Scope (Instance,Class ve Local Variables)

Object Nasil Kullanilir ?

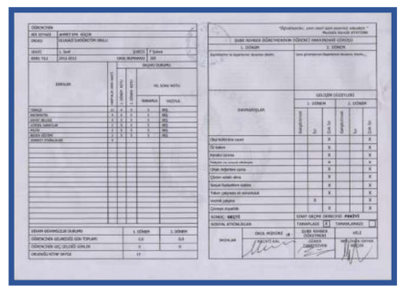
Ogretmen

Ogrenci

Dersler

09:00	TÜRKÇE-1
09:30	MATEMATİK-1
10:00	TÜRKÇE-2
10:30	MATEMATİK-2
11:00	TÜRKÇE-3
11:30	MATEMATİK-3
12:00	TÜRKÇE-4
12:30	MATEMATİK-4
13:00	İYEP TÜRKÇE

Personel



Notlar

Scope (Instance,Class ve Local Variables)

- Bir Class icerisinde olusturulan variable'lar icin Scope, o variable'a nereden, nasil ulasilabilecegini ve nerede gecerli oldugunu ifade eder.
- Scope'a uymayan bir kullanimda Java Compile Time Error verir.
- Java'da olusturulan variable'lar icin 4 Scope mevcuttur

  1) Instance (Object) Variables // ogretmenin adi gibi, ogrencinin notu gibi

  2) Static (Class) Variables // okul adi, adresi gibi

  3) Local (Method) Variables

  4) Loop Variables

Scope (Instance,Class ve Local Variables)

Instance (Object) Variable

Class'in icinde ancak main method'un disinda olmalidir

Static olmamalidir

Olusturulmasi yeterlidir, deger atanmasi sart degildir.

```
public class Example {

    int sayi;

    public static void main(String[] args) {

    }

}
```

Default Value

Eger instance bir variable olusturur ama deger atamazsaniz, Java otomatik olarak default degerleri assign eder. (String icin null, sayisal data turleri 0, boolean false)

Scope (Instance,Class ve Local Variables)

Instance (Object) Variable

class icerisinde veya baska class'larda direk kullanilamaz, kullanmak istedigimizde MUTLAKA object olusturmali ve object uzerinden ulasilmalidir.

```
public class Example {

    int sayi;
    char ilkHarf;
    String isim;
    boolean ogrenciMi;


    public static void main(String[] args) {

        Example ex1=new Example();
        System.out.println(ex1.sayi);
        System.out.println(ex1.ilkHarf);
        System.out.println(ex1.isim);
        System.out.println(ex1.ogrenciMi);
    }
```

Outputs

```
0

null
false
```

Ornek :

Bir okul uygulamasi yaptigimizi dusundugumuzde, ogretmenIsmi, ogrenciIsmi, matematiNotu gibi degiskenler bir kisi ile iliskilendirilmedikce anlamli olmaz

Scope (Instance,Class ve Local Variables)

Class (static) Variable

Class'in icinde ancak main method'un disinda olmalidir.

Static olmalidir

Olusturulmasi yeterlidir, deger atanmasi sart degildir.

```
public class Example {

   static int sayi;

    public static void main(String[] args) {

    }


}
```

Scope (Instance,Class ve Local Variables)

Class (static) Variable, class icerisinde direk kullanilabilir, baska class'larda kullanmak istedigimizde object olusturmaya ihtiyac duymadan classIsmi.variableIsmi ile variable'a ulasabilir ve kalici olarak degistirebiliriz.

```
public class Example {

    static int okulId;
    static String okulIsmi;
    static boolean acikMi;


    public static void main(String[] args) {

        System.out.println(okulId);
        System.out.println(okulIsmi);
        System.out.println(acikMi);

    }
```

Outputs

```
0
null
false
```

Ornek : Bir okul uygulamasi yaptigimizi dusunun okulIsmi, okulId, acikMi gibi degiskenler bir kisiyi degil okulla ilgili herkesi ilgilendirir ve bir kisi okul ismini veya okul telefon numarasini degistirirse okulla ilgili herkes icin okul ismi degisir.

Scope (Instance,Class ve Local Variables)

Instance Vs Class Variables

Instance (Object) Variable, class icerisinde veya baska class'larda direk kullanilamaz, kullanmak istedigimizde MUTLAKA object olusturmali ve object uzerinden ulasilabilir.

Class (static) Variable, class icerisinde direk kullanilabilir, baska class'larda kullanmak istedigimizde object olusturmaya ihtiyac duymadan classIsmi.variableIsmi ile variable'a ulasabilir ve kalici olarak degistirebiliriz.

Static variable'lar herkes icin ortaktir (okul ismi gibi) , instance variable'lar ise objeye baglidir (matematikNotu, ogrenciIsmi gibi)

Static variable yetkisi olan herkes tarafindan degistirilebilir ve bu degisim her obje icin gecerlidir. Instance variable da yetkisi olan herkes tarafindan degistirilebilir ancak yapilan degisiklik sadece o obje ile ilgilidir, geneli kapsamaz.

