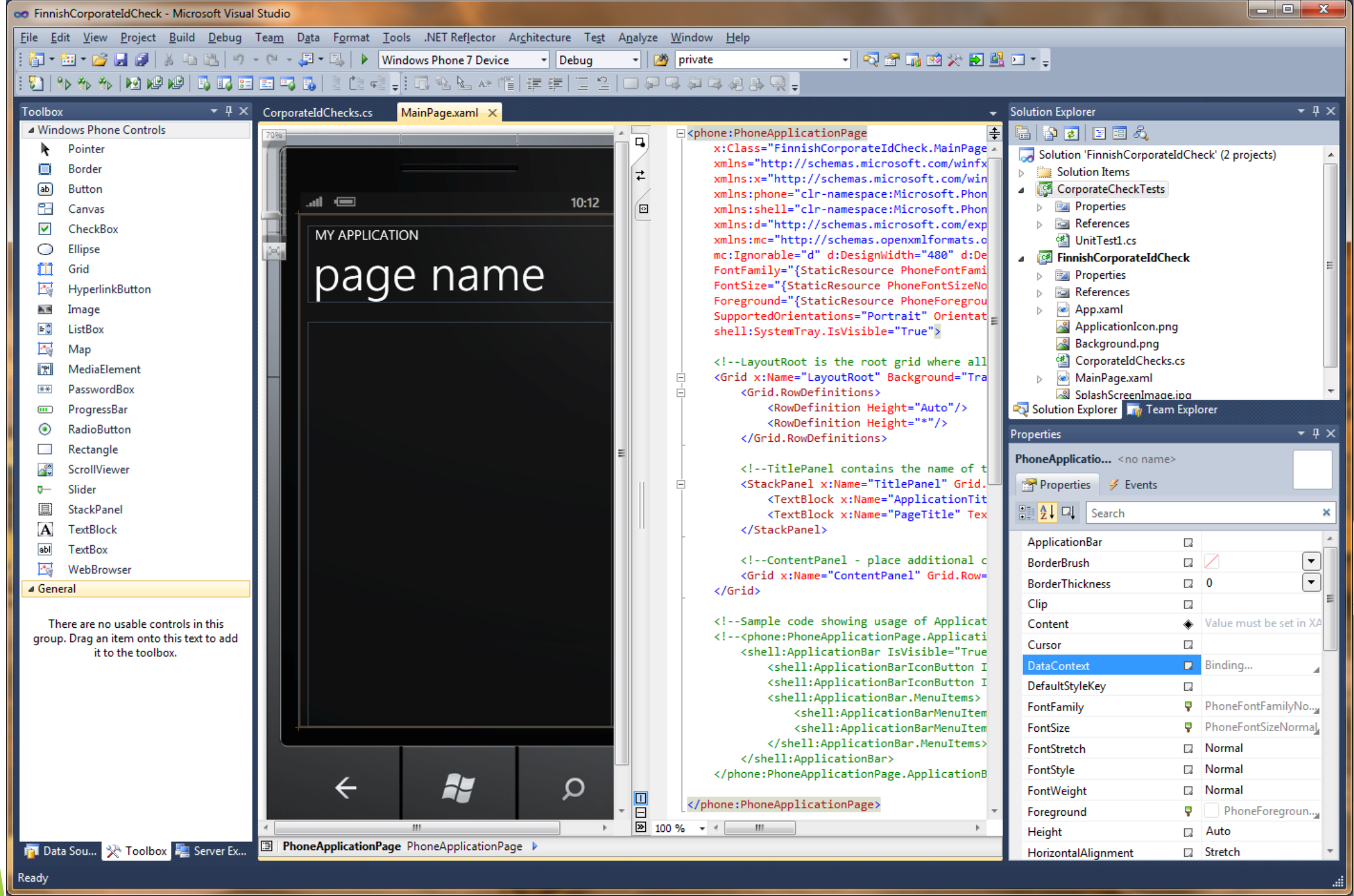

uc Use Cases
System Boundary
Waiter
receive order
Order Food
<<extend>>
Order Wine
confirm order
place order
Serve Food
Cook Food
Chef
<<extend>>
{if wine was ordered}
Serve Wine
Client
Eat Food
<<extend>>
Drink Wine
{if wine was served}
facilitate payment
pay
Cashier
accept payment
Pay for Food
<<extend>>
{if wine was consumed}
Pay for Wine


Client Tier
Middle Tier
Data Access
Screens
Methods
Controls
Data Workspace
WCF RIA Services
Silverlight 4
Browser Host
Desktop Host
Submit Pipeline
Queries
Data Workspace
WCF RIA Services
ASP.NET 4
IIS 7
SQL Server
Windows Azure
SharePoint
Other


VS2012Available - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL DESIGN FORMAT TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Local Machine
Debug
Solution Explorer
Search Solution Explorer (Ctrl+;
VS2012Available
Properties
AssemblyInfo.cs
References
Assets
Common
App.xaml
MainPage.xaml
MainPage.xaml.cs
MainPage
MainPage()
OnNavigate
_contentLoa
InitializeCor
Package.appxmanifest
VS2012Available_Temp
MainPage.xaml
What amazing app will you build today?
25%
Design
XAML
Team Explorer - Home
Search Work It
Home | DevDiv
My Work
Request Code Review
Pending Changes
Source Control Explorer
Work Items
New Query
Builds
Reports
Web Access
Settings
Document Outline
Data Sources
Server Explorer
Toolbox
Device
Ready


