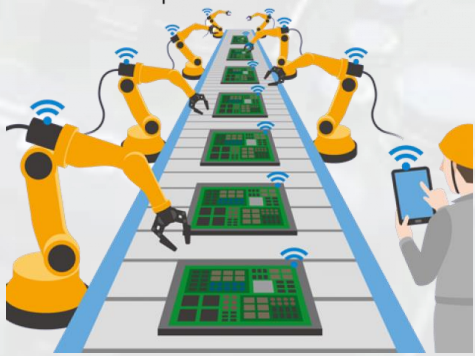
Week #03 – Classes and Objects

INC382 Capstone Project

for

Automation Engineers



Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**

Objectives

- Class and its Property and Method
- Objects Creation
- OOP and Keywords
- Class Inheritance

Practice for Knowledge Acquisition

Derek Banas, C#

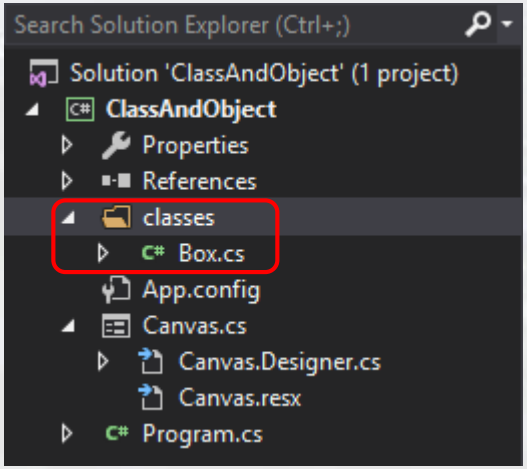
Kudvenkat, C#

When we practice something, we are involved in the deliberate repetition of a process with the intention of reaching a specific goal

Class Creation

1) Create a folder **classes** in the project and add a new class **Box** (Box.cs) into the folder. Then modify the **Box.cs** as shown

```
using System.Windows.Forms;
namespace ClassAndObject.classes
{
    public class Box        Class name, Box
    {
        public int _Width { get; set; }
        public int _Height { get; set; }
        public Panel _Body = new Panel();
                                Class Properties
        public Box()
        {
            Constructor
        }
        public Box(int size)
        {
            Constructor
        }
        public Box(int width, int height)
        {
            Constructor
        }
    }
}
```



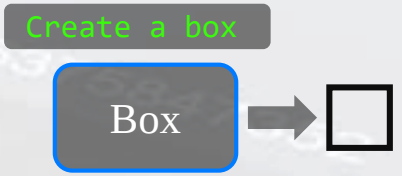
Constructors

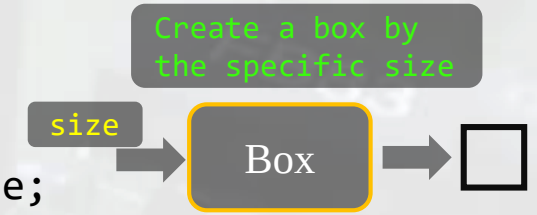
2) Modify the constructors

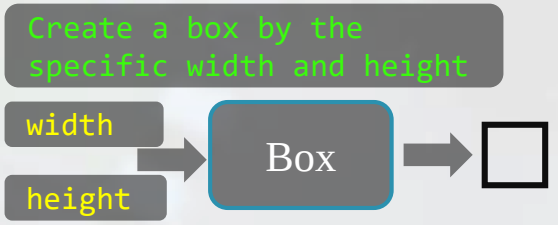
```
public Box()
{
    _Width = _Height = 20;
    _Body.Width = _Body.Height = 20;
}

public Box(int size)
{
    _Width = _Height = size;
    _Body.Width = _Body.Height = size;
}

public Box(int width, int height)
{
    _Width  = width;
    _Height = height;
    _Body.Width  = _Width;
    _Body.Height = _Height;
}
```