Scope (Instance,Class ve Local Variables)

Local Variable

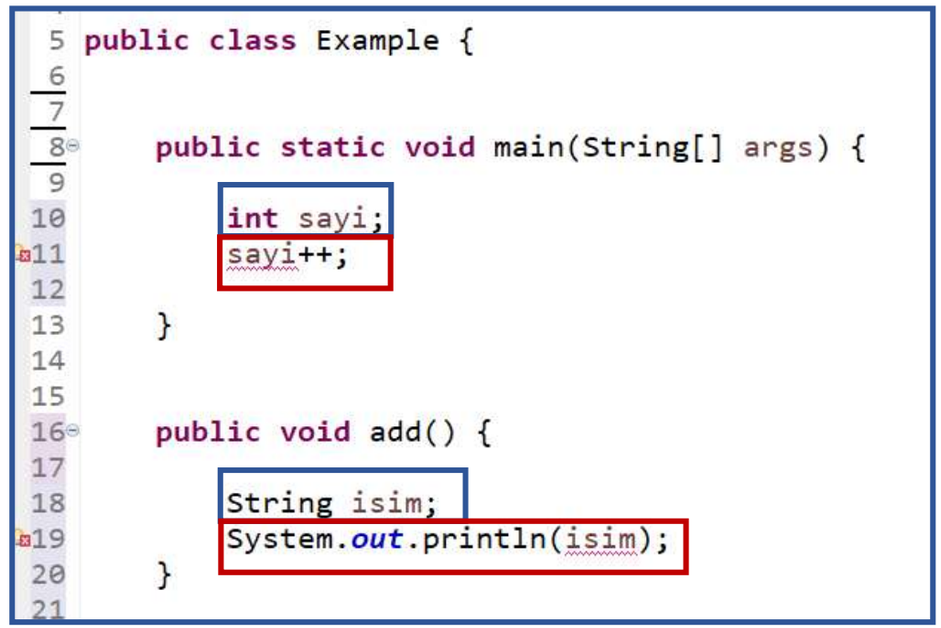
- Herhangi bir method icerisinde olusturulan variable'lardir (main method dahil).
- Sadece o method icerisinde gecerlidir
- Baska methodlarda da kullanilacak variable'lari, local olusturmak yerine class level'da olusturmak gereklidir.
- Class level'da olusturulacak variable, main method'da kullanilacaksa static olarak olusturulmalidir. Bu durumda bu variable kullanacak, diger method'lar da static olmalidir.

```
public class Example {

    public static void main(String[] args) {

        int sayi;

    }

    public void add() {

        String isim;
    }
}
```

Scope (Instance,Class ve Local Variables)

Local Variable

- Java local variable'lara default deger atamaz.
- Sadece olusturdugunuzda Java sikayet etmez. (variable olusturuldu method icerisinde deger atanacak diye bekler.)
- Olusturulan local variable'lara deger atamadan kullanmaya calisirsaniz Java sikayet eder(CTE)



```
5  public class Example {
6
7
8      public static void main(String[] args) {
9
10         int sayi;
11         sayi++;
12
13     }
14
15
16     public void add() {
17
18         String isim;
19         System.out.println(isim);
20     }
21
```

Scope (Instance,Class ve Local Variables)

Loop Variables

- Bir loop icinde olusturulan variable'lar sadece o loop icerisinde gecerlidir.
- Loop icerisinde olusturulan variable'lara loop disindan ulasilamaz ve loop disinda kullanilamaz.
- Loop icerisinde olusturulan local variable'lari disarida kullanmaya calisirsaniz Java sikayet eder(CTE)

```
public static void main(String[] args) {
    for (int i = 0; i < 10; i++) {
        int sayi=10;
        System.out.println(sayi);
    }
    System.out.println(sayi);
}
```

Scope (Instance,Class ve Local Variables)

```
public class MyClass{
      int num1;
      String name = "Ali";
    public static void main(String args){
      add();
         product (5);
    }
     public static add(){
         num1 ++;
         int num2 = 6;
         char letter;
         System.out.println("Do addition ");
    }
     public product(int num3){
         name = "Veli";
         num2++;
         System.out.println(num3 * num3);
    } }
```

1) Hangileri instance variable'dir ?
2) Hangileri local variable'dir?
3) num1 icin default value nedir ?
4) Java hangi satirlarin altini kirmizi cizer?
5) Kac satir compile time error verir?

Scope (Ozet)

Scope : Class icerisinde olusturulan variable'larin kapsamini (nereden eriselebilecegini) belirler

Temel olarak 4 Scope'dan bahsedebiliriz

Class Level'da olusturulan variable'lar class'in tamaminda gecerlidir, ancak direk erisim icin static keyword belirleyicidir

1- static olarak tanimlanan variable'lara tum method'lardan ulasilabilir

2- static olarak tanimlanmayan (instance) variable'lara sadece static olmayan method'lardan ulasilabilir

Local olarak olusturulan variable'lar sadece tanimlandiklari scope'da gecerlidirler. (Herkes oturdugu mahallede taninir)

3- bir method'da olusturulan variable'lara sadece o method'dan ulasilabilir

4- Loop icerisinde olusturulan variable'a loop disindan erisilemez

```
2  public class ScopeNedir {
3      static int sayi=5;
4      String ders="Java";
5      public static void main(String[] args) {
6          sayi=100;
7          ders="Java Course";
8          int mainsayi=20;
9          ders2="API";
10         for (int i = 0; i < 10; i++) {
11             System.out.println(i);
12             String ders3="SQL";
13         }
14
15         System.out.println(i);
16         ders3="API";
17     }
18     public static void staticMethod(){
19         sayi=110;
20         System.out.println(ders);
21         mainSayi=10;
22         System.out.println(ders2);
23
24     public void staticOlmayanMethod(){
25         System.out.println(sayi);
26         ders="Java Course";
27         System.out.println(mainSayi);
28         String ders2="Selenium";
   }}
```