MergeConsole - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Program.cs XmlExample.vb XmlStuffLibrary
MergeConsole.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using MainLibrary;
namespace MergeConsole
{
class Program
Solution Explorer - MainLibraryILMerge
Solution 'MergeConsole' (3 projects)
MainLibraryILMerge
Properties
References
Example.cs
MergeConsole
Properties
References
MainLibraryILMerge
System
System.Core
System.Data
System.Data.DataSetExtensions
Output
Show output from: Build
------ Build started: Project: XmlStuffLibrary, Configuration: Debug Any CPU ------
XmlStuffLibrary -> C:\dev\CSharpVBTogetherAtLast\MergeConsole\XmlStuffLibrary\bin\Debug\XmlStuffLibrary.dll
------ Build started: Project: MainLibraryILMerge, Configuration: Debug Any CPU ------
C:\Windows\Microsoft.NET\Framework\v3.5\Csc.exe /noconfig /nowarn:1701,1702 /platform:x86 /errorreport:prompt /warn:4
Compile complete -- 0 errors, 0 warnings
MainLibraryILMerge -> C:\dev\CSharpVBTogetherAtLast\MergeConsole\MainLibrary\bin\Debug\MainLibrary.dll
MERGING: XmlStuffLibrary
"C:\Program Files (x86)\Microsoft\Ilmerge\Ilmerge.exe" /out:bin\Debug\MainLibrary.dll "obj\Debug\MainLibrary.dll"
------ Build started: Project: MergeConsole, Configuration: Debug Any CPU ------
C:\Windows\Microsoft.NET\Framework\v3.5\Csc.exe /noconfig /nowarn:1701,1702 /platform:x86 /errorreport:prompt /warn:4
Compile complete -- 0 errors, 0 warnings
MergeConsole -> C:\dev\CSharpVBTogetherAtLast\MergeConsole\MergeConsole\bin\Debug\MergeConsole.exe
========== Build: 3 succeeded or up-to-date, 0 failed, 0 skipped ==========
Find Results 1 Find Symbol Results Error List Output
Ready Ln 16 Col 1 Ch 1 INS


default.gif (C:\Backup\My Documents\Glass Onion\site\Untitled_1.aspx) - Microsoft Expression Web Designer
File Edit View Insert Format Tools Table Site Data View Task Panes Window Help
Type a question for help
Folder List
C:\Backup\My Documents\Glass Onio
_notes
gems
images
png
portal
style
Templates
default.dwt
defaultMain.dwt
aboutus.htm
background.png
classes.htm
contactus.htm
default.asp
default.htm
defaultOld.htm
gallery.htm
template.png
template.zip
Thumbs.db
Untitled_1.aspx
Web Site
classes.htm
Untitled_1.aspx
defaultMain.dwt*
default.dwt
<body> <div#Layer1> <DWT:editable> <form#form1> <asp:calendar#Calendar1>
Templates/default.dwt
<form id="form1" runat="server">
<asp:Calendar runat="server" id="Calendar1" BorderColor="Black" NextPrevFormat="FullMonth" Wi
<SelectedDayStyle BackColor="#CC3333" ForeColor="White" />
<TodayDayStyle BackColor="#CCCC99" />
<SelectorStyle BackColor="#CCCCCC" Width="1%" ForeColor="#333333" Font-Size="8pt" Font-Na
<DayStyle Width="14%" />
<OtherMonthDayStyle ForeColor="#999999" />
<NextPrevStyle ForeColor="White" Font-Size="8pt" />
<DayHeaderStyle BackColor="#CCCCCC" Height="10pt" ForeColor="#333333" Font-Size="7pt" Fon
<TitleStyle BackColor="Black" Height="14pt" ForeColor="White" Font-Size="13pt" Font-Bold=
</asp:Calendar>
Glass Onion Studios
content
April May June
Su Mo Tu We Th Fr Sa
30 1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31 1 2 3
4 5 6 7 8 9 10
CLASSES
GALLERY
CONTACT US
Design Split Code
Toolbox
Standard
AdRotator
BulletedList
Button
Calendar
CheckBox
CheckBoxList
ContentPlaceHolder
DropDownList
FileUpload
HiddenField
HyperLink
Image
ImageButton
ImageMap
Label
LinkButton
ListBox
Literal
Localize
MultiView
Tag Properties
Tag Properties CSS Properties
<asp:calendar#Calendar1>
Accessibility
AccessKey
Caption
CaptionAlign NotSet
TabIndex 0
UseAccessi... True
Appearance
BackColor White
BorderColor Black
DayName... Shortest
Font Times New Ro...
ForeColor Black
NextPrevF... FullMonth
TitleFormat Month
BorderStyle NotSet
BorderWidth
CssClass
FirstDayOf... Default
NextMonth... >
PrevMonth... <
Apply Styles
Apply Styles Manage Styles
New Style... Options
Attach Style Sheet...
Select CSS style to apply:
Clear Styles
Current Page
Accessibility
Page Line Issue Type Checkpoint Problem Summary
Visual Aids: On Style Application: Auto 104 KB Quirks 748 x 407 IE6 CSS 2.1


