30.09.2021

Design Patternler Ile Çözüm Üretmek

Bora Kaşmer (Head Of Software Architect) VBT

http://www.borakasmer.com(tr)
https://borakasmer.medium.com(en)
https://github.com/borakasmer

MVP Microsoft® Most Valuable Professional

YENI BIR TIP GELDIĞI ZAMAN, TÜM KODU TEKRARDAN MI DEĞIŞTIRECEĞIZ ?

```
public void ProcessCalculation(ProcessType type)
{
    try
    {

        .
        .
        ....ToDoSomething()

        if (type == ProcessType.Crm)
        {
            CallCrm();
        }
        else if (type == ProcessType.Finance)
        {
            CallFinance();
        }
        else if (type == ProcessType.Service)
        {
            CallService();
        }
    }
    catch
    {
        if (type == ProcessType.Crm)
        {
           BackCrm();
        }
        else if (type == ProcessType.Finance)
        {
            BackFinance();
        }
        else if (type == ProcessType.Service)
        {
            BackService();
        }
    }
}
```

CHAIN OF RESPONSIBILITY

```
interface IProcess
{
    3 references
    string nextMethodName { get; set; }
    3 references
    string backMethodName { get; set; }
    2 references
    void NextMethod();
    2 references
    void BackMethod();
}
```

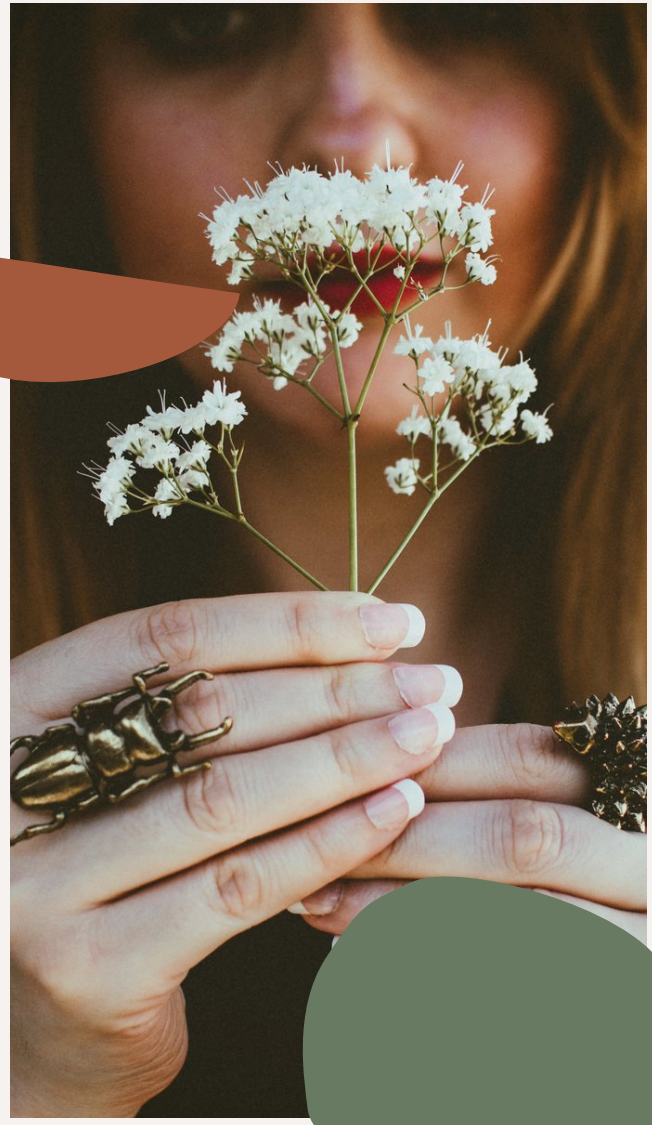
```
public class Process : IProcess
{
    3 references
    public string nextMethodName { get; set; }
    3 references
    public string backMethodName { get; set; }
    1 reference
    public Process(string _nextMethodName, string _backMethodName)
    {
        this.nextMethodName = _nextMethodName;
        this.backMethodName = _backMethodName;
    }
    2 references
    public void NextMethod()
    {
        Type thisType = this.GetType();
        MethodInfo theMethod = thisType.GetMethod(this.nextMethodName);
        theMethod.Invoke(this, null);
    }
    2 references
    public void BackMethod()
    {
        Type thisType = this.GetType();
        MethodInfo theMethod = thisType.GetMethod(this.backMethodName);
        theMethod.Invoke(this, null);
    }
```

```
public void DoServiceProcess()
{
    Process process = new Process("CallService", "BackService");
    process.ProcessCalculation();
}
```

```
using System;
using System.Reflection;
```

```
public void ProcessCalculation()
{
    try
    {
        /*
                    .
                    .
                    ....ToDoSomething()
        */
        NextMethod();

    }
    catch
    {
        BackMethod();
    }
}
```



- WeServisi ile Mobile Servis Birbirine karısmıs.
- Her Sınıf için Ortak Bir Interface Belirlenir!
- Farklı Özellikleri Olan Sınıflara Yeni Bir Interface Tanımlanır.
- Böylece Her Sınıfı Özellestirmis ve Kod Kalabalıgından Kurtulunmus olunur.

