C# Propery – get, set

using System;

namespace GetSet\_Property

{

class Rectangle

{

private double width;

private double height;

public double Area()

{

return width\*height;

}

//public Rectangle(double width, double height) // 생성자

//{

// this.width = width;

// this.height = height;

//}

// Getter

public double GetWidth()

{

return width;

}

public double GetHeight()

{

return height;

}

// Setter

public void SetWidth(double width)

{

this.width = width;

}

public void SetHeight(double height)

{

this.height = height;

}

}

class RectWithProp

{

//private double width;

//private double height;

//public RectWithProp(double width, double height) // 생성자

//{

// this.width = width;

// this.height = height;

//}

public double Area()

{

return Width \* Height;

}

public double Width { get; set; } // Width 속성

public double Height { get; set; } // Height 속성

}

class RectWithPropFull

{

private double width;

public double Width

{

get { return width; }

set { if(value > 0 ) width = value; }

}

private double height;

public double Height

{

get { return height; }

set { if (value >= 0) height = value; }

}

}

class Rect

{

public int Width { get; set; }

public int Height { get; set; }

public int Area()

{

return Width \* Height;

}

}

// expression-bodied members

class RectWithExpressionBodiedMembers

{

private double width;

public double Width

{

get => width;

set => width = value;

}

private double height;

public double Height

{

get => height;

set => height = value;

}

}

class PropertyTest

{

static void Main(string[] args)

{

Rectangle r = new Rectangle();

r.SetWidth(10.0);

r.SetHeight(10.0);

Console.WriteLine("r의 면적은 {0}", r.GetWidth() \* r.GetHeight());

RectWithProp r1 = new RectWithProp();

r1.Width = 10.0;

r1.Height = 10.0;

Console.WriteLine("r1의 면적은 {0}", r1.Width \* r1.Height);

RectWithPropFull r2 = new RectWithPropFull();

r2.Width = 10.0;

r2.Height = 10.0;

Console.WriteLine("r2의 면적은 {0}", r2.Width \* r2.Height);

RectWithPropFull r3 = new RectWithPropFull();

r2.Width = 10.0;

r2.Height = -10.0;

Console.WriteLine("r3의 면적은 {0}", r3.Width \* r3.Height);

}

}

}

상수 :

using System;

namespace Cal\_FieldsAndConstants

{

class Product

{

public string name;

public int price;

}

class MyMath

{

public static double PI = 3.14;

}

class MyCalendar

{

public const int months = 12;

public const int weeks = 52;

public const int days = 365;

public const double daysPerWeek = (double)days / (double)weeks;

public const double daysPerMonth = (double)days / (double)months;

}

class FieldsAndConstants

{

static void Main(string[] args)

{

Product p = new Product();

p.name = "시계";

p.price = 100000;

Console.WriteLine("{0} : {1:C}", p.name, p.price);

Console.WriteLine("원주율 : {0}", MyMath.PI);

Console.WriteLine("한달은 평균 {0:F3}일", MyCalendar.daysPerMonth);

}

}

}

Static :

using System;

namespace DDay\_InstanceMethod

{

struct Date

{

public int year, month, day;

public static bool IsLeapYear(int year)

{

return year % 4 == 0 && (year % 100 != 0 || year % 400 == 0);

}

static int[] days = { 0, 31, 69, 90, 120, 151, 181, 212, 243, 273, 304, 334 };

public int DayOfYear()

{

return days[month - 1] + day +

(month > 2 && IsLeapYear(year) ? 1 : 0);

}

}

class InstanceMethod

{

static void Main()

{

Date xmas = new Date();

xmas.year = 2018;

xmas.month = 12;

xmas.day = 25;

Console.WriteLine("xmas: {0}/{1}/{2}는 {3}일째 되는 날입니다",

xmas.year, xmas.month, xmas.day, xmas.DayOfYear());

if(Date.IsLeapYear(2018) == true)

Console.WriteLine("2018년은 윤년입니다");

else

Console.WriteLine("2018년은 평년입니다");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Smethod\_StaticMethod

{

class Methods

{

// using instance method

/\*

static void Main(string[] args)

{

int a = 10, b = 30, c = 20;

Methods x = new Methods();

Console.WriteLine("가장 큰 수는{0}", x.Larger(x.Larger(a, b), c));

}

private int Larger(int a, int b) // static이 아닙니다.