FinnishCorporateIdCheck - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Format Tools .NET Reflector Architecture Test Analyze Window Help
Windows Phone 7 Device
Debug
private
Toolbox
Windows Phone Controls
Pointer
Border
Button
Canvas
CheckBox
Ellipse
Grid
HyperlinkButton
Image
ListBox
Map
MediaElement
PasswordBox
ProgressBar
RadioButton
Rectangle
ScrollViewer
Slider
StackPanel
TextBlock
TextBox
WebBrowser
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
CorporateIdChecks.cs
MainPage.xaml
10:12
MY APPLICATION
page name
<phone:PhoneApplicationPage
x:Class="FinnishCorporateIdCheck.MainPage
xmlns="http://schemas.microsoft.com/winfx
xmlns:x="http://schemas.microsoft.com/win
xmlns:phone="clr-namespace:Microsoft.Phon
xmlns:shell="clr-namespace:Microsoft.Phon
xmlns:d="http://schemas.microsoft.com/exp
xmlns:mc="http://schemas.openxmlformats.o
mc:Ignorable="d" d:DesignWidth="480" d:De
FontFamily="{StaticResource PhoneFontFami
FontSize="{StaticResource PhoneFontSizeNo
Foreground="{StaticResource PhoneForegrou
SupportedOrientations="Portrait" Orientat
shell:SystemTray.IsVisible="True">
<!--LayoutRoot is the root grid where all
<Grid x:Name="LayoutRoot" Background="Tra
<Grid.RowDefinitions>
<RowDefinition Height="Auto"/>
<RowDefinition Height="*"/>
</Grid.RowDefinitions>
<!--TitlePanel contains the name of t
<StackPanel x:Name="TitlePanel" Grid.
<TextBlock x:Name="ApplicationTit
<TextBlock x:Name="PageTitle" Tex
</StackPanel>
<!--ContentPanel - place additional c
<Grid x:Name="ContentPanel" Grid.Row=
</Grid>
<!--Sample code showing usage of Applicat
<!--<phone:PhoneApplicationPage.Applicati
<shell:ApplicationBar IsVisible="True
<shell:ApplicationBarIconButton I
<shell:ApplicationBarIconButton I
<shell:ApplicationBar.MenuItems>
<shell:ApplicationBarMenuItem
<shell:ApplicationBarMenuItem
</shell:ApplicationBar.MenuItems>
</shell:ApplicationBar>
</phone:PhoneApplicationPage.ApplicationB
</phone:PhoneApplicationPage>
100 %
PhoneApplicationPage PhoneApplicationPage
Solution Explorer
Solution 'FinnishCorporateIdCheck' (2 projects)
Solution Items
CorporateCheckTests
Properties
References
UnitTest1.cs
FinnishCorporateIdCheck
Properties
References
App.xaml
ApplicationIcon.png
Background.png
CorporateIdChecks.cs
MainPage.xaml
SplashScreenImage.jpg
Solution Explorer
Team Explorer
Properties
PhoneApplicatio... <no name>
Properties
Events
Search
ApplicationBar
BorderBrush
BorderThickness
0
Clip
Content
Value must be set in XA
Cursor
DataContext
Binding...
DefaultStyleKey
FontFamily
PhoneFontFamilyNo...
FontSize
PhoneFontSizeNormal
FontStretch
Normal
FontStyle
Normal
FontWeight
Normal
Foreground
PhoneForegroun...
Height
Auto
HorizontalAlignment
Stretch
Data Sou...
Toolbox
Server Ex...
Ready

C# Εισαγωγικά

Λίγα λόγια για την C#

- Δημιουργήθηκε από την Microsoft (2000)
- Βασικός πρωτεργάτης: Anders Hejlsberg
- Πιστοποιημένη από ECMA και ISO/IEC
- Βασικές υλοποιήσεις: Visual C#, .NET Framework, Mono, DotGNU
- Μοντέρνα, αντικειμενοστρεφής, γενικού σκοπού
- Δομημένη γλώσσα
- Εύκολη στην εκμάθηση

Σημαντικά χαρακτηριστικά της C#

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading
- LINQ and Lambda Expressions
- Integration with Windows

Βασικές υλοποιήσεις σε C#

- Windows applications
- Web applications
- Web services
- Mobile applications

Προϊόντα Visual Studio

- 2012 editions
  - Ultimate
  - Premium
  - Professional
  - Test Professional
  - Team Foundation Server
  - Compare editions
- Express 2012 products
  - Express for Web
  - Express for Windows 8
  - Express for Windows Desktop
  - Express for Windows Phone
  - Team Foundation Server Express

Microsoft

Επικοινωνήστε μαζί μας Είσοδος Καρότσι (0)

Καλώς ορίσατε στο κατάστημά μας

Αναζήτηση βάσει λέξης-κλειδιού, αριθμού SKU ή αριθμού στοιχείου

Αγορά προϊόντων Προσφορές για σπουδαστές

Για προγραμματιστές

Ταξινόμηση κατά: Δημοφιλέστεροι πωλητές

Προβολή 1-4 Αποτελέσματα

Visual Studio Premium 2012 με MSDN

Το Visual Studio Premium 2012 παρέχει εργαλεία σχεδιασμού και διαχείρισης έργου για μια ευέλικτη διαδικασία ανάπτυξης λογισμικού.

Έναρξη στις €7.534,00

Visual Studio Professional 2012 με MSDN

Το Visual Studio Professional 2012 με συνδρομή MSDN είναι η σωστή επιλογή για τη δόμηση πολυεπίπεδων εφαρμογών για το web, το cloud και συσκευές.

Έναρξη στις €1.477,00

Visual Studio Ultimate 2012 με MSDN

Το Visual Studio Ultimate 2012 με MSDN προορίζεται για τις ομάδες ανάπτυξης λογισμικού που θέλουν να επωφεληθούν από τις τάσεις της αναδυόμενης τεχνολογίας.

Έναρξη στις €16.374,00

Visual Studio Test Professional 2012 με MSDN

Το Visual Studio Professional 2012 με MSDN 2012 είναι μια ολοκληρωμένη λύση σχεδιασμού, εκτέλεσης και διαχείρισης δοκιμών.

