OOP

Classi astratte e interfacce

Ereditarietà multipla

DEFINIZIONE DI CLASSE ASTRATTA

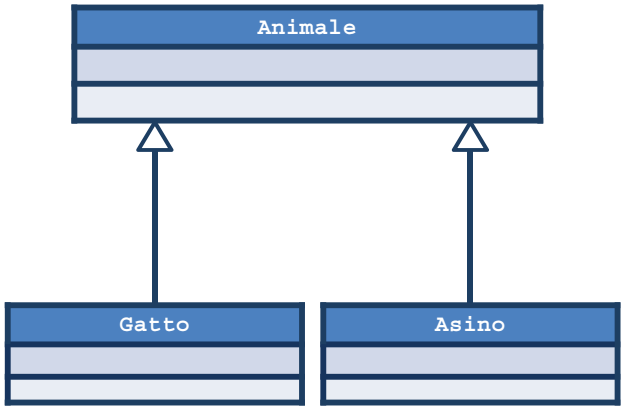
Una classe astratta è una classe che mappa un concetto astratto del mondo reale

ALESSANDRO URSOMANDO

DEFINIZIONE DI CLASSE ASTRATTA

