

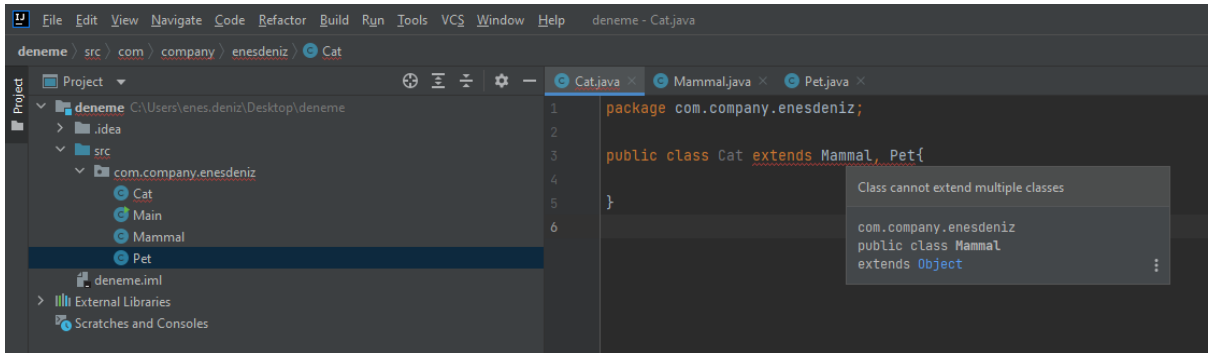
JAVA DİLİNDE ÇOKLU KALITIM

Çoklu kalıtım (Multiple Inheritance) kavramı bazı dillerde varken bazılarında yoktur. Şu an inceleyeceğimiz Java dilinde de doğrudan çoklu kalıtım bulunmamaktadır. Ama bu kavramın yerini dolduracak bazı best practiceler geliştirilmiştir. Java 8 sürümü ile kavramın implementesi biraz daha kolaylaştırılmıştır. Bu durumlardan aşağıda bahsedilecektir.

Javada çoklu kalıtımı engelleyen en büyük engel Has a(Kompozisyon) ve Is a(Kalıtım) ilişkilendirilmesidir. Kalıtım almak Is a kavramına dahil olur. Java dili örnek vermek gerekirse;

Bir kedinin hem Memeli hemde evcil hayvan olma durumunu engellemektedir. Çünkü burada Is a ilişkisi vardır.

Engellenen durum:



Java'da böyle bir tanımlama bulunmamaktadır.

DIAMOND PROBLEM

Bahsi geçen olay bizlere, Diamond Problem adı verilen soruna yol açar. Durumu somut olarak örneklendirecek olursak; Doğadaki bütün hayvanları temsil eden “Hayvanlar Alemi” isimli bir sınıf var.

“Hayvanlar Alemi” sınıfının “Yemek ye” isimli metodu olsun. Bu sınıfın altında bulunan birçok hayvan sınıfı var.

Örneğin bu hayvan sınıfları arasında bulunan “Kedi” ve “Köpek” isimli iki sınıfı inceleyelim. Bu iki sınıfta “Hayvanlar Alemi” sınıfından kalıtım yoluyla türetildiği için

“Yemek Ye” metoduna sahip. Ama “Kedi” ve “Köpek” sınıflarının kendine has yemek yeme özellikleri olduğu için

“Yemek ye” metodunu kendilerine göre tekrar (Override) oluşturuyorlar. İlerleyen süreçte evde beslemek üzere olan türlerin “Kedi” ve “Köpek” türleri olduğu için “Ev Hayvanları” isimli bir sınıf oluşacaktır.

Bu sınıf tabii ki de “Hayvanlar Alemi” sınıfından değil “Kedi” ve “Köpek” sınıfından kalıtım ile türeteceğiz. Çünkü her hayvan ev hayvanı değildir.

Şimdi bizim “Kedi” ve “Köpek” sınıfından türetilen “Ev Hayvanları” sınıfımızın da “Yemek ye” metoduna sahip olması gerekiyor. [1].

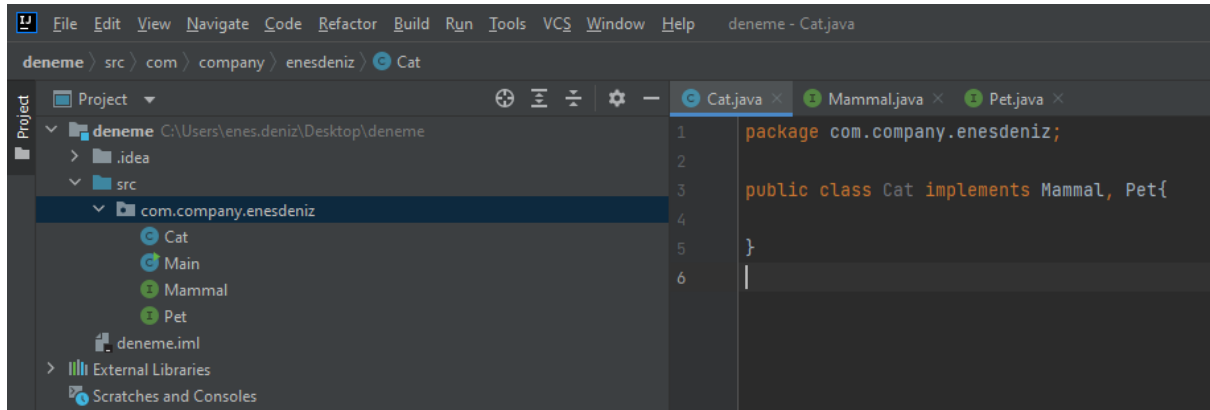
Sorun da tam olarak burada baş gösteriyor. Ev hayvanımıza verdiğimiz Yemek Ye() komutu sonrası ilgili class bu methodu hangi classtan alması gerektiği durumuna karar veremiyor.

