# \_15\_03\_2018\_oop\_concepts

# Keywords

- Encapsulation

- Inheritance

- Polymorphism

- Abstraction

- access modifiers / public,private sohbetleri

- Base class

- Derived class

- Data hiding

- Fields

- Class and Object members

- Custom (user defined) class, Built in calss

- Return

# Qaynaqlar

- https://www.slideshare.net/DonchoMinkov/objectoriented-programmingwith-c

Bu slideshow da genis olaraq ve step by step izah olunub OOP Conceptleri. Ciddi analiz edilse ve arasdirilsa bunun uzerine gozel basa dusulebiler

# Asagidaki suallara etrafli cavab yazmaginiz teleb olunur

1. In how many ways you can pass parameters to a method?

2. Can you return multiple values from a function in C#? (How?)

3. What is namespace in C#?

4. What is the purpose of using statement in C#?

5. What are value types in C#

6. What are reference types in C#?

7. What is encapsulation?

8. Can you create a function in C# which can accept varying number of arguments?

9. What is a structure in C#?

10. What are the differences between a class and structure.

11. What is inheritance?

12. Is multiple inheritance supported in C#?

13. How to inherit a class in C#?

14. What is the difference between “continue” and “break” statements in C#?

15. Can “this” be used within a static method?.

1)

1. Value parameters.

2. Reference parameters.

3.Output parameters.

2) ref,out,class,struct,Tuple - Bunlar ile Methoddan coxlu sayda deyer dondurmey olar.

struct ile

struct Result

{

public int topla;

public int vur;

}

static void Main(string[] args)

{

int a = 10;

int b = 20;

var ntc = IkiSayiToplaVur(a, b);

Console.WriteLine(ntc.topla);

Console.WriteLine(ntc.vur);

}

private static Sonuc IkiSayiToplaVur(int a, int b)

{

var ntc = new Result

{

vur = a \* b,

topla = a + b

};

return ntc;

}

3)C# -da Namespace Nedir?

Eyni Vezifeye Qulluq Eden Ozellikleri , Sinifleri ve Fonksiyonlari Bir Roof Altinda Toplayarken , Bu roof - a Verdiymiz Ad-a Ad Sahesi(Namespace) - Deyilir. Bir Namespace icerisine bir - biriyle Elaqeli Sinif ve Fonksiyonlar Qoymaqa Diqqet Etmeliyik.

Meselen:

Yeni Console Aciriq. Lahiyeye Ad Veririk.C# sistemi Lahiyeye Verdiyimiz adi Goturerek Automatic Olaraq Hemin Lahiye Adinda Bir (Namespace) - yaradir. Biz Tek Console Appda bir nece Class Declare Etsek ve Access Modifier olaraq Public Teyin Etsek Bir Clasa Diger Class Icinde Reach Ede Bilerik Bilerik Bunun Sebebi C# sisteminin Automatic Olaraq Yaratdiqi (Namespace-dir) - ve Biz Her Defe Yeni Class Yaradanda Eyni Roof (Namespace) - Altinda Yaratdiqmiz Ucun Class Daxilinde Class - a Reach Ede Bilirik.

Amma Biz Colden Lahiyeye bir Class Daxil Etsek ve ya Ferqli Bir Namespace Altinda Bir Class Declare Etsek bu Declare Etdiymiz Classa Bawqa Class icinden Reach Ede Bilmerik Bunun Sebebi Ferqli Namespacelerin Olmasidir. Reach Etmek Ucun (using.namespace\_name) So Declare Etdiymiz Namespace - i Yuxarda Cagirmaq Lazimdir.

Nested Namespacelerde Declare Etmek Olar. Meselen Bir (namespace firstNs) namespace - miz var ve Onun Icinde (secondNs) - Adli (namespace - miz var) bu (namespaceni) - use Etmek Ucun Yuxarda (using firstNs.secondNs) - ilk (namespace) - nin ad (.) - ikinci (namespacenin) - adi.

Duwunule Biler ki Bele Yazdiqda Program Her iki (namespace) - i Gorur amma Bele Yazdiqda Sistem Yalniz ilk (namespace) - nin icindeki ikinci (namespace) - i Gorur. Error Olmamasi Ucun Use Edeceyiniz (namespace) - nin Adi Yazilmalidir.

5)C# - da Value Tipler Nedir?

Bir Data Tipi Oz yaddaw sahesi icerisinde bir deyer Dawiyirsa bu Value Type - dir. Bunu Menasi Bu Data Tiplerinin Deyiwenleri BirBawa Olaraq Ozlerinde Deyerler Saxlayirlar.

