/\* \*\*\* Encapsulation nedir ?\*\*\*

- Hassas dataları korumak için kullanılan data saklama yöntemidir

\*\*\* Hassas dataları nasıl saklarız ?

- Dataları private access modifier kullanarak saklarız

\*\*\* Saklanan datalara diğer class lardan nasıl ulaşırız ?

- Getter ve Setter methodunu kullanarak ulaşabiliriz

\* Bir class taki tüm veriler için getter setter methodlarını kullanmak mantıklı değildir.

\* Encapsulation methodlar ve variablelar için yani class üyeleri için kullanılabilir.

\* Tüm dataların saklanması yani private yapılması OOP conseptine uygun olmaz.

\* Sadece okunması istiyorsak getter methodunu kullanırız.

\* Değiştirilebilsin istiyorsak setter methodu kullanırız.

\*

\*\*\* Encapsulation 2 adımda yapılır

-İlk olarak class üyelerini(variable ,method) private yaparız

-Daha sonra public olan getter() ve setter() methodları üretiriz

!!! getter() methodu datayı yalnızca okumamıza yarar, data da değişiklik yapamaz

!!! setter() methodu değer atanmasına izin verir ancak bilgilerin görülmesine izin vermez

\*\*\* Immmutable Class nedir?

--- Sadece getter methodu oluşturulan classlarda data değerleri değiştirilemez, sadece okunabilir

Bu tarz classlara immutable class denir.

Yani bir classı encapsule eder hiç setter methodu oluşturmazsak o classı immutable yaparız

\*\*\* Data türünü boolean oluşturduğumuzda Java get methodlarını isGecerMi() olarak oluşturur

\*\*\* setter() methodları için naming convention nedir? - Tüm data türleri için isimler "set" ile başlar

Aşağıdakilerden hangileri doğrudur?

A- Encapsulation package private instance variable ları kullanır

(Package private default access modifier demektir,bu kullanılmaz yalnızca private instance variable kullanılır)

B- Encapsulation private instance variable kullanır (doğru)

C- Encapsulation setter lara izin verir (doğru)

D- Immutability package private instance variable ları kullanır

(Immutable olması için private olmalı ve getter() methodları olmalıdır)

E- Immutability private instance variable ları kullanır (doğru)

F- Immutability setter() methodlarına izin verir

(Yanlış, ımmutable olması için yalnızca getter() kullanılabilir

\*\*\* Inheritance \*\*\*

1- Child class lar ,Parent clas ın tüm özelliklerini taşırlar

2- Child class lar, Parent class ın bazı özelliklerini kendilerine adapte ederler

3- Child class ların Parent class ta olmayan kendilerine has özellikleri de olabilir

4- Bir objenin farklı class larla karşılaştırıldığı zaman farklı görevleri ve farklı özellikleri olabilir

\*\*\* Hangi class ta çalışıyorsanız o class ı ve onun PARENT classlarını görebilirsiniz ve o özellikler üzerinden çalışabilirsiniz

Child classlarına has özelliklerini göremezsiniz

\*\*\* Java da inheritance,bir objenin/classın başka bir objenin/classın tüm özelliklerini ve davranışlarını elde ettiği mekanizmadır

\*\*\* Java da inheritance ın arkasındaki fikir, daha önceden oluşturulmuş classların üzerine yeni classlar oluşturabilmektir

\*\*\* Inheritance sayesinde yeni oluşturduğumuz bir classın var olan bir classın tüm methodlarını ve variablelarını kullanabiliriz

\*\*\* Inheritance bu işlemin adıdır, Inheritance sayesinde child class,parent class taki public veya protected primitive dataları

objectleri veya methodları problemsiz bir şekilde kullanabilir

!!! Inheritance sayesinde yazılan bir code un tekrar tekrar kullanılabilmesi(reusability) mümkün olur

Geneli kapsayan class üyeleri parent classa, daha spesifik olanlar ise child classlarda oluşturulur

Not 1: Child classlar public ve protected dataları problemsiz bir şekilde inherit edebilir

Not 2: Private datalar inherit edilemez

Not 3: Default datalar child ve parent aynı package da oldukları zaman inherit edilebilir

Not 4: Static methodlar veya variablelar inherit edilemez çünkü zaten her yerden class ismiyle ulaşılabilir

\* Protected inheritance için tanımlanmış access modifier dır, public de ınheritance için kullanılabilir

---- Interview Sorusu----

Neden Enheritance kullanırız ?

- Inheritance sayesinde parent olarak tanımlanan class( ve onun parent classlarındaki) protected/public class

üyelerini kullanabiliriz. (Reusability)

---------------------------

Inheritance faydaları nelerdir ?

1- Tekrarlardan kurtuluruz

2- Daha az kod yazarak işlemlerimizi yapabiliriz (less code)

3- Kolayca update yapabiliriz

4- Applicationın bakımı ve sürdürülmesi (maintance) kolaylaşır

\*\*\* Eğer variable lara access modifier yazmazsak Java access modifier olarak deault acc. mod. atar

Böyle olursa sadece o package altındaki child classlar parent classini inherit edebilir

Başka packagelardaki child classların da parent class ı inherit edebilmesi için variable ve methodların access

modifier larını protected yaparız

public yaparsak da inherit edilebilir ama biz herkesin erişmesini değil de sadece child classların erişmesini istediğimizden

protected ı tercih ederiz