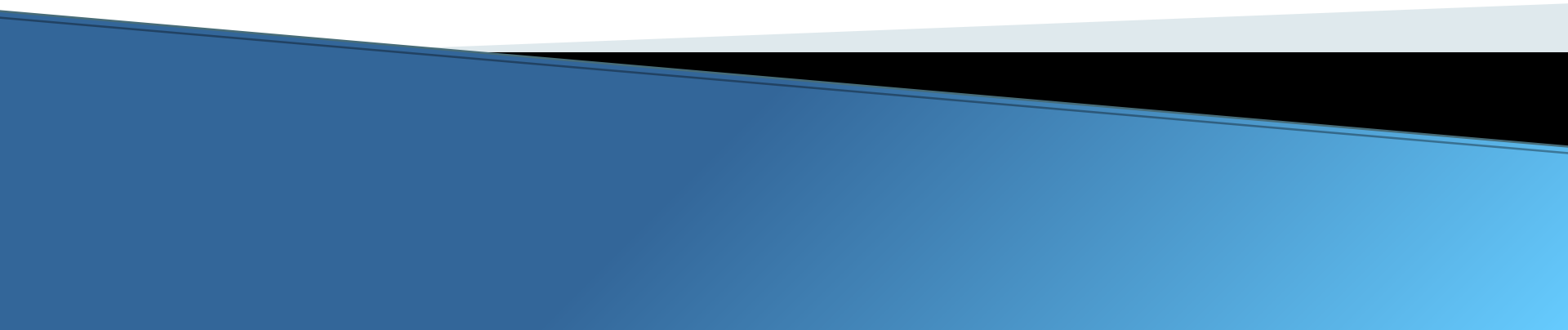


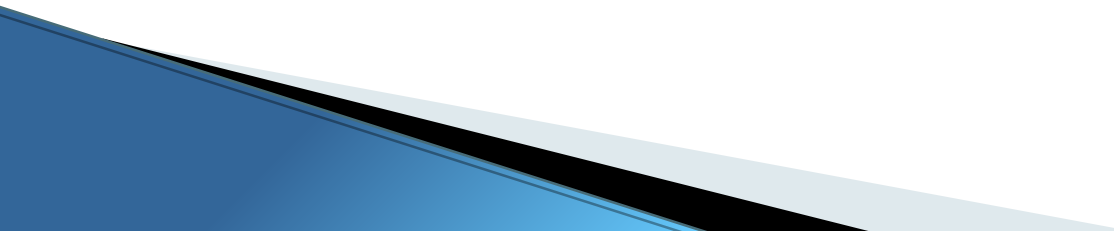
Herentzia anizkoitza

Java Interfazeak

SOFTWARE INGENIARITZA



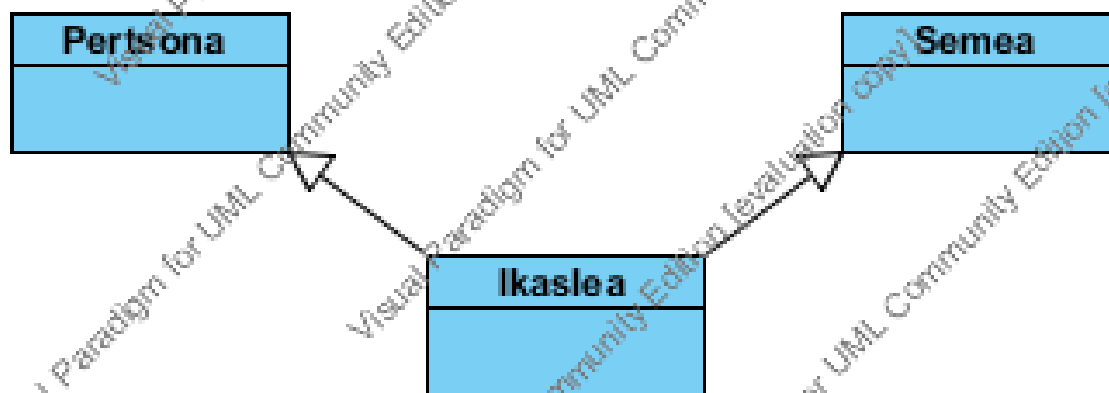
EDUKIAK

- ▶ Motibazioa
 - ▶ Abantailak
 - ▶ Diamantearen arazoa
 - ▶ Interfazeak
- 

Motibazioa

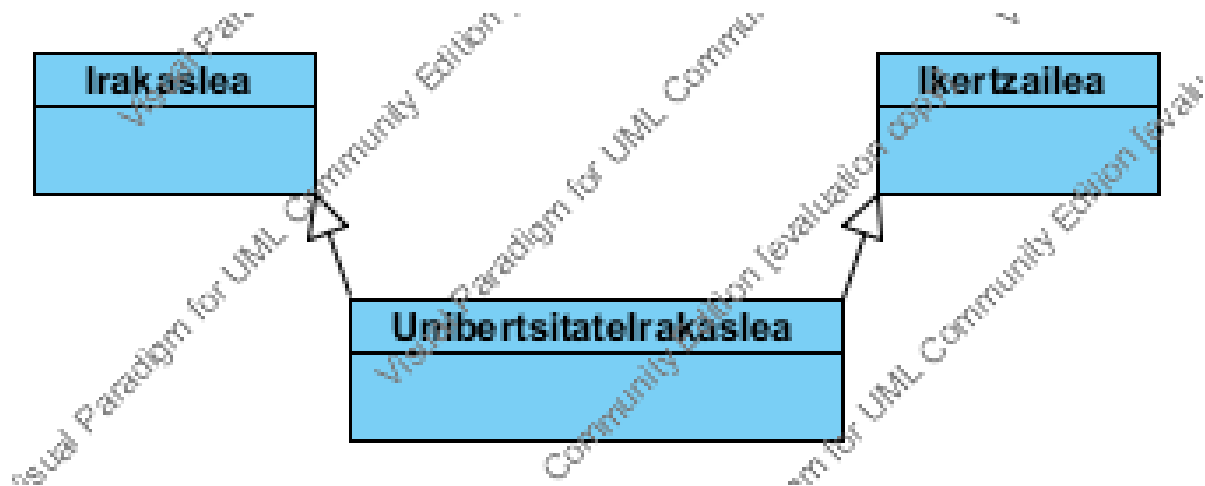