Έναρξη στις €2.671,00

Βασική Δομή ενός προγράμματος C#

Από τι αποτελείται ένα πρόγραμμα γραμμένο σε γλώσσα C#? (1/2)

- Δήλωση ενός namespace
- Μια ή περισσότερες τάξεις
- Τις ιδιότητες των τάξεων
- Τις μεθόδους των τάξεων
- Μια κύρια μέθοδο “Main”
- Σχόλια
- Διάφορες δηλώσεις ή/και εκφράσεις
- Λέξεις κλειδιά

Από τι αποτελείται ένα πρόγραμμα γραμμένο σε γλώσσα C#? (2/2)

- Το “namespace” στην C# είναι μια συλλογή ή ένα σύνολο από τάξεις
- Κάθε πρόγραμμα πρέπει να περιλαμβάνει namespaces, συνήθως το όνομα του namespace που περιλαμβάνει τις τάξεις που θα χρησιμοποιήσουμε ή που θα δημιουργήσουμε, καθώς και κάποιο/κάποια namespace/s τα οποία περιλαμβάνουν τάξεις που χρησιμοποιούμε συχνά στην C#
- Για να συμπεριλάβουμε στο πρόγραμμά μας ένα ή περισσότερα namespaces που προϋπάρχουν στην C#, κάνουμε χρήση της κωδικής λέξης “using”
- Η μέθοδος “Main” είναι το σημείο εισόδου κάθε προγράμματος C#. Αυτή προσδιορίζει τι θα κάνει το πρόγραμμά μας όταν ξεκινήσει να εκτελείται
- Προσοχή:
  - Η C# είναι case sensitive!
  - Κάθε δήλωση ή έκφραση πρέπει να τελειώνει με το χαρακτήρα ;

Το πρώτο πρόγραμμα «Hello World»

```
using System;
namespace HelloWorldApplication
{
   class HelloWorld
   {
      static void Main(string[] args)
      {
         /* this is my hello world programm*/
         Console.WriteLine("Hello World");
         Console.ReadKey();
      }
   }
}
```

Εκτέλεση ενός προγράμματος C# με το Visual Studio

- Εκκίνηση του Visual Studio.
- Από το μενού επιλέγουμε File->New->Project.
- Επιλογή Visual C# από τα templates και έπειτα Windows.
- Επιλογή Console Application.
- Επιλογή ονόματος για το project μας και OK.
- Βλέπουμε το πρόγραμμά μας στον Solution Explorer.
- Γράφουμε τον κώδικά μας μέσα στον Code Editor.
- Πατάμε το κουμπί «Run» ή το F5 key to run the project.

Εκτέλεση ενός προγράμματος C# σε περιβάλλον command-line

- Ανοίγουμε έναν κειμενογράφο (text editor) και πληκτρολογούμε τον κώδικα της προηγούμενης διαφάνειας.
- Σώζουμε το αρχείο με κάποιο όνομα με επέκταση (cs) π.χ. «helloworld.cs»
- Ανοίγουμε το περιβάλλον command prompt και μεταβαίνουμε στο σημείο που βρίσκεται το σωσμένο αρχείο.
- Πληκτρολογούμε csc helloworld.cs και πατάμε enter για να το μεταγλωττίσουμε
- Εάν δεν βρεθούν λάθη, το πρόγραμμα C# θα μεταγλωττιστεί επιτυχώς και θα παραχθεί το εκτελέσιμο αρχείο (στον ίδιο κατάλογο) helloworld.exe
- Τέλος, πληκτρολογώντας το όνομα του παραπάνω εκτελέσιμου, εκτελείται το πρόγραμμά μας!

Το όνομα μια τάξης σε C#

- Πρέπει να ξεκινάει από γράμμα
- Ακολουθείται από μια σειρά από:
  - Γράμματα
  - Αριθμούς
  - Underscore _
- Οι άλλοι χαρακτήρες απαγορεύονται
- Επίσης δεν μπορεί να είναι μια λέξη κλειδί της C#

Reserved Keywords in C#

abstract	as	base	bool	break	byte	case
catch	char	checked	class	const	continue	decimal
default	delegate	do	double	else	enum	event
explicit	extern	false	finally	fixed	float	for
foreach	goto	if	implicit	in	in	int
interface	internal	is	lock	long	namespace	new
null	object	operator	out	out	override	params
private	protected	public	readonly	ref	return	sbyte
sealed	short	sizeof	stackalloc	static	string	struct
switch	this	throw	true	try	typeof	uint
ulong	unchecked	unsafe	ushort	using	virtual	void
volatile	while					

Contextual Keywords in C#

add	alias	ascending	descending	dynamic	from	get
global	group	into	join	let	orderby	partial
partial	remove	select	set			

Ένα παράδειγμα βασικού προγράμματος

```
using System;
namespace RectangleApplication
{
    class Rectangle
    {
        double length;
        double width;
        public void Acceptdetails()
        {
            length = 4.5;
            width = 3.5;
        }
        public double GetArea()
        {
            return length * width;
        }
        public void Display()
        {
            Console.WriteLine("Length: {0}", length);
            Console.WriteLine("Width: {0}", width);
            Console.WriteLine("Area: {0}", GetArea());
        }
    }

    class ExecuteRectangle
    {
        static void Main(string[] args)
        {
            Rectangle r = new Rectangle();
            r.Acceptdetails();
            r.Display();
            Console.ReadLine();
        }
    }
}
```