S.O.L.I.D
INTERFACE SEGREGATION

```
interface IService
{
    0 references
    Boolean CheckToken();
    1 reference
    String GetUserId();
 /* String GetMobileDeviceId();
    Boolean CheckMobileVersion();
    String GetMobilePlatform(); */
}
1 reference
interface IMobileService :IService
{
    0 references
    String GetMobileDeviceId();
    0 references
    Boolean CheckMobileVersion();
    0 references
    String GetMobilePlatform();
}
```

```
public class WebService : IService
{
    0 references
    public bool CheckToken()…
    1 reference
    public string GetUserId()…
    /* public bool CheckMobileVersion()…
    /* public string GetMobileDeviceId()…
}

2 references
public class MobileService : IMobileService
{
    0 references
    public bool CheckToken()…
    1 reference
    public string GetUserId()…
    0 references
    public string GetMobileDeviceId()…
    0 references
    public string GetMobilePlatform()…
    0 references
    public bool CheckMobileVersion()…
}
```

WEBSERVICE

```
static void Main(string[] args)
{
    WebService service = new WebService();
    MobileService mobileService = new MobileService();
    Console.WriteLine($"UserID: {service.}");
    Console.ReadLine();
}
```

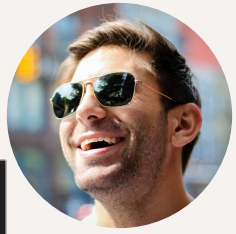
CheckToken
Equals
GetHashCode
GetType
GetUserId string GetUserId()
ToString

MOBILESERVICE

```
static void Main(string[] args)
{
    WebService service = new WebService();
    MobileService mobileService = new MobileService();
    Console.WriteLine($"UserID: {service.GetUserId()}");
    Console.WriteLine($"MobileDeviceID: {mobileService.}");
    Console.ReadLine();
}
```

CheckMobileVersion
CheckToken
Equals
GetHashCode
GetMobileDeviceId string GetMobileDeviceId()
GetMobilePlatform
GetType
GetUserId
ToString

IŞ AKIŞINDA YENI BIR X MAIL ATMA YÖNTEMI GELSE NE YAPACAGIZ ?



```
public enum MailType
{
    1 reference
    HumanResource,
    2 references
    IT,
    1 reference
    Manager,
    1 reference
    Worker
}
```