Mundu errealean entitateek guraso bat baino gehiagotik heredatzen dituzte ezaugarriak eta jokabideak.

Herentzia anizkoitza objektuei bideratutako programazio lengoiaen mekanismo bat da; hots, klase batek superklase bat baino gehiagotik jokabide eta ezaugarriak heredatzea.



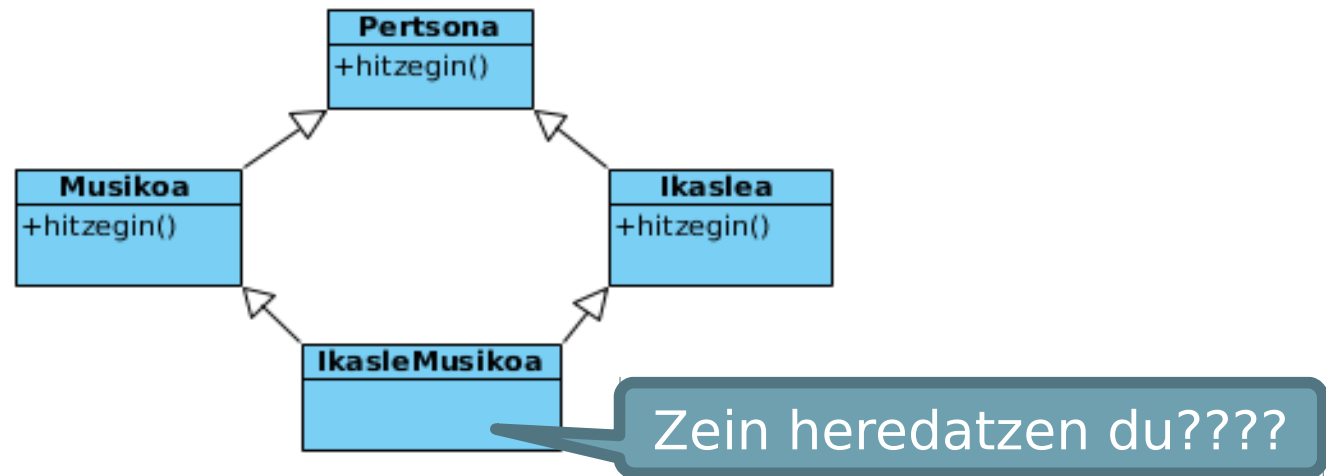
Abantailak

- Errealistagoaz gain, herentzia sinplearen berrerabilpen-aukerak handitzen ditu.



