

Advanced Programming C#

Lecture 2

dr inż. Małgorzata Janik majanik@if.pw.edu.pl

Winter Semester 2017/2018



C# Classes, Properties, Controls

- using
 - like import in Java: bring in namespaces
- namespace
 - disambiguation of names
 - like Internet hierarchical names and Java naming
- class
 - like in Java
 - single inheritance up to object

Namespaces



- Permits isolation of names
- Can be nested
- Access via fully qualified names



Language Features : Classes

- Single inheritance
- Multiple interface implementation
- Class members
 - Constants, fields, methods, properties, indexers, events, operators, constructors, destructors
 - Static and instance members
 - Nested types
- Member access
 - Public, protected, internal, private

Properties

- Properties are "smart fields"
 - Natural syntax, accessors, inlining

```
public class Button: Control
{
    private string caption;
    public string Caption {
        get {
            return caption;
        }
        set {
            caption = value;
            Repaint();
        }
    }
    Button b = new Button();
    b.Caption = "OK";
    String s = b.Caption;
}
```

foreach loop

Iteration of arrays

public static void Main(string[] args) {
 foreach (string s in args) Console.WriteLine(s);
}

Iteration of user-defined collections

foreach (Customer c in customers.OrderBy("name")) {
 if (c.Orders.Count != 0) {
 ...
 }
}

Pong multi-ball game



Windows Forms

Adding external resources

- Solution Explorer
 - Properties
 - Double click Resources.resx
 - Add Resource...
 - Add Existing File...

http://images.all-free-download.com/images/graphiclarge/soccer_ball_clip_art_13012.jpg



Reso	urces.resx	😐 🗙 Ball.cs [Design]	Program.cs	Form1.Designer.cs	Ball.Designer.cs	Ball.cs	Form1.cs [Design]	Form1.cs	Ŧ	Solution Explorer	- ₽×
abr	Strings 👻 🍍	ු Add Resource 👻 Ren	move Resource	📰 🔹 Access Modifier: 1	Internal 👻					© © ☆ 'o - ≒ 🖒 © ©	
										Search Solution Explorer (Ctrl+;)	p.
	Name	A	Value				Comment			a J Solution 'WindowsFormsApplica	tion2' (1 project
	String1									▲ ✓ 🖙 WindowsFormsApplication2	
										🔺 🕯 🌽 Properties	
										C# AssemblyInfo.cs	
										A 🗃 🛐 Resources.resx	
										Resources.Designer	.cs

Task 1: PictureBox Movement



10/23

Creating new class

- We want the ball to have additional property: velocity (Vx, Vy).
 - We want to extend PictreBox class, adding this additional property.
- Application \rightarrow Add \rightarrow Class...

```
- Ball.cs
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace WindowsFormsApplication_Lab2_Pong
{
        class Ball
        {
        }
}
```



Class inheritance

- Inheritance **Ball** : **PictureBox**
- PictureBox exists in system.Windows.Forms namespace.
- Fields can be created and <u>immidately</u> <u>initialized.</u>

using System.Windows.Forms;

```
namespace WindowsFormsApplication_Lab2_Pong
{
    class Ball : PictureBox
    {
        private double vx = 2;
        private double vy = 2;
    }
}
```

- Build & Start your project
- You can now add Ball control from the Toolbox!

Properties

{

}

Pr	•	д	×				
ball WindowsFormsApplication2.Ball							
	💱 🖓 🗲 🖉						
Ŧ	Padding	0; 0; 0; 0					
Ŧ	Size	25; 22					
	SizeMode	StretchImage					
	Tag						
	UseWaitCursor	False					
	Visible	True					
	Vx	2					
	Vy	2					
	WaitOnLoad	False					
					-		

Remark: Use Invalidate(); Repaint(); and Update(); methods in case of redrawing.

```
class Ball : PictureBox
  private double vx = 2;
  private double vy = 2;
 public double Vx
        ł
            get
            {
                return vx;
            }
            set
            {
                vx = value;
            }
        }
        public double Vy
        ł
            get
                return vy;
            }
            set
            {
                vy = value;
            }
        }
```

Task 2: Ball Movement



+ timer

(D) timerBallMovement

C# 2017/2018, Lecture 2

C# 2017/2018, Lecture 2

Task 3: Ball Generation

Collections (List)

• We can create empty list with:

```
List<int> listInt = new List<int>();
```

```
List<Ball> listBalls = new List<Ball>();
```

• Adding to the list:

listInt.Add(3);
listBalls.Add(ballFirst);

Collections (List)

• We can create empty list with:

List<int> listInt = new List<int>();

```
List<Ball> listBalls = new List<Ball>();
```

• Adding to the list:

listInt.Add(3);
listBalls.Add(ballFirst);

• Iterating:

```
foreach (Ball ball in listBalls)
{
}
```

Suggestions

- Create new global Ball list
- Create new temp ball (new)
- Set its properties
 - Copy from generated code of Form.Designer.cs
- Add to the balls list
- Add to the panel control list

panelPole.Controls.Add(temp);

Change location of balls using foreach loop

Collisions with walls

Pong	
	Stop Generate

Collisions with the paddle

Collisions with the paddle

Ball direction after collision with paddle \rightarrow only the place where ball hit the paddle (relative intersection point) is taken into account


```
double relativeIntersectY = (paddleX + (PADDLEWIDTH / 2)) - ballX;
double normalizedRelativeIntersectionY = (relativeIntersectY / (PADDLEWIDTH / 2));
double bounceAngle = normalizedRelativeIntersectionY * MAXBOUNCEANGLE;
ball.Vx = BALLSPEED * -Math.Sin(bounceAngle);
ball.Vy = BALLSPEED * Math.Cos(bounceAngle);
```


THE END

dr inż. Małgorzata Janik majanik@if.pw.edu.pl