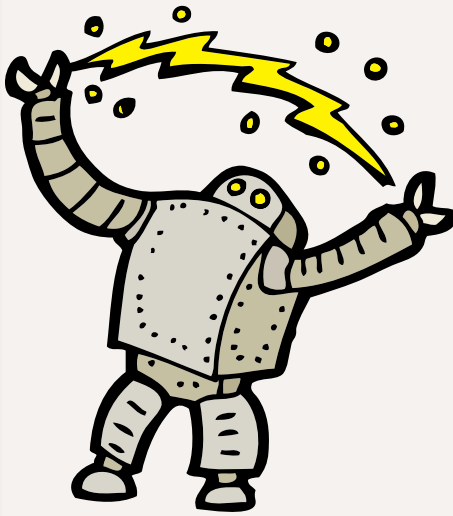
```
void SendMail(MailType type, string body, string cc)
{
    switch (type)
    {
        case MailType.HumanResource:
            {
                Console.WriteLine($"HumanResource Mail Send : {body}");
                break;
            }
        case MailType.IT:
            {
                Console.WriteLine($"IT Mail Send : {body}");
                break;
            }
        case MailType.Manager:
            {
                Console.WriteLine($"Manager Mail Send : {body}");
                break;
            }
        case MailType.Worker:
            {
                Console.WriteLine($"Worker Mail Send : {body}");
                break;
            }
    }
}

SendMail(MailType.IT, "Publish Yapalım", "bora@borakasmer.com");
Console.ReadLine();
```

```
public interface IMail
{
    1 reference
    void SendMail(string body, string cc);
}
```

```
class HumanResource : IMail
{
    1 reference
    public void SendMail(string body, string cc)
    {
        Console.WriteLine($"HumanResource Mail Send : {body}");
    }
}
1 reference
class IT : IMail
{
    1 reference
    public void SendMail(string body, string cc)
    {
        Console.WriteLine($"IT Mail Send : {body}");
    }
}
0 references
class Manager : IMail
{
    1 reference
    public void SendMail(string body, string cc)
    {
        Console.WriteLine($"Manager Mail Send : {body}");
    }
}
0 references
class Worker : IMail
{
    1 reference
    public void SendMail(string body, string cc)
    {
        Console.WriteLine($"Worker Mail Send : {body}");
    }
}
```

```
public class Postman
{
    2 references
    IMail _mail = null;
    1 reference
    public Postman(IMail mail) => _mail = mail;
    1 reference
    public void SendMail(string body, string cc)
    {
        _mail.SendMail(body, cc);
    }
}
```



```
static void Main(string[] args)
{
    Postman postman = new Postman(new IT());
    postman.SendMail("Publish Yapalım", "bora@borakasmer.com");
    Console.ReadLine();
}
```

```
Boras-MBP:strategy borakasmer$ dotnet run
IT Mail Send : Publish Yapalım
```

```
abstract class VehiclePrototype
{
    3 references
    public string ChassisNumber { get; set; }
}

2 references
class Cbr650R : VehiclePrototype
{
    1 reference
    public int WarrantyYear { get; set; }
    1 reference
    public int Km { get; set; }
    1 reference
    public double Weight { get; set; }
    1 reference
    public int Engine { get; set; }
    0 references
    public string Color { get; set; }
    1 reference
    public int Year { get; set; }
    0 references
    public double Price { get; set; }

}
```

```
class AfricaTwin : VehiclePrototype
{
    1 reference
    public int WarrantyYear { get; set; }
    1 reference
    public int Km { get; set; }
    1 reference
    public double Weight { get; set; }
    1 reference
    public double Engine { get; set; }
    0 references
    public string Color { get; set; }
    1 reference
    public int Year { get; set; }
    0 references
    public double Price { get; set; }
}
```

```
public static List<VehiclePrototype> VehicleList { get; set; }
0 references
static void Main(string[] args)
{
    VehicleList = new List<VehiclePrototype>();
    for (int i = 0; i < 10; i++)
    {
        Cbr650R motor = new Cbr650R();
        motor.WarrantyYear = 4;
        motor.Km = 0;
        motor.Weight = 207;
        motor.Engine = 650;
        motor.Year = DateTime.Today.Year;
        motor.ChassisNumber = i.ToString() + "Cbr650";

        VehicleList.Add(motor);
    }
    for (int i = 0; i < 10; i++)
    {
        AfricaTwin motor = new AfricaTwin();
        motor.WarrantyYear = 3;
        motor.Km = 0;
        motor.Weight = 245;
        motor.Engine = 1000;
        motor.Year = DateTime.Today.Year;
        motor.ChassisNumber = i.ToString() + "AfricaTwin";

        VehicleList.Add(motor);
    }

    foreach (var motor in VehicleList)
    {
        Console.WriteLine("Motor ChassisNumber: " + motor.ChassisNumber);
    }
```

BU KOD'UN CIDDI PERFORMANS PROBLEMI YOK MU! 1000 KIŞI AYNI ANDA OYNAMAYA KALKSA, MOTORLARIN HAREKET ETMESI BILE MUCIZE OLABILIR...

PROTOTYPE DESIGN PATTERN

```
abstract class VehiclePrototype
{
    2 references
    public abstract VehiclePrototype Clone();
    3 references
    public string ChassisNumber { get; set; }
}

4 references
class Cbr650R : VehiclePrototype
{
    1 reference
    public Cbr650R()
    {
        WarrantyYear = 4;
        Km = 0;
        Weight = 207;
        Engine = 650;
        Year = DateTime.Today.Year;
    }
    2 references
    public int WarrantyYear { get; set; }
    2 references
    public int Km { get; set; }
    2 references
    public double Weight { get; set; }
    2 references
    public int Engine { get; set; }
    0 references
    public string Color { get; set; }
    2 references
    public int Year { get; set; }
    0 references
    public double Price { get; set; }
    2 references
    public override VehiclePrototype Clone()
    {
        return this.MemberwiseClone() as VehiclePrototype;
    }

}
```