Diamantearen arazoa

Klase batek izen bereko ezaugarri edo funtzionalitate bat bi superklase ezberdinetatik heredatzen duenean.



JAVAK ez du herentzia anizkoitza onartzen!!!

Interfazeak

- JAVA interfazea klase abstraktu hutsa da
 - Metodo guztiak abstraktuak
- Atributuak eduki ditzake
 - Beti *static* and *final* izango dira.
- Beste edozein klase bezala deklaratu, baina “class” hitz erreserbatua erabili ordez “interface”.

```
public interface ilInterfazelzena { ... }
```

Interfazeak

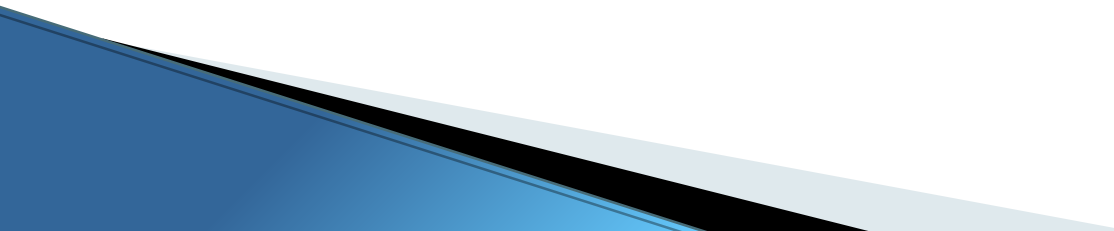
- Interfaze baten deklaraturako metodoak beti publikoak; beste klase batzuetan inplementatu.

```
public class klaselzena implements Interfazelzena  
    { ... }
```

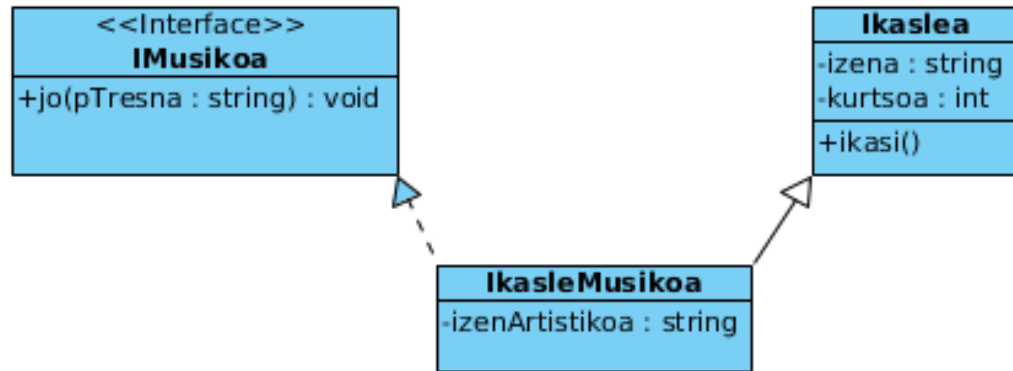
- Interfazeek beste interfaze batzuk hedatu (extends) ditzakete
- Klase batek interfaze bat baino gehiago inplementatu (implements) dezake

Interfazeak

Zertarako balio dute interfazeek?

1. JAVAn herentzia anizkoitzaren gabezia betetzeko.
 2. Klase batek beste objektu batzuek eskainitako datu eta metodoak erabili ahal izateko, horien superklase konkretuegatik arduratu barik.
 3. Klaseen ezaugarri sematikoak definitzeko.
- 

Interfazeak



```
public class IkasleMusikoa extends Ikaslea
implements IMusikoa {
    public String izenArtistikoa;
    public void jo() {
        ...
    }
}
```