Object Creation

3) Go to the **MainForm.cs** and modify the code

```
using ClassAndObject.classes;
using System.Drawing;
using System.Windows.Forms;
namespace ClassAndObject
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            Box box1 = new Box();
            box1._Body.BackColor = Color.Red;

            Box box2 = new Box(50);
            box2._Body.BackColor = Color.Lime;
            box2._Body.Left = 30;

            Box box3 = new Box(20, 50);
            box3._Body.BackColor = Color.Blue;
            box3._Body.Left = 90;

            this.Controls.Add(box1._Body);
            this.Controls.Add(box2._Body);
            this.Controls.Add(box3._Body);
        }
    }
}
```

Class Inheritance

4) Create a new class **SmartBox**, and add the following lines

```
using System.Drawing;

namespace ClassAndObject.classes
{
    public class SmartBox : Box
    {
        public Color _Color { get; set; }
        public int _X { get; set; }
        public int _Y { get; set; }

        public SmartBox(int width, int height, Color color)
        {
            this._Color  = color;
            this._Width  = width;
            this._Height = height;

            this._Body.Width     = width;
            this._Body.Height    = height;
            this._Body.BackColor = color;

        }
    }
}
```

Inheritance enables new objects to take on the properties of existing objects. A class that is used as the basis for inheritance is called a superclass or base class. A class that inherits from a superclass is called a subclass or derived class.

Object Creation

5) Go to the **MainForm.cs** and modify the code

```
using ClassAndObject.classes;
using System.Drawing;
using System.Windows.Forms;
namespace ClassAndObject
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            SmartBox box1 = new SmartBox(20, 20, Color.Red);
            SmartBox box2 = new SmartBox(50, 50, Color.Lime);
            SmartBox box3 = new SmartBox(20, 50, Color.Blue);
            box2._Body.Left = 30;
            box3._Body.Left = 90;
            this.Controls.Add(box1._Body);
            this.Controls.Add(box2._Body);
            this.Controls.Add(box3._Body);
        }
    }
}
```

Methods

6) Go to the **SmartBox.cs** and add the following methods into the class

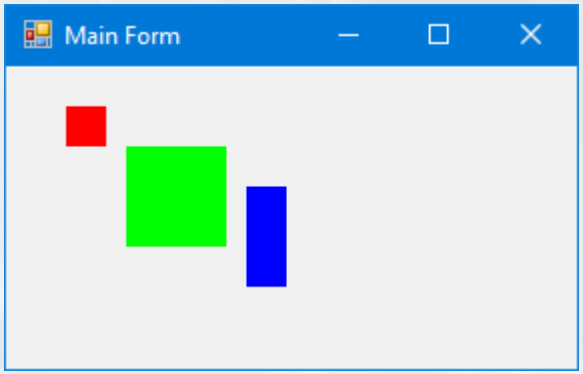
```
public void SetPositionX(int x)
{
    this._X = x;
    this._Body.Left = x;
}
public void SetPositionY(int y)
{
    this._Y = y;
    this._Body.Top = y;
}
public void SetPosition(int x, int y)
{
    this.SetPositionX(x);
    this.SetPositionY(y);
}
public int GetPositionX(int x)
{
    return this._X;
}
public int GetPositionY(int y)
{
    return this._Y;
}
```

Objects & Methods

7) Back to the **MainForm.cs** and modify the code

```
using ClassAndObject.classes;
using System.Drawing;
using System.Windows.Forms;
namespace ClassAndObject
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            SmartBox box1 = new SmartBox(20, 20, Color.Red);
            SmartBox box2 = new SmartBox(50, 50, Color.Lime);
            SmartBox box3 = new SmartBox(20, 50, Color.Blue);
            box1.SetPosition(30,  20);
            box2.SetPosition(60,  40);
            box3.SetPosition(120, 60);
            this.Controls.Add(box1._Body);
            this.Controls.Add(box2._Body);
            this.Controls.Add(box3._Body);
        }
    }
}
```