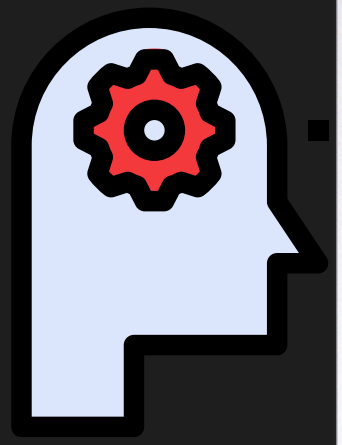
```
class AfricaTwin : VehiclePrototype
{
    1 reference
    public AfricaTwin()
    {
        WarrantyYear = 3;
        Km = 0;
        Weight = 245;
        Engine = 1000;
        Year = DateTime.Today.Year;
    }
    2 references
    public int WarrantyYear { get; set; }
    2 references
    public int Km { get; set; }
    2 references
    public double Weight { get; set; }
    2 references
    public double Engine { get; set; }
    0 references
    public string Color { get; set; }
    2 references
    public int Year { get; set; }
    0 references
    public double Price { get; set; }
    2 references
    public override VehiclePrototype Clone()
    {
        return this.MemberwiseClone() as VehiclePrototype;
    }
}
```

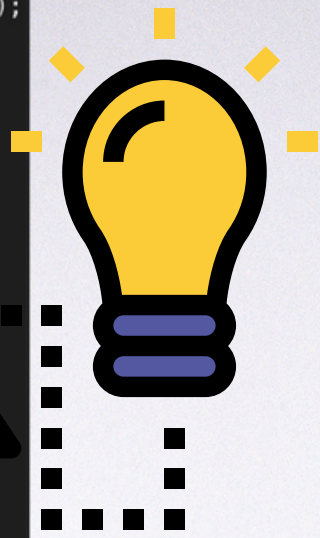
```
VehicleList = new List<VehiclePrototype>();
for (int i = 0; i < 10; i++)
{
    var index = VehicleList.FindIndex(f => f.GetType() == typeof(Cbr650R));
    Cbr650R motor = index == -1 ? new Cbr650R() : (Cbr650R)VehicleList[index].Clone();
    if (index == -1)
    {
        motor.WarrantyYear = 4;
        motor.Km = 0;
        motor.Weight = 207;
        motor.Engine = 650;
    }
    motor.Year = DateTime.Today.Year;
    motor.ChassisNumber = i.ToString() + "Cbr650";

    VehicleList.Add(motor);
}
for (int i = 0; i < 10; i++)
{
    var index2 = VehicleList.FindIndex(f => f.GetType() == typeof(AfricaTwin));
    AfricaTwin motor = index2 == -1 ? new AfricaTwin() : (AfricaTwin)VehicleList[index2].Clone();
    if (index2 == -1)
    {
        motor.WarrantyYear = 3;
        motor.Km = 0;
        motor.Weight = 245;
        motor.Engine = 1000;
    }
    motor.Year = DateTime.Today.Year;
    motor.ChassisNumber = i.ToString() + "AfricaTwin";

    VehicleList.Add(motor);
}

foreach (var motor in VehicleList)
{
    Console.WriteLine("Motor ChassisNumber: " + motor.ChassisNumber);
}
Console.ReadLine();
```





```
public static void Main(string[] args)
{
    string Description1 = "Futbol maçlarını çok sevsem de, genelde izlemiyorum";
    string Description2 = "Bebeğimin sağlığı için anne sütüne çok önem veriyorum.";
    string Description3 = "Spor'a giderken arabayı vurdum.";

    bool expResult = false;
    string word = "futbol";
    string word2 = "araba";

    string word3 = "anne";
    string word4 = "bebe";
    string word5 = "sürt";
    string word6 = "çiz";


    Console.Write(Description1 + " (Erkek):");
    //futbol || araba
    expResult = Description1.ToLower().Contains(word.ToLower()) || Description1.ToLower().Contains(word2.ToLower());
    Console.WriteLine(expResult);

    //anne || bebe , sürt && araba , çiz && araba
    Console.Write(Description2 + "(Bayan):");
    expResult = ((Description2.ToLower().Contains(word3.ToLower()) || Description2.ToLower().Contains(word4.ToLower()))
        || (Description2.ToLower().Contains(word5.ToLower()) && Description2.ToLower().Contains(word2.ToLower()))
        || (Description2.ToLower().Contains(word6.ToLower()) && Description2.ToLower().Contains(word2.ToLower())));
    Console.WriteLine(expResult);

    //anne || bebe , sürt && araba , çiz && araba
    Console.Write(Description3 + "(Bayan):");
    expResult = ((Description3.ToLower().Contains(word3.ToLower()) || Description3.ToLower().Contains(word4.ToLower()))
       || (Description3.ToLower().Contains(word5.ToLower()) && Description3.ToLower().Contains(word2.ToLower()))
       || (Description3.ToLower().Contains(word6.ToLower()) && Description3.ToLower().Contains(word2.ToLower())));
    Console.WriteLine(expResult);

    Console.ReadLine();
}
```

```
Futbol maçlarını çok sevsem de, genelde izlemiyorum (Erkek):True
Bebeğimin sağlığı için anne sütüne çok önem veriyorum.(Bayan):True
Spor'a giderken arabayı vurdum.(Bayan):False
```