**<<HASAN KOCAMAN>>**

6 ARALIK 2021 DERS 34

**YOUTUBE JAVA DERS 34 ACCESS MODIFIER ENCAPSULATION**

*String clas’indan uretilen Stringler immutable’dir yani degisiklik yapilamaz*

*StringBuilder mutable’dir ve atama yapmadan degisiklik yapar*

String’ten farki bu

StringBuilder objesinde = assignment olmali olmazsa degisiklik yapmaz

Varargs ve StringBuilderdan socrative yapildi

**================================================**

Public int blMethod (boolean b1, boolean… b2) { Return b2.length; }

blMethod(true, false,true) bastaki ilk eleman b1 e gidecek son iki eleman b2’ye gidecek yani b2’nin length’i 2 olacak

**=================================================**

Hangisi compile time error verir

Varargs once yazildiysa yani en son degilse cte verir**.Kisacasi varargs her zaman en sona yazilir**

Ornegin public void method (int…i, int j) yazilirsa hata verir

**==========================================================**

Public void method (int i, int j, int…k)

System.out.println(k.length)

Method (1,2,3,4,5) ilk 1 int I olur 2 int j ye gider geri kalan 3 eleman varargs a gider varargs in length I 3 olur

**==========================================================**

**String clasindan uretilen Stringler immutable dir yani degistirilemez**

**Stringbuilder clasindan uretilen Stringler mutabledir yani degistirilebilir**

**Stringbuilder ile Stringbuffer ayni isi yapar ama StringBuilder daha hizlidir**

**==========================================================**

StringBuilder strBld=new StringBuilder(“ceylan”);

Str.substring(3);

System.out.println(strBld);

ciktisi ne olur

substring method string den geliyor degisiklik yaptirmaz ancak yazdirirsak degisiklik gorulur

burda ceylan yazdirir cunku degisiklik assign edilmedi

ornegin:

StringBuilder str = new StringBuilder("admin");

System.out.println("string = " + str); //string = admin

System.out.println("substring is = " + str.substring(3));//substring is = in

System.out.println("substring is = " + str.substring(1, 4));//substring is = dmi

**=================================================**

StringBuilder strBld = new StringBuilder(“ceylan”);

strBld.insert(0,”S”); 0.indexe S harfini insert ettik burda kaynak yaptik

System.out.println(strBld); ciktisi Sceylan olur

**===============================================**

**StringBuilder uretme sekilleri:**

StringBuilder strBld=new StringBuilder(“Ayse”);

StringBuilder strBld=new StringBuilder();

StringBuilder strBld=new StringBuilder(7);//capacity 7 length ise 0 dir icinde String ifade yok

===============================================================

String s1=”Hurriyet”;

StringBuilder s2=new StringBuilder (“Hurriyet”);

If(s1==s2) { //java buna itiraz eder cunku String ile StringBuilder farkli data turleri CTE verir

System.out.println(“merhaba”);

}

If(s1.equals(s2) { //java buna itiraz etmez ama false verir

System.out.println(“hoscakal”);

}

Bu kod calismaz Compile Time Error verir

Burda concat olur ilk satir 12”Java”45 olur

StringBuilder sb=new StringBuilder(5+7+”Java”+4+5);//icine String yazarsak StringBuilder itiraz etmez

String isim=”Mesut”;

**sb.append**(isim,2,4);//append sona ekler burda ismin 2.ve 4. Indexleri arasini ekler 2 dahil 4 haric

//yani12Java45su olur

**sb.delete**(4,6); 4 dahil 6 haric siler substring( inclusive, exclusive )

yani 12Ja45su kalir geriye ve onu yazdirir

System.out.println(sb);

StringBuilder sb1=new StringBuilder (“OCAJP8”);

sb1.subSequence(2,4); 2 ile 4 arasini alir AJ

***subSequence degisiklik yapmiyor yazdirmazsan degisikligi goremezsin***

**sb1.deleteCharat**(3);3.index’teki harfi yani siler J’ yi siler

**sb1.reverse**() // reverse tersine cevirir 8PACO olur

System.out.println(sb1);

**subSequence degisiklik yapmiyor yazdirmazsan degisikligi goremezsin**

ornek

StringBuilder sb = new StringBuilder("Java ne kadar kolay");