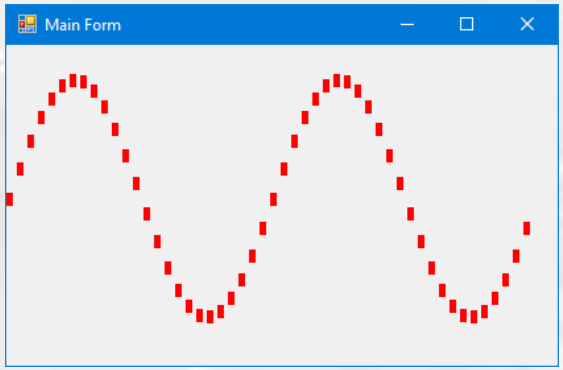
List of Objects

8) Modify the **MainForm.cs** as shown below

```
using ClassAndObject.classes;
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Windows.Forms;
namespace ClassAndObject
{
    public partial class MainForm : Form
    {
        private List<SmartBox> Boxes = new List<SmartBox>();
        public MainForm()
        {
            InitializeComponent();

            for(int i=0; i<50; i++)
            {
                SmartBox box = new SmartBox(5, 10, Color.Red);
                int x = i * 8;
                int y = (int) (this.Height/2 - Math.Sin(Math.PI *
                                                4 * i / 50) * this.Height/2);

                box.SetPosition(x, y);
                this.Controls.Add(box._Body);
                Boxes.Add(box);
            }
        }
    }
}
```

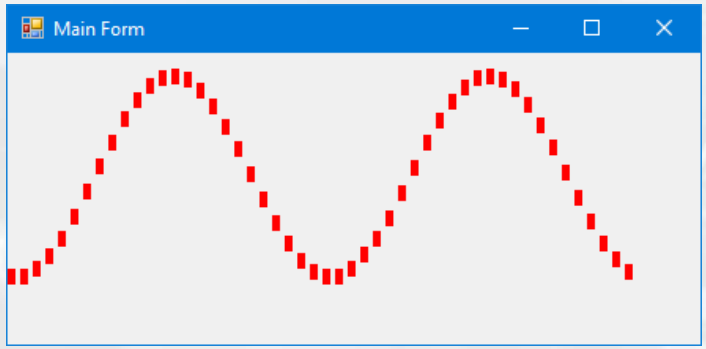


Objects and Animation

9) Modify the **MainForm.cs** as shown below

```
namespace ClassAndObject
{
    public partial class MainForm : Form
    {
        private List<SmartBox> Boxes = new List<SmartBox>();
        public MainForm()
        {
            InitializeComponent();
            for(int i=0; i<50; i++)
            {
                SmartBox box = new SmartBox(5, 10, Color.Red);
                this.Controls.Add(box._Body);
                Boxes.Add(box);
            }
            Timer timer = new Timer();
            timer.Interval = 1000 / 60;
            timer.Tick += Timer_Tick;
            timer.Enabled = true;
        }

        double alpha = 0.0;
        private void Timer_Tick(object sender, EventArgs e)
        {
            for (int i = 0; i < Boxes.Count; i++)
            {
                int x = i * 8;
                int y = (int)(this.Height / 3 - Math.Sin(alpha + Math.PI *
                                    4 * i / Boxes.Count) * this.Height / 3.5);
                Boxes[i].SetPosition(x, y);
                alpha += Math.PI / 5000;
            }
        }
    }
}
```



It’s time to learn More and More..

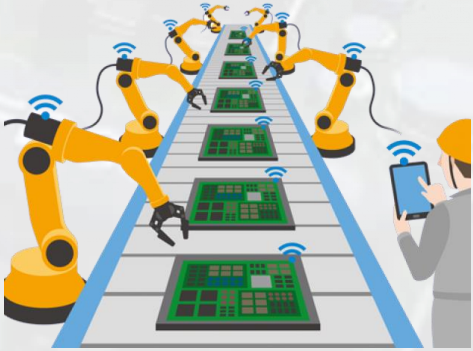
Derek Banas, C#

Kudvenkat, C#

www.c-sharpcorner.com

www.tutorialspoint.com

THANK YOU!



Santi Nuratch., Ph.D.

Embedded Computing and Control Lab. @ INC-KMUTT

santi.inc.kmutt@gmail.com, santi.nur@kmutt.ac.th

Department of Control System and Instrumentation Engineering,
King Mongkut's University of Technology Thonburi, **KMUTT**