Meselen Integer Tamsayi Deyiwenine Baxaq:

int i = 100;

Sistem 100 Deyerini (i) - ucun Ayrilmiw Yaddaw Sahesinde Saxlayir

Value Tipler Eslinde Temel Olaraq Istifade Etdiyimiz Deyiwken tanımlamalarıdır.

int eded = 1;

bool eded2 = false;

float eded3 = 3.5;

Bele Tanimlamalar Value Type Sinfine Daxildir.

Awagidaki Data Type - lerin Hamisi Value Typlardir

bool

byte

char

decimal

double

enum

float

int

long

sbyte

short

struct

uint

ulong

ushort

6)C# - da Refrence Type Nedir

Value Type - lardan Farkli Olaraq Reference Typle - lar Deyerlerini Birbawa Saxlamir. Bunun Evezine Deyerin Saxlandigi Adressi Ozunde Saxlayir. Bawqa Bir Deyiwle, Bir Referans Tipi Datalari Saxlayan Bawqa Bir Yaddaw Sahesine Bir Iwaretci icerir.

Meselen Awagidaki String Deyiwenine Baxaq:

string s = "Hello Cav";

Bu numunede Sistem (s) - Deyiweni Ucun Yaddawda Rastgele(Random) - bir Konum Choose Eliyir. Bir Referans Deyiseninin Deyeri Actual Data Deyerinin Yaddaw Konumudur.

Meselen:

Sistem (s) - deyeri Ucun Yaddawda random olaraq (0x803200) - yer Ayirir ve bu (s) - deyiwenin Esil Deyeri (0x803200) - da yaranan (0x600000) - konumudur.

Wekil ile izah - http://www.tutorialsteacher.com/csharp/csharp-value-type-and-reference-type

Another Example:

Bir Class Create Edirem ve Compilere - Deyirem ki Men bu Sinifden Bir Obyekt Yaradacam ve Obyekt Sinife aid Ozunde Bir Ozellik Saxlayacaq. So, Onu Referans Verirem ki Get Yaddawda Ona Gore Davran.

ClassTest data = new ClassTest();

Burada Eslinde (data) - Refernce Typedir.

Awagidaki Data Tipleri Reference Tiplerdir

String

All arrays, even if their elements are value types

Class

Delegates

7)C# - Kapsulleme Nedir?

class Person

{

string Name;

string wvNo;

int examResult;

}

Bu Field-lerime Bawqa Bir Class - dan Reach Ede Bilmek Ucun Access Modofierini (public) - Etmek Lazimdi. Bu Istifade Wekli Yanliwdir, Bele Istifade Zamani Field - lerimizi Diger Siniflere Qarwi Savunmasiz Buraxmiw Oluruq ve Hemcinin Filed-lerimize Qaydasiz Deyer Teyin Etme Riski Ortaya Cixir Meselen(wexsiyyet vesiqesi nomrei 11 ededded ibret olmalidir - public istifadesinde 12 Yaza Bilerem ve ya examResult 0 - 100 arasi olmalidir public istifadesinde 130 - yaza bilerem) Encapsulation Bunlarin Qarwisini Alir. Class Icinde Declare Edilen Field-lerimizin Default Access Modifieri-nin Private Olmasinin Sebebi Budur.

Bu Fiedler - imizi Private Olaraq Declare Ede Bilmeliyik, amma Private Declare Etdiyim Bu Fieldleri public" Declare edilmiw bir aracı terefinden kontrol edebileceyim obyektlere ehtiyacimiz var. Bu Obyektlerede Property Deyilir. Her Filed Ucun Property Declare Etmeliyem ve Buna Encapsulation Deyilir. Ne Ucun Encapsulation Use Edirik.

1)Kontrolsuz Data Giriwinin Qarwisini Alir ve Bunu Etmek Ucun (get,set) - Metodlardan Istifade Edirik.(get,set) - metodlari ile hemcinin Obyektlerimi Sadece Oxunabilir ve ya Yazilabilir Etmek Olur. Filedlerin Propertileri Awagidaki Kimi Teyin Olunur.

class Person

{

private string Name;

private ulong wvNo;

private int examResult;

public string name

{

get

{

return Name;

}

set

{

Name = value;

}

}

public ulong wvno

{

get

{

return wvNo;

}

set

{

if (value.ToString().Length == 11)

wvNo = value;

else

Console.WriteLine("Xeta! wv Seriya Nomresi 11 eded Olmalıdır.");

}

}

public int examresult

{

get

{

return examResult;

}

set

{

if (value > 0 && value <= 100)

examResult = value;

else

Console.WriteLine("Xeta! Imtahan Neticesi 1-100 Aralığında Olmalıdır. ");

}

}

}

class Program

{

static void Main(string[] args)

{

Person p1 = new Person();

p1.name = "Serdar Yılmaz";// Name Propertisinin SET metodu Iwe Duwdu.

p1.wvno = 14548617718;// wvNo Propertisinin SET metodu Iwe Duwdu.

p1.examresult = 110; // examResult Propertisinin SET metodu Iwe Duwdu.