System.out.println(sb.substring(14)); // kolay

//substring den sonra . koyarsak string methodlarini kullabilirim

System.out.println(sb.subSequence(14, 19));// kolay

//CharSequence subSequence(int start, int end) {

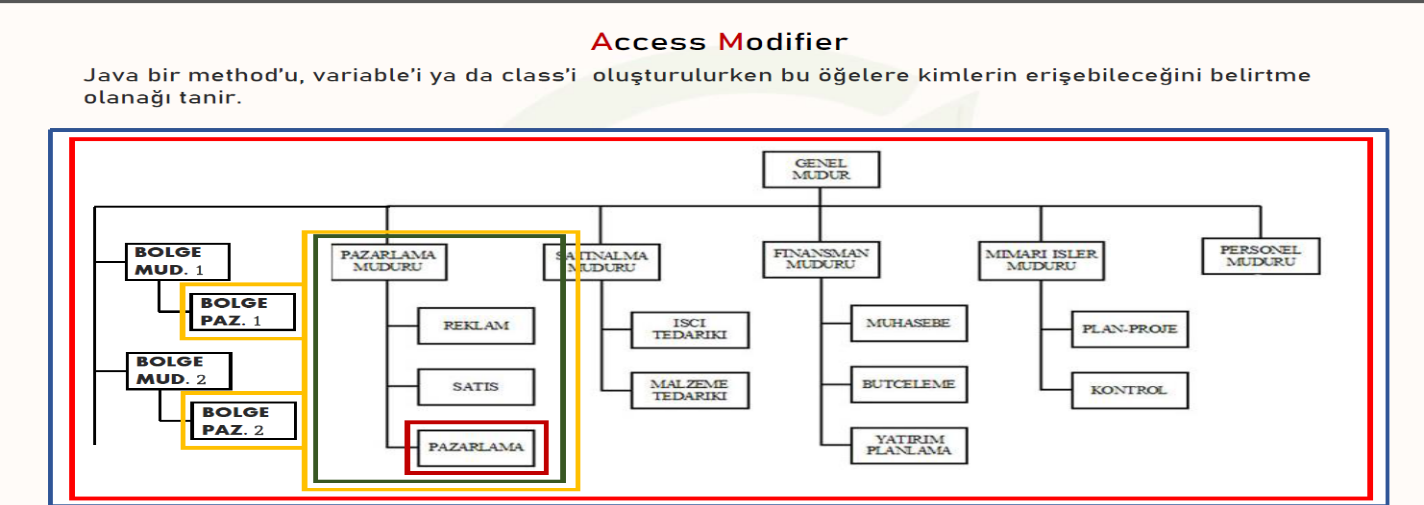
// return substring(start, end);

//subsequence method degil o yuzden methodlari getirmez

// ikisi de ayni sonucu dondurur ancak substring() String class'indan geldigi icin

// substring() kullandigimda arkasindan String method'larini kullanabilirim

// Ancak, subSequence kullandigimda bu mumkun olmaz

=======================================================================



============================================================================

**Private sadece ve sadece icinde bulundugu clas’ta kullanilir** Kendi eviniz

======================================================================

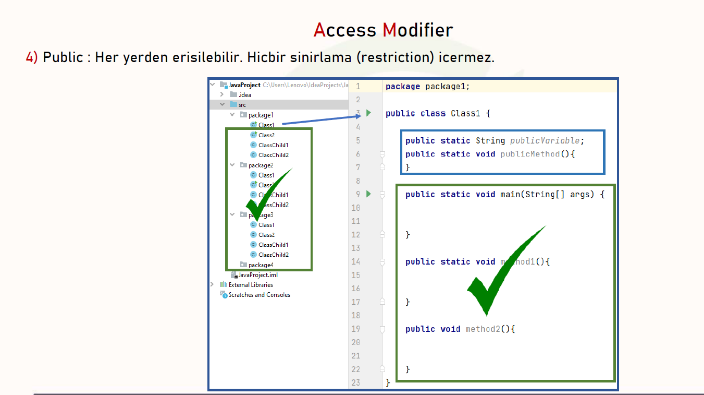
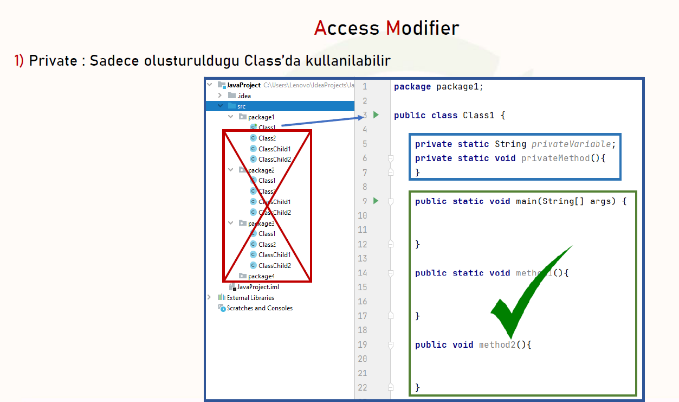
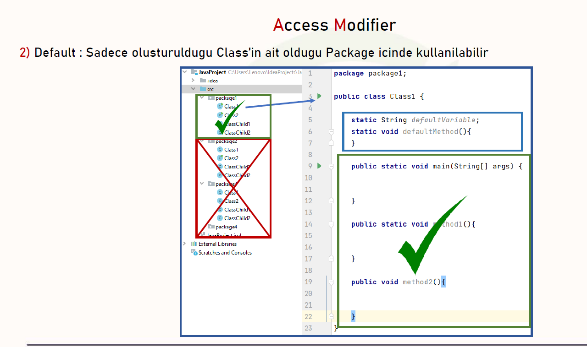
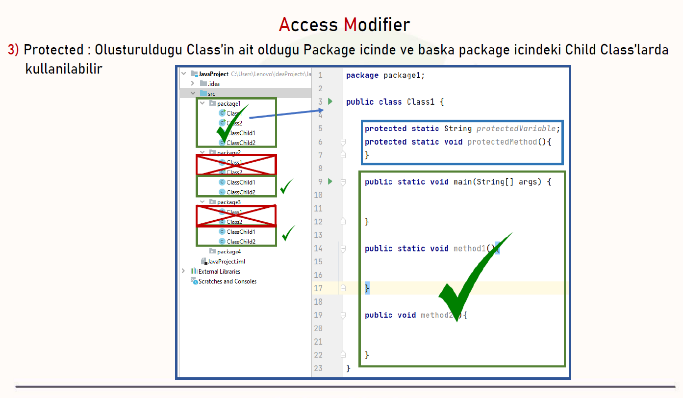
Default oldugunda basina default yazilmaz. Yani yazilmiyorsa bu default demektir