{

return (a >= b) ? a : b;

}

\*/

// using static method

static void Main(string[] args)

{

int a = 10, b = 30, c = 20;

Console.WriteLine("가장 큰 수는{0}", Larger(Larger(a, b), c));

}

private static int Larger(int a, int b)

{

return (a >= b) ? a : b;

}

}

}

Pyramid :

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DP\_PyramidMethod

{

class Program

{

static void Main(string[] args)

{

DrawPyramid(3);

DrawPyramid(5);

DrawPyramid(7);

}

static void DrawPyramid(int n)

{

for (int i = 1; i <= n; i++)

{

for (int j = i; j < n; j++)

Console.Write(" ");

for (int k = 1; k <= 2 \* i - 1; k++)

Console.Write("\*");

Console.WriteLine();

}

}

}

}

Search :

using System;

namespace BinarySearch

{

class Program

{

static void Main(string[] args)

{

Random r = new Random();

int[] v = new int[30];

for (int i = 0; i < 30; i++)

v[i] = r.Next(1000);

PrintArray("정렬 전", v);

// (1) 정렬

Array.Sort(v);

PrintArray("정렬 후", v);

Console.Write("=> 검색할 숫자를 입력하세요: ");

int key = int.Parse(Console.ReadLine());

int count = 0; // 비교횟수

// (2) 선형탐색

for (int i = 0; i < v.Length - 1; i++)

{

count++;

if (v[i] == key)

{

Console.WriteLine("v[{0}] = {1}", i, key);

Console.WriteLine("선형탐색의 비교횟수는 {0}회 입니다.", count);

break;

}

}

// (3) 이진탐색

count = 0;

int low = 0;

int high = v.Length - 1;

while (low <= high)

{

count++;

int mid = (low + high) / 2;

if (key == v[mid])

{

Console.WriteLine("v[{0}] = {1}", mid, key);

Console.WriteLine("이진탐색의 비교횟수는 {0}회 입니다.", count);

break;

}

else if (key > v[mid])

low = mid + 1;

else

high = mid - 1;

}

}

private static void PrintArray(string s, int[] v)

{

Console.WriteLine(s);

for (int i = 0; i < v.Length; i++)

Console.Write("{0,5}{1}", v[i], (i % 10 == 9) ? "\n" : "");

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_BubbleSort

{

class Program

{

static void Main(string[] args)

{

int[] v = { 3, 5, 2, 7, 1 };

PrintArray(v);

for (int i = 4; i > 0; i--)

{

for (int j = 0; j < i; j++)

if (v[j] > v[j + 1])

{

int t = v[j];

v[j] = v[j + 1];

v[j + 1] = t;

}

PrintArray(v);

}

}

private static void PrintArray(int[] v)

{

foreach(var i in v)

Console.Write("{0, 5}", i);

Console.WriteLine();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Diagnostics;

namespace \_BubbleSort

{

class Program

{

static void Main(string[] args)

{

Stopwatch stopwatch = new Stopwatch(); //객체 선언

stopwatch.Start(); // 시간측정 시작

int[] v = { 3, 5, 2, 7, 1 };

PrintArray(v);

for (int i = 4; i > 0; i--)

{

for (int j = 0; j < i; j++)

if (v[j] > v[j + 1])

{

int t = v[j];

v[j] = v[j + 1];

v[j + 1] = t;

}

PrintArray(v);

}

stopwatch.Stop(); //시간측정 끝

System.Console.WriteLine("time : " +

stopwatch.ElapsedMilliseconds + "ms");

}

private static void PrintArray(int[] v)

{

foreach(var i in v)

Console.Write("{0, 5}", i);

Console.WriteLine();

}

}

}

학점 계산 :

using System;

namespace Grading

{

class Program

{

static void Main(string[] args)

{

Console.Write("점수를 입력하세요: ");

int score = int.Parse(Console.ReadLine());

string grade = null;

if (score >= 90)

grade = "A";

else if (score >= 80)

grade = "B";

else if (score >= 70)

grade = "C";

else if (score >= 60)

grade = "D";

else

grade = "F";

Console.WriteLine("학점은 {0}", grade);

switch (score / 10)

{

case 10:

case 9:

grade = "A";

break;

case 8:

grade = "B";

break;

case 7:

grade = "C";

break;

case 6:

grade = "D";

break;

default:

grade = "F";

break;

}

Console.WriteLine("학점은 {0}", grade);

}

}

}