Τύποι δεδομένων στη C#

- **Value types**
- **Reference types**
- **Pointer types**

Value Types

- Προέρχονται από την τάξη «System.ValueType»
- Είναι οι τύποι δεδομένων που περιέχουν «πραγματικά» δεδομένα
- Για να βρούμε το ακριβές μέγεθος ενός τύπου δεδομένων μπορούμε να χρησιμοποιήσουμε τη μέθοδο sizeof()

Τύπος	Περιγραφή	Κλίμακα	Αρχική τιμή
bool	Boolean value	True or False	False
byte	8-bit unsigned integer	0 to 255	0
char	16-bit Unicode character	U +0000 to U +ffff	'\0'
decimal	128-bit precise decimal values with 28-29 significant digits	$(-7.9 \times 10^{28}$ to $7.9 \times 10^{28}) / 10^{0 \text{ to } 28}$	0.0M
double	64-bit double-precision floating point type	$(+/-)5.0 \times 10^{-324}$ to $(+/-)1.7 \times 10^{308}$	0.0D
float	32-bit single-precision floating point type	-3.4×10^{38} to $+ 3.4 \times 10^{38}$	0.0F
int	32-bit signed integer type	-2,147,483,648 to 2,147,483,647	0
long	64-bit signed integer type	-923,372,036,854,775,808 to 9,223,372,036,854,775,807	0L
sbyte	8-bit signed integer type	-128 to 127	0
short	16-bit signed integer type	-32,768 to 32,767	0
uint	32-bit unsigned integer type	0 to 4,294,967,295	0
ulong	64-bit unsigned integer type	0 to 18,446,744,073,709,551,615	0
ushort	16-bit unsigned integer type	0 to 65,535	0

Reference Types

- Αυτοί οι τύποι δεδομένων δεν περιέχουν την πραγματική τιμή των δεδομένων, αλλά μια αναφορά σε αυτά
- Στην πράξη, περιέχουν τη διεύθυνση μιας θέσης μνήμης
- Βασικοί τύποι δεδομένων αναφορών:
  - Object Type
  - Dynamic Type
  - String Type

Pointer Types

- Οι δείκτες στην C# είναι της ίδιας κατηγορίας τύπων δεδομένων όπως και στην C++
- Οι δείκτες περιέχουν τις διευθύνσεις θέσεων μνήμης άλλων μεταβλητών
- Δεν τους χρησιμοποιούμε συχνά στην C# αφού μπορούμε να κάνουμε τις περισσότερες λειτουργίες με τη χρήση ειδικών τάξεων
- Για να τους χρησιμοποιήσουμε κάνουμε χρήση «unsafe code»

Βασικές μέθοδοι μετατροπής τύπων δεδομένων στη C#

Methods & Description
ToBoolean Converts a type to a Boolean value, where possible.
ToByte Converts a type to a byte.
ToChar Converts a type to a single Unicode character, where possible.
ToDateTime Converts a type (integer or string type) to date-time structures.
ToDecimal Converts a floating point or integer type to a decimal type.
ToDouble Converts a type to a double type.
ToInt16 Converts a type to a 16-bit integer.
ToInt32 Converts a type to a 32-bit integer.
ToInt64 Converts a type to a 64-bit integer.
ToSbyte Converts a type to a signed byte type.
ToSingle Converts a type to a small floating point number.
ToString Converts a type to a string.
ToType Converts a type to a specified type.
ToUInt16 Converts a type to an unsigned int type.
ToUInt32 Converts a type to an unsigned long type.
ToUInt64 Converts a type to an unsigned big integer.

Δήλωση μεταβλητών στη C#

<data_type> <variable_list>;

π.χ.

int i, j, k;

char c;

float f, wage;

double d;

Αρχικοποίηση μεταβλητών στη C#

- variable_name = value;

Ή εναλλακτικά, δήλωση και αρχικοποίηση μαζί:

- <data_type> <variable_name> = value;

π.χ.

int d = 3, f = 5;

Τελεστές στη C# (Operators)

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Misc Operators (Λοιποί τελεστές)

Arithmetic Operators

Τελεστής	Περιγραφή	Παράδειγμα
+	Πρόσθεση 2 τιμών	A + B
-	Αφαίρεση μιας τιμής από μια άλλη	A - B
*	Πολλαπλασιασμός 2 τιμών	A * B
/	Διαίρεση διαιρετού και διαιρέτη	B / A
%	Υπολογισμός υπολοίπου διαίρεσης	B % A
++	Αύξηση τιμής κατά 1 μονάδα	A++
--	Μείωση τιμής κατά 1 μονάδα	A--

Relational Operators

Τελεστής	Περιγραφή	Παράδειγμα (A=1, B=2)
==	Έλεγχος για το αν 2 τελεστές είναι ίσοι	(A == B) ψευδής.
!=	Έλεγχος για το αν 2 τελεστές είναι άνισοι	(A != B) αληθής.
>	Έλεγχος για το αν ένας τελεστής είναι μεγαλύτερος από έναν άλλο	(A > B) ψευδής.
<	Έλεγχος για το αν ένας τελεστής είναι μικρότερος από έναν άλλο	(A < B) αληθής.
>=	Έλεγχος για το αν ένας τελεστής είναι μεγαλύτερος, ή ίσος από έναν άλλο	(A >= B) ψευδής.
<=	Έλεγχος για το αν ένας τελεστής είναι μικρότερος, ή ίσος από έναν άλλο	(A <= B) αληθής.

Logical Operators

Τελεστής	Περιγραφή	Παράδειγμα (A=true, B=false)
&&	Λογικό «ΚΑΙ»	(A && B) is false.
||	Λογικό «Η»	(A || B) is true.
!	Λογικό «ΌΧΙ»	!(A && B) is true.