Ornek

static String Hasan; basinda public, private filan yazmiyor demekki default

**default ise sadece ayni package icindekiler gorebilir apartman gibi**

**default diger adi package access modifier’dir**

***protected ise kendi package ulasir ayrica baska package’lardaki child clas’lara da ulasabilir*** public her yerden ulasiliyor

C01 obj=new C01();

// System.out.println(obj.sayiPrivate);

// System.out.println(obj.sayiDefault); // sadece ayni package dakiler ulasabilir

// System.out.println(obj.sayiProtected); // ayni package'daki class'lar ve child class'lar ulasabilir

**encapsulation(data hiding)**

**kapsulleme**

****

encapsulation iki adimda yapilir

1. Class uyelerini (variable ve methodlari private yap)
2. Public olan getter ve setter methodlar uretmelisiniz

C05 obj1=new C05();

System.out.println(obj1.getAsgariMaas());// 4000

// obj1.asgariMaas=3000;

*// C05 class'inda asgariMaas'i private yapip getter() kullandigim icin*

*// asgari maasi gorebiliyorum ama degistiremiyorum*

*// classic acces modifier'larimiz ile bir variable'a ulasabilirsek*

*// istedigimiz zaman degistirme secenegimiz de olur*

**ulassin ama degistiremesin= getter**

**degistirsin ama goremesin=setter**

*//bir class'daki herhangi bir class uyesine*

*// ulasabilsin ama degistiremesin diyorsaniz*

*//1-class uyesini private yapin*

*//2- private clas uyelerine hicbir sekilde ulasilamayacagindan dolayi*

*//sadece gorebilme yetkisi verin getter()*

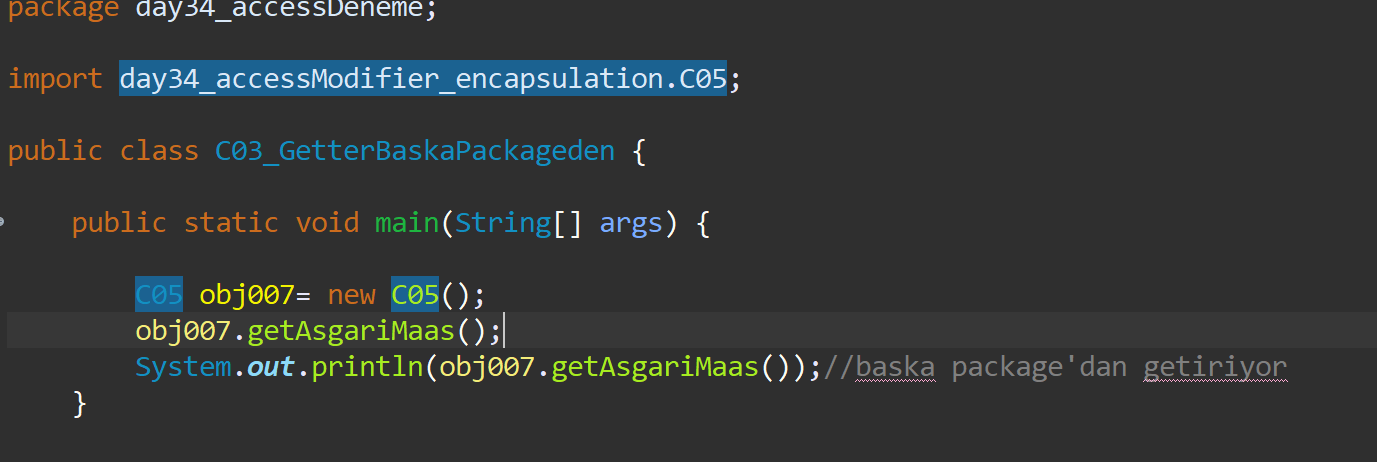
Kod yazdigimiz ekranda sag tikla🡪 source->generate getters and setters tikla

Daha sonra setter veya getteri secebiliriz

Method getter ise sadece gosterecek return asgariMaas

get AsgariMaas methodu public olursa bu methoda baska packagedan da ulasilabilir

hem gorsun hem degistir diyorsan getter setter ihtiyac yok public yap gec….



Dikkat et burda C05 constructor’i getAsgariMaas methodunun oldugu clas’ta

Obj007. dedigimizde o yuzden o clas’a gidip bakabiliyor