Farid Gulmammadov, 30.03.2018

1. C# da hər hansı Loop üçün müəyyən sətirə “continue” ifadəsini veririksə, istədiyimiz əmri həmin sətir üçün ekrana çıxarmır, lakin digər sətirlər üçün icra etməkdə davam edir. Məsələn,

for (int i = 0; i < 10; i++)

{

if (i == 1)

{

continue;

}

else

{

Console.WriteLine(i);

}

Burada i=1 üçün ConsoleWriteLine(i) –ni ekrana verməyəcək, lakin digərləri üçün bunu icra etməkdə davam edəcək.

“continue” əvəzinə “break” olduqda isə i=1 olduqdan sonra Loopu dayandıracaq və davam etməyəcək. Başqa sözlə, “break” blokdan çıxıb Loop prosesini dayandırmağı nəzərdə tutur.

1. “This” classda olan obyektin konkret xüsusiyyətini (ad, soyad və s.) metodun içində ifadə etmək üçün istifadə olunur. Misal üçün:

public string Name

public string Position

public Employee(string name, string position)

{

this.Name = name;

this.Position = position;

}

Burada “This” yazılmaya da bilər. Static metod class-da olan hər hansı spesifik bir obyektə/nümunəyə bağlı olmadığına görə ” this” ifadəsi bu metod daxilində ifadə oluna bilməz. Başqa sözlə, static metod classdan çağırılır və hər hansı spesifik obyektə bağlı deyil.

1. String və string: texniki olaraq hər ikisi string datatipini nəzərdə tutsa da, təyinatı və yerindən asılı olaraq böyük və kiçiklə yazıla bilər. Məsələn, hər hansı obyektə müraciət ediriksə,

string object = “book” string kiçiklə yazılmış olur.

Class-ı çağırırıqsa, o zaman

string place = String.Format(“Take”, object) String böyüklə yazılmış olur.

string ifadəsi bir növ System.String üçün psevdonimdi. Digər datatiplər də buna analojidir.

bool: System.Boolean

byte: System.Byte

sbyte: System.SByte

short: System.Int16

int: System.Int32

long: System.Int64

float: System.Single

double: System.Double

decimal: System.Decimal

char: System.Char

1. Equals(),GetHashCode(),GetType(),ToString() metodları object classının baza metodlarıdır.

Equals() – iki obyekti müqayisə edir və onlar bərabər olduqda TRUE kimi qəbul edir.

GetHashCode() – obyektlərə fərdi kod/nömrə vermək üçün istifadə olunur. Hash Table yaratmaq üçün istifadə oluna bilər.

GetType() – Obyektin tipini müəyyən edən metoddur. Məsələn, aşağıdakı misalda

class Program

{

static void Main()

{

A a1 = new A();

A a2 = new B();

A a3 = new C();

Console.WriteLine(a1.GetType());

Console.WriteLine(a2.GetType());

Console.WriteLine(a3.GetType());

}

}

Ekranda nəticə: A

B

C

ToString() - Classda obyekti və ya obyektin xüsusiyyətlərini stringə çevirir Məsələn,

class Notebook

{

public int HDD

public int RAM

public Notebook(int\_hdd, int\_ram)

{

HDD = \_hd;

RAM=\_ram

}

Public override string ToString()

{

return String.Format (“HD = {250}, RAM = {4}”, HD, RAM);

}

1. “Public” keywordu ilə metodu obyektdən çağırmaq olar, lakin “static” keywordu ilə metodu yalnız classdan çağırmaq olur.
2. İnheritance nümunə:

class İt

{

public string Ayaq

public string Goz

public string Burun

}

class Qoyun : İt

{

public string Yun

}

class DağKecisi: Qoyun

{

public string Buynuz

}

7. Method overloading : eyni adda iki metodun fərqli signature ilə yazılmasını nəzərdə tutur. Metod overloading-e yaranmış vəziyyətdən asılı olaraq ehtiyac yarana bilər. Məsələn,

class Total

{

public static int Sumof(int number1, int number2)

{

return Sumof(number1, number2, 0);

}

public static int Sumof(int number1, int number2, int number3)

{

return number1 + number2 + number3;

}

}

İki versiyada sumof metodları göstərilmişdir. Metodlarin adlari eyni olsa da, birinci sumof-da iki, ikincidə isə üç element vardır. İkinci sumof-da cəm əməliyyatı həyata keçirilir. Əgər bizə üç yox, iki elementin cəmi lazım olarsa, ikinci sumof metodu çağırıb üçüncü element yerinə sadəcə 0 yaza bilərik.

8. Variable- müxtəlif datatiplərdə yazılan adlardır ki, datatipdən asılı olaraq yaddaşda müvafiq yer tutur. Property-nin sintaksı variable-la oxşar olsa da, onda accessor-lardan (get, set) istifadə olunur ki, variable-dan onları fərqləndirir. Bundan başqa property ilə class-da private kimi verilən obyektin adını (və ya başqa xüsusiyyəti üzrə adı) dəyişə bilərik, variable-la isə bu mümkün deyil.

9. Visual Studioya alternativlər:

PrimalScript (149 usd)

PrimalCode (249 USD)

SharpDevelop (Free)

Antechinus (35 USD)

CodeWright (249 USD)

SourceInsight (249 USD)

Visual SlickEdit (99 USD)

Improve C# plugin for Eclipse (Free)