Bitwise Operators

Τελεστής	Περιγραφή	Παράδειγμα (A = 60, B = 13, ή αλλιώς A = 0011 1100, B = 0000 1101
&	Binary AND Operator copies a bit to the result if it exists in both operands.	(A & B) will give 12. which is 0000 1100
|	Binary OR Operator copies a bit if it exists in either operand.	(A | B) will give 61, which is 0011 1101
^	Binary XOR Operator copies the bit if it is set in one operand but not both.	(A ^ B) will give 49, which is 0011 0001
~	Binary Ones Complement Operator is unary and has the effect of 'flipping' bits.	(~A) will give -61, which is 1100 0011 in 2's complement due to a signed binary number.
<<	Binary Left Shift Operator. The left operands value is moved left by the number of bits specified by the right operand.	A << 2 will give 240, which is 1111 0000
>>	Binary Right Shift Operator. The left operands value is moved right by the number of bits specified by the right operand.	A >> 2 will give 15, which is 0000 1111

Assignment Operators

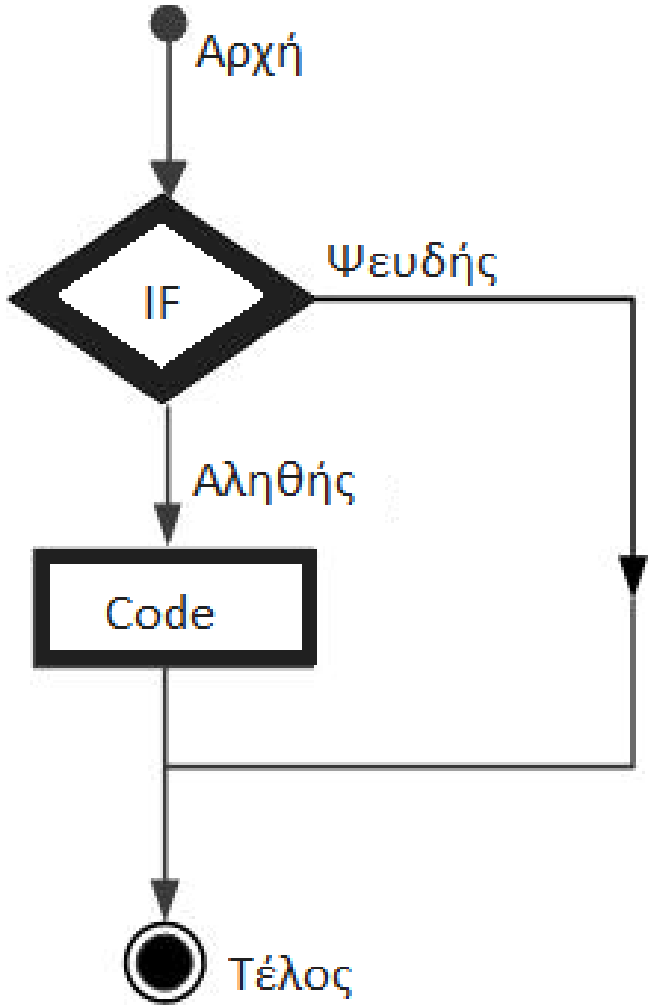
Τελεστής	Περιγραφή	Παράδειγμα
=	Simple assignment operator, Assigns values from right side operands to left side operand	C = A + B θα δώσει τιμή A + B στο C
+=	Add AND assignment operator, It adds right operand to the left operand and assign the result to left operand	C += A είναι ισοδύναμο με C = C + A
-=	Subtract AND assignment operator, It subtracts right operand from the left operand and assign the result to left operand	C -= A είναι ισοδύναμο με C = C - A
*=	Multiply AND assignment operator, It multiplies right operand with the left operand and assign the result to left operand	C *= A είναι ισοδύναμο με C = C * A
/=	Divide AND assignment operator, It divides left operand with the right operand and assign the result to left operand	C /= A είναι ισοδύναμο με C = C / A
%=	Modulus AND assignment operator, It takes modulus using two operands and assign the result to left operand	C %= A είναι ισοδύναμο με C = C % A
<<=	Left shift AND assignment operator	C <<= 2 ταυτίζεται με C = C << 2
>>=	Right shift AND assignment operator	C >>= 2 ταυτίζεται με C = C >> 2
&=	Bitwise AND assignment operator	C &= 2 ταυτίζεται με C = C & 2
^=	bitwise exclusive OR and assignment operator	C ^= 2 ταυτίζεται με C = C ^ 2
|=	bitwise inclusive OR and assignment operator	C |= 2 ταυτίζεται με C = C | 2

Misc Operators

Τελεστής	Περιγραφή	Παράδειγμα ()
sizeof()	Returns the size of a data type.	sizeof(int), will return 4.
typeof()	Returns the type of a class.	typeof(StreamReader);
&	Returns the address of an variable.	&a; will give actual address of the variable.
*	Pointer to a variable.	*a; will pointer to a variable.
? :	Conditional Expression	If Condition is true ? Then value X : Otherwise value Y
is	Determines whether an object is of a certain type.	If(Ford is Car) // checks if Ford is an object of the Car class.
as	Cast without raising an exception if the cast fails.	Object obj = new StringReader("Hello"); StringReader r = obj as StringReader;