Name, wvNo ve examResult Propertilerinin GET metodu Iwe Duwdu.

Console.WriteLine("Ad:{0} - wexsiyyet vesiqesi No:{1} - imatahan Neticesi:{2}", p1.name, p1.wvno, p1.examresult);

}

}

Sinifin Name , wvNo , examResult Fieldlerini Diger Sinfilerden Reach Olunmasinin Qarwisini Aldiq(private ile) ve Bu Fieldlere (name,wvno,examresult) - Propertileri ile kontrol eledik

Get Metodu - Bir Propertinin Deyerini Oxumaq Istediymizde O Propertiye Aid olan (GET) - metodu iwe duwur.

Set Metodu - Bir Propertiye Deyer Atmaq Isteyende Hemin Propertiye Aid (SET) - meodu iwe duwur. Atilan Deyere Set metodunun Icindeki (value) - keywordu ile Reach olunur (value) - keywordunun Qabaqcadan Teyin Edilmiw Bir Deyeri Yoxdur Sadece Metod iwe duwende anda Hansi Tipde Deyer Teyin Edilse (value) - O tipde Olacaq.

Take Note: Bir Propertinin sadece GET metodunu Declare ederek Propertini sadece OxunaBilir Hala Getire Bilerik.

Access Modifier - lerin Istifadesi

Get ve Set Metodlarinda Access Modifierler - den Istifade Ederek Access - in Serhedlerini Mueyye Ede Gostere Bilerik.

private string Name;

public string name

{

get

{

return AdSoyad;

}

private set

{

AdSoyad = value;

}

}

Yuxarda Name - ni Private Eledik buna Gorede Bawqa Klasdan name Propertisinin Deyeri Ancaq Oxuna Biler Her Hansi Bir Deyer Atila Bilmez(set metodunada yalniz Clasin Icinden Reach Elemek Mumkundur). Eger (set) - metodunu Private Olaraq Bildirmesydik Automatic Olaraq name Propertisi public oldugu ucun (set) metoduda public olacaqdi

Take Note: GET ve SET metodları ucun herhansi bir Access Modifier bildirilmezse, Default olaraq Propertinin Access Modifieri - ni alır.

9)Struct - bir Qrup Datanin bir arada tutulması ucun Yaradilan yeni unit - e (struct) deyilir. (struct) - value typedir ve value type oldugu ucun yaddawin (stack) - hissesinde yerlewir.Qurucu Declare Etmek Ucun "struct" - keywordundan istifade edilir.Bir "structun" umumi Declare Olunma Wekli Awagidaki Kimidir.

struct Student

{

public int nomre;

public string ad, soyad;

}

Ogrenci wagird = new Ogrenci();

Console.Write("Nomre yazin : ");

wagird.nomre = Convert.ToInt16(Console.ReadLine().ToString());

Console.Write("Ad Giriniz :");

wagird.ad = Console.ReadLine().ToString();

Console.Write("Soyad Giriniz :");

wagird.soyad = Console.ReadLine().ToString();

Console.WriteLine("--------------------");

Console.WriteLine("Yeni wagird Bilgileri");

Console.WriteLine("Okul Nomre : {0}", wagird.nomre);

Console.WriteLine("Adı : {0}", wagird.ad);

Console.WriteLine("Soyadı : {0}", wagird.soyad);

Console.ReadKey();

}

}

10)C# -da Class ve Struct Arasindaki ferq nedir?

1)struct Value Type Oldugu Ucun Yaddawin Stack Hissesinde Saxlanilir amma Class Reference Type Oldugu Ucun Yaddawin Heap Hissesinde Yer Tutur.

2)Classlar - da varsayilan(default) Qurucular(Constructors) yazila bilir, struct da ise bu mumkun Deyil amma struct-da parametirli Constructorlar Declare Edde Bilerik. Diqqet Edilmesi Lazim Olan Noqte ise structda Constructir Yazands ilk Deyer Teyin Edilmesi Olmalidir. Classlar-da buna ehtiyyac yoxdur cunku Compiler - Automatic olaraq bizim ucun elave Edir.

Meselen:

class Rectangle

{

public int width;

public int height;

public Rectangle()//Default Qurucu metot

{

width = 4;

height = 5;

}

public int GetArea()

{

int area;

alan = width \* height;

return area;

}

}

class Program

{

static void Main(string[] args)

{

Rectangle rect = new Rectangle();

Console.WriteLine(rect.GetArea());

Console.ReadKey();

}

}

Yuxardaki Numunede Rectangle Adinda Class Teyin Etdik ve Default Constructor Komeyi ile (width,height) - deyiwenlerine varsayilan bir deyer elave etdik. Program Icerisinde Bu Sinifden bir Obyekt Yaradanda hemin deyerler Varsayilan Olaraq Elave Edilecekdir.Diqqet yetirsek GetAre() metodumuza bir Deyer Elavesi Etmediymiz halda program doğru bir şekilde iwlemiwdir.

