**====ABSTRACT====**

(melek gibi beden yok) public abstract class Honda{

motor(); deri(){“koltuklar deri”}

public class Civic extends Honda{ public class Accord extens Honda{ public class Crv extends Honda

motor(){2.5} deri(){“gerçek deri”}

motor(){2.0} deri(){“suni”}

motor(){1.6}

public abstract class Lastik extends Honda{

Bazen child class lar parent class taki metodların body lerini hep değiştirerek kullanırlar. Bu tip durumlarda parent class taki metoda body (süslü parantez ve içini ) koymak manasız olur, çünkü parent class taki metod body hiç kullanılmaz. Bu manasızlığı gidermek için Java bize body siz metod üretme imkanı verir. Body siz metodlara abstract metod denir.

Parent class taki body siz metodları (abstract metod) child class lar override etmek (metod signature ına dokunmadan süslünün içini değiştirmek) zorundadırlar. Biz bir aplication oluştururken child class ları parent class taki bazı metodları override etmek zorunda bırakmak istersek o metodları abstract metod yaparız.

abstract metod oluşturmak için; a)”{ }” body kısmını yazmayın

b)Access modifier dan sonra abstract keyword ünü kullanın 🡺public abstract void motor();

abstract keyword ü ile {} metod body bir arada kullanılmaz.

abstract metodlar sadece abstract class lar içinde oluşturulabilirler.

abstract class oluşturmak için Access modifier dan sonra abstract keyword ünü kullanın.

abstract class içinde hem abstract hem de abstract olmayan (concrete) metodlar kullanılabilirler.[motor(); ve deri() {“koltuklar deri”}]

Bir concrete class (mesela child ı gibi ) abstract class a extend ettiğinde, abstract class içindeki tüm abstract metodları override etmek zorundadırlar. Aksi takdirde Java CTE verir. Fakat concrete metodları override etmek zorunda değildirler. Concrete metodların override edilmesi isteğe bağlıdır.(üstteki şekilde bu anlatılmış, Lastik class ı hariç).

Bir abstract class başka bir abstract class a extend ettiğinde herhangi bir metodu override etmek zorunda değildir (şekildeki Lastik abstract class ı gibi). İsterse override eder isterse etmez.

Concrete class larda abstract metod üretilemez. (Civic class ında public abstract void yakit(); şeklinde metod olamaz.)

“final class” ların child class ı olamaz (adı üstünde son class, altında bir class olamaz ki child ı olsun). abstract class lar child class lara talimat vermek için oluşturulurlar. Eğer bir class ı “final” yaparsanız o class ın child class ı olamayacağından abstract olması mantıklı değildir. Bu yüzden Java abstract class ların “final” olmasına müsaade etmez.

abstract metod üretmemizin sebebi child class ların kullanabilmesidir. Eğer metodu “final” yaparsak bu, o metod override edilemez demektir yani, child class lar kullanamaz demektir. Bu çelişkiden dolayı Java abstract metodların “final” olmasına müsaade etmez.

abstract metod lar private olamazlar, çünkü abstract metod lar başka class lar tarafından kullanılmak için üretilir. Halbuki private metodlar başka class lar tarafından kullanılamazlar.

abstract metod lar static olamazlar. Çünkü static metodlar override edilemezler. Halbuki biz abstract metodları override etmek için üretiriz.

**ABSTRACT CLASS LARDAN OBJECT ÜRETİLEMEZ!!!!**