Δομές Ελέγχου

Συνθήκη IF



Δηλώσεις ελέγχου

Δηλώσεις	Περιγραφή
if statement	Μια δήλωση **if** περιλαμβάνει μια έκφραση Boolean, ακολουθούμενη από μια ή περισσότερες δηλώσεις
if...else statement	Η δήλωση if…else διαφέρει από την απλή if, στο γεγονός ότι περιέχει ένα τμήμα δηλώσεων που θα εκτελεστεί αν η αρχική συνθήκη είναι ψευδής
nested if statements	Εμφωλευμένες συνθήκες if, if…else, η μία μέσα στην άλλη, σε όσο βάθος ορίζει ο προγραμματιστής
switch statement	Με τη συνθήκη switch ελέγχουμε αν η τιμή μιας μεταβλητής επαληθεύεται μέσα σε μια δοσμένη λίστα τιμών
nested switch statements	Εμφωλευμένες συνθήκες switch

if syntax

```
if(boolean_expression)
{
/* κώδικας που θα εκτελεστεί εφόσον η συνθήκη είναι
αληθής */
}
```

if..else syntax

```
if(boolean_expression)
{ /* κώδικας που θα εκτελεστεί εφόσον η συνθήκη είναι
αληθής */ }
else
{ /* κώδικας που θα εκτελεστεί εφόσον η συνθήκη είναι
ψευδής */ }
```

Nested if syntax

```
if( boolean_expression 1)
{ /* κώδικας που θα εκτελεστεί εφόσον η συνθήκη 1 είναι
αληθής */
if(boolean_expression 2)
{ /* κώδικας που θα εκτελεστεί εφόσον η συνθήκη 2 είναι
αληθής */ } }
```

switch syntax

```
switch(expression){
    case constant-expression  :
       statement(s);
       break;  /* προαιρετικό */
    case constant-expression  :
       statement(s);
       break;  /* προαιρετικό */
            …..
    /* οποιοσδήποτε αριθμός δηλώσεων */
    default :  /* προαιρετικό */
       statement(s);
}
```

Nested switch syntax

```
switch(ch1)
{
   case 'A':
      statement(s);
      switch(ch2)
      {
         case 'A':
            statement(s);
            break;
         case 'B': /* κώδικας εντός του B */
      }
      break;
   case 'B': /* κώδικας εκτός του B */
}
```

Επαναληπτικοί βρόγχοι

Βρόγχοι επανάληψης στη C#

Βρόγχος	Περιγραφή
while loop	Εκτελεί επαναληπτικά ένα τμήμα κώδικα, εφόσον ικανοποιείται η συνθήκη. Ο έλεγχος της συνθήκης γίνεται πριν εκτελεστεί το τμήμα κώδικα.
for loop	Εκτέλεση συγκεκριμένου κώδικα ορισμένες φορές, δοσμένες από τον προγραμματιστή. Στον βρόγχο δηλώνεται και η μεταβλητή βρόγχου.
do...while loop	Εκτελεί επαναληπτικά ένα τμήμα κώδικα, εφόσον ικανοποιείται η συνθήκη. Ο έλεγχος της συνθήκης γίνεται αφού εκτελεστεί το τμήμα κώδικα.
nested loops	Εμφώλευση βρόγχων.

Δηλώσεις ελέγχου βρόγχων

Δήλωση ελέγχου	Περιγραφή
break statement	Τερματίζει την εκτέλεση του βρόγχου και μεταφέρει την εκτέλεση του προγράμματος στον κώδικα αμέσως μετά τον βρόγχο.
continue statement	Έχει ως επίπτωση να μην εκτελεστεί το υπόλοιπο του κώδικα του βρόγχου μετά την εντολή continue για την ίδια επανάληψη και να μεταβεί το πρόγραμμα στον επόμενο κύκλο επανάληψης.

While syntax

```
while(condition)
{
   statement(s);
}
```

for syntax

```
for ( init; condition; increment )
{
   statement(s);
}
```

do… while syntax

```
do
{

   statement(s);

} while( condition );
```

Nested for syntax

```
for ( init; condition; increment )
{
   for ( init; condition; increment )
   {
      statement(s);
   }
   statement(s);
}
```

Επίπεδα πρόσβασης στην C#

- Public
- Private
- Protected
- Internal
- Protected internal

Public access

Με τον προσδιοριστή πρόσβασης «public», μια τάξη επιτρέπει να είναι ορατά τόσο τα κατηγορήματά της, όσο και οι λειτουργίες της στον «έξω κόσμο»

Private access

Με τον προσδιοριστή πρόσβασης «private», μια τάξη δεν επιτρέπει να είναι ορατά τα κατηγορήματά της, ούτε και οι λειτουργίες της στον «έξω κόσμο». Η πρόσβαση σε αυτά είναι εφικτή μόνο από συναρτήσεις της ίδιας της τάξης. Ούτε οι τάξεις που κληρονομούν από αυτήν μπορούν να έχουν πρόσβαση.

Protected access

Με τον προσδιοριστή πρόσβασης «protected», μια τάξη επιτρέπει να είναι ορατά τα κατηγορήματά της και οι λειτουργίες της μόνο στην ίδια και στα «παιδιά» της. Χρησιμοποιείται συνήθως κατά την κληρονομικότητα.