Şekil itibari ile bu problem Elmas şekline benzediği için Diamond Problem adı ile adlandırılır.

PROBLEMİ NASIL AŞARIZ ?

İlgili sorunu aşabilmek için Java dilinde bu sınıfları kalıtım almak yerine Interface ile implemente etmek fikri uygulanmaktadır.

Yapı şu şekildedir:



Java 8 sürümü ile birlikte bilinen bazı yapılar yıkıldı ve yerine farklı durumlar getirildi. Normalde java da Interface e ait bir methodun body(gövde) kısmı boş olması gerekliydi.

Bu methodun implemente edilen sınıf tarafından Override edilerek kullanılması gerekliydi. Fakat çoklu kalıtımın sağlanabilmesi için gövdesi dolu olan methodlar geliştirebilir olduk.

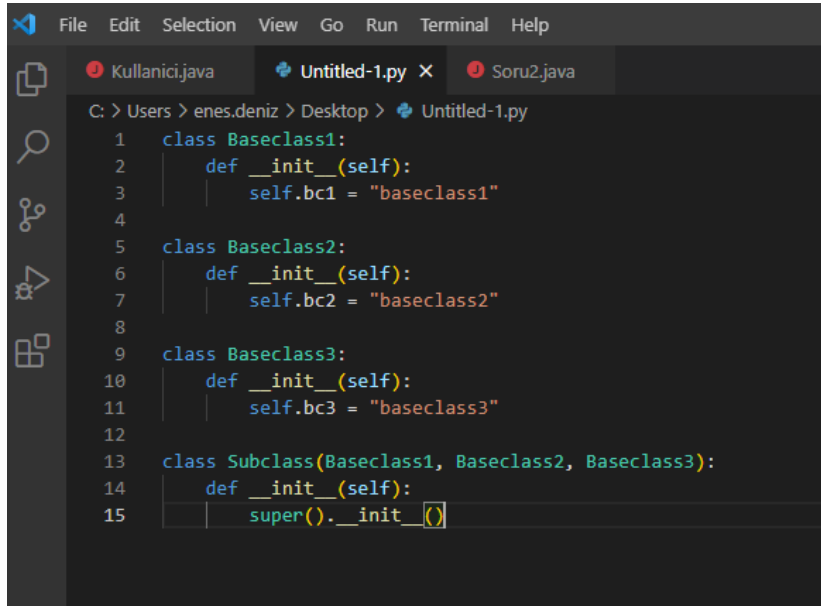
ÇOKLU KALITIM BULUNAN DİLLER

Çoklu kalıtımı destekleyen diller şunlardır: C++ , Common Lisp (Common Lisp Nesne Sistemi (CLOS) aracılığıyla), EuLisp (EuLisp Nesne Sistemi TELOS aracılığıyla), Curl , Dylan , Eiffel , Logtalk , Object REXX , Scala (mixin sınıflarının kullanımı yoluyla), OCaml , Perl , POP-11 , Python , R , Raku ve Tcl (8.6'dan yerleşik veya önceki sürümlerde Artımlı Tcl (Incr Tcl) aracılığıyla). [2].

Python dilinde çoklu kalıtımın nasıl sağlandığını inceleyeceğiz;

Pythonda bulunan çoklu kalıtım senaryosunda, belirtilen herhangi bir davranış ilk önce geçerli sınıfta aranır. Bulunmazsa, arama aynı sınıfta iki kez arama yapmadan önce derinlemesine, sol-sağ tarzında üst sınıflara devam eder.

Yani, yukarıdaki örnekte MultiDerivedsınıfın arama sırası (MultiDerived, Base1, Base2, object) şeklindedir . Bu sıraya aynı zamanda MultiDerivedsınıfın doğrusallaştırılması da denir ve bu sırayı bulmak için kullanılan kurallar kümesi Yöntem Çözüm Sırası (MRO) olarak adlandırılır .

A screenshot of a Python IDE window. The window has a menu bar with 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', 'Terminal', and 'Help'. Below the menu bar, there are three tabs: 'Kullanici.java', 'Untitled-1.py', and 'Soru2.java'. The 'Untitled-1.py' tab is active, showing a Python script. The script defines three base classes: 'Baseclass1', 'Baseclass2', and 'Baseclass3'. Each base class has an '__init__' method that assigns a value to a class attribute ('bc1', 'bc2', and 'bc3' respectively). Then, a 'Subclass' is defined that inherits from all three base classes. The 'Subclass' also has an '__init__' method that calls 'super().__init__()' to initialize the parent classes.

```
C: > Users > enes.deniz > Desktop > Untitled-1.py
1  class Baseclass1:
2      def __init__(self):
3          self.bc1 = "baseclass1"
4
5  class Baseclass2:
6      def __init__(self):
7          self.bc2 = "baseclass2"
8
9  class Baseclass3:
10     def __init__(self):
11         self.bc3 = "baseclass3"
12
13     class Subclass(Baseclass1, Baseclass2, Baseclass3):
14         def __init__(self):
15             super().__init__()
```

KAYNAKÇA

- [1] Websitesi : <https://www.erhanbaran.com/diamond-problem-nedir/> Eriřim Tarihi : 03/02/2022
- [2] Websitesi : https://stringfixer.com/tr/Diamond_problem Eriřim Tarihi : 03/02/2022