Indi ise Bunu Struct-da Yoxlayaq:

struct Rectangle

{

public int width;

public int height;

public Rectangle()

{

width = 4;

height = 5;

}

public int GetArea()

{

int area;

alan = width \* height;

return area;

}

}

class Program

{

static void Main(string[] args)

{

Rectangle rect = new Rectangle();

Console.WriteLine("Alan Hesabı : {0}", rect.GetArea());

Console.ReadKey();

}

}

struc - ile etdiymizde bele bir error alacaqiq //structs cannot contain explicit parameterless constructors

Bunun Sebebi Construclarda Parametr Olmamasidir

Parameter ile Yazdigimizda

struct Rectangle

{

public int width;

public int height;

public Rectangle(int m\_width, int m\_height)

{

width = m\_width;

height = m\_height;

}

public int GetArea()

{

int area;

area = width \* height;

return area;

}

}

class Program

{

static void Main(string[] args)

{

Rectangle rect = new Rectangle(4,5);

Console.WriteLine("Alan Hesabı : {0}", rect.GetArea());

Console.ReadKey();

}

}

struct- da yazdiqmiz Constructorda parameter oldugu ve ilkin deyerler elave olundugu ucun error almadan kod iwledi

11)C# Inheritance Nedir?

Bir Class - dan Yeni Class Yaratmaq Emliyyatina Inhertance Deyilir(miras alma). Inheritance Istifadesinde Hansi Sinfden Inheritance Alinirsa O, Sinife Aid Propertiler Metodlar Yeni Sinfdede Movcud Olur. Isteye Bagli Olaraq Yeni Sinife Yeni Propertiler veya Metodlar - da Elave Etmek Olar.

class Person

{

public string adi,soyadi;

}

class personInfo:Person //Person Sinfinden Inheritance yoluyla infoPerson Sinfi Creat Edildi.

{

public string job,mail;

}

class AnaProgram

{

static void Main(string[] args)

{

personInfo pers=new personInfo();

pers.adi="Cav";

pers.soyadi="Ekberov";

pers.job="Programmer";

pers.mail="acavidan.p102@code.edu.az";

Console.ReadKey();

}

}

12)C# - da multiple inheritance desteklnemir Cunku Her Sinif sadece Bir Sinifden Toreye Biler. Deyesen interface ile ele bir wey elemek olurdu.

13) "Main Class:Derived Class"

14) Break ve Continue Dongulerde Istifade Edilen Keywordlardir. Break keywordu Donguden cixmaq Ucun Istifade Olunur:

Meselen:

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

{

if (i == 15)

{

break;

}

Console.WriteLine(i); // 1 , 2 , 3 ,4 , 5 , 6 , 7 , 8 , 9 , 11 , 12 , 13 , 14

}

Dongumuz 15 - e Geldiyinde Automatic Olaraq Dongu Dayandirilacaq 15 - i Gostermeyecek;

Take Note: (break) - keywordu for,while,do-while,foreach,switch - donguleri ile istifade oluna biler.

Continue - Dongude bir Sonraki Addima Jump Elemek Ucun Istifade Olunur.

for (int i = 1; i <= 20; i++)

{

//kontrol değişkenimiz(i) 10 oldugunda Dongumuzu Qir

if (sayac == 10)

{

continue; //10'u Jump Eliyir ve Dongumuzun Qalani ile davam edir

}

Console.Write("{0} ", i);// 1 , 2 , 3 ,4 , 5 , 6 , 7 , 8 , 9 , 11 , 12 , 13 , 14 , 15 , 16 , 17 , 18 , 19 , 20

}

Console.ReadLine();

}

10 - nu Gostermeyecek

Take Note: continue for,while,do-while,foreach - ile istifade Olunsada (switch) - ile istifade Olunmur

Aralarindaki Ferq - continue = for,while,do-while,foreach: break = for,while,do-while,foreach,switch (continue - jump eliyir break - Qirir)

15)Static Method Teyin Olunanda BirBawa Class Istifadesi ile Call Olundugu Ucun Classin Orneyini(instance of class)(object) Yaratmaq Olmur. this keywordu da Obyekte Referans Etdiyi Ucun Staticde this Istifade Oluna Bilmez. Qisa Olaraq: static Clasa Aiddi Obyekte Yox ve (this) de Obyekte referans Verir.