Συναρτήσεις στη C#

Γενική σύνταξη συνάρτησης C#

```
<Access Specifier> <Return Type> <Method Name>(Parameter List)
{
   Method Body
}
```

Παράδειγμα χρήσης συνάρτησης

```
class MyNumberClass
{
   public int FindMax(int num1, int num2)
   {
      int result;

      if (num1 > num2)
         result = num1;
      else
         result = num2;

      return result;
   }

}
```

Κλήση της προηγούμενης συνάρτησης σε πρόγραμμα

```
using System;

namespace CalculatorApp
{
   class MyNumberClass
   {
…… //κώδικας που γράψαμε πριν
}
      static void Main(string[] args)
      {
         int a = 10;
         int b = 20;
         int max_n;
         MyNumberClass n = new MyNumberClass();
         max_n= n.FindMax(a, b);
         Console.WriteLine("Max value is : {0}", max_n);
         Console.ReadLine();
      }
   }
```

Extra: Η συνάρτηση «παραγοντικό» (recursion)

```
public int factorial(int num)
        {
            int result;
            if (num == 1)
            {
                return 1;
            }
            else
            {
                result = factorial(num - 1) * num;
                return result;
            }
        }
```

Πίνακες στη C#

Δήλωση πινάκων/αρχικοποίηση

datatype[] arrayName;

Παράδειγμα:
double[] balance = new double[100];

Απόδοση τιμών σε πίνακες

Element[0] = 100
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109

Πρόσβαση σε τιμή πίνακα:
int myvalue = Element[8];

Πίνακες 2 διαστάσεων

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

```
int [ , ] m;
```

```
int [,] a = int [3,4] = {
 {0, 1, 2, 3} ,    /*  row 0 */
 {4, 5, 6, 7} ,    /*  row 1 */
 {8, 9, 10, 11}    /* row 2 */
};
```

```
int[,] a = new int[5, 2] {{0,0}, {1,2}, {2,4}, {3,6}, {4,8} };
```

C# Strings

System.String class

π.χ.

```
string fname, lname;
        fname = “Efthimios”;
        lname = “Alepis”;
```

String properties

Property Name & Description
Chars Gets the *Char* object at a specified position in the current *String* object.
Length Gets the number of characters in the current String object.

String methods

Method Name & Description
public static int Compare(string strA, string strB) *Compares two specified string objects and returns an integer that indicates their relative position in the sort order.*
public static string Concat(string str0, string str1) *Concatenates two string objects.*
public bool Contains(string value) *Returns a value indicating whether the specified string object occurs within this string.*
public static string Copy(string str) *Creates a new String object with the same value as the specified string.*
public bool EndsWith(string value) *Determines whether the end of the string object matches the specified string.*
public bool Equals(string value) *Determines whether the current string object and the specified string object have the same value.*

Χρήση κληρονομικότητας στη C#

Base and Derived Class (superclass, childclass)

```
<acess-specifier> class <base_class>
{
 ...
}
class <derived_class> : <base_class>
{
 ...
}
```

Πολλαπλή κληρονομικότητα στη C#?

- Προσοχή! Η C# ΔΕΝ υποστηρίζει πολλαπλή κληρονομικότητα (multiple inheritance)
- Για να επιτευχθεί κάτι τέτοιο, πρέπει να γίνει χρήση των “Interfaces”

Υπερφόρτωση συναρτήσεων (function overloading)

```
void print(int i)
{
   Console.WriteLine("Printing int: {0}", i );
}

void print(double f)
{
   Console.WriteLine("Printing float: {0}" , f);
}

void print(string s)
{
   Console.WriteLine("Printing string: {0}", s);
}
```

C# Interface

- Το “interface” στη C# είναι ένα είδος προγραμματιστικού «συμβολαίου», το οποίο πρέπει να τηρήσουν όσες τάξεις κληρονομήσουν από αυτό
- Μας ενδιαφέρει το «τι» πρέπει να έχει μια τάξη
- Τα παιδιά που κληρονομούν από το interface ασχολούνται με το «πως» θα υλοποιήσουν αυτό το οποίο ορίζει το interface

Χειρισμός εξαιρέσεων στη C#

C# Exception handling

•**try**: A try block identifies a block of code for which particular exceptions will be activated. It's followed by one or more catch blocks.
•**catch**: A program catches an exception with an exception handler at the place in a program where you want to handle the problem. The catch keyword indicates the catching of an exception.
•**finally**: The finally block is used to execute a given set of statements, whether an exception is thrown or not thrown. For example, if you open a file, it must be closed whether an exception is raised or not.
•**throw**: A program throws an exception when a problem shows up. This is done using a throw keyword.

```
try
{
   // statements causing exception
}
catch( ExceptionName e1 )
{
   // error handling code
}
catch( ExceptionName e2 )
{
   // error handling code
}
catch( ExceptionName eN )
{
   // error handling code
}
finally
{
   // statements to be executed
}
```

```
try
{
    result = num1 / num2;
}
catch (DivideByZeroException e)
{
    Console.WriteLine("Exception caught: {0}", e);
}
finally
{
    Console.WriteLine("Result: {0}", result);
}
```

Input/Output Classes in C#

Basic I/O Classes

I/O Class	Περιγραφή
BinaryReader	Reads primitive data from a binary stream.
BinaryWriter	Writes primitive data in binary format.
BufferedStream	A temporary storage for a stream of bytes.
Directory	Helps in manipulating a directory structure.
DirectoryInfo	Used for performing operations on directories.
DriveInfo	Provides information for the drives.
File	Helps in manipulating files.
FileInfo	Used for performing operations on files.
FileStream	Used to read from and write to any location in a file.
MemoryStream	Used for random access to streamed data stored in memory.
Path	Performs operations on path information.
StreamReader	Used for reading characters from a byte stream.
StreamWriter	Is used for writing characters to a stream.
StringReader	Is used for reading from a string buffer.
StringWriter	Is used for writing into a string buffer.

C# Advanced Topics!

- C# - Attributes
- C# - Reflection
- C# - Properties
- C# - Indexers
- C# - Delegates
- C# - Events
- C# - Collections
- C# - Generics
- C# - Anonymous Methods
- C# - Unsafe Codes
- C# - Multithreading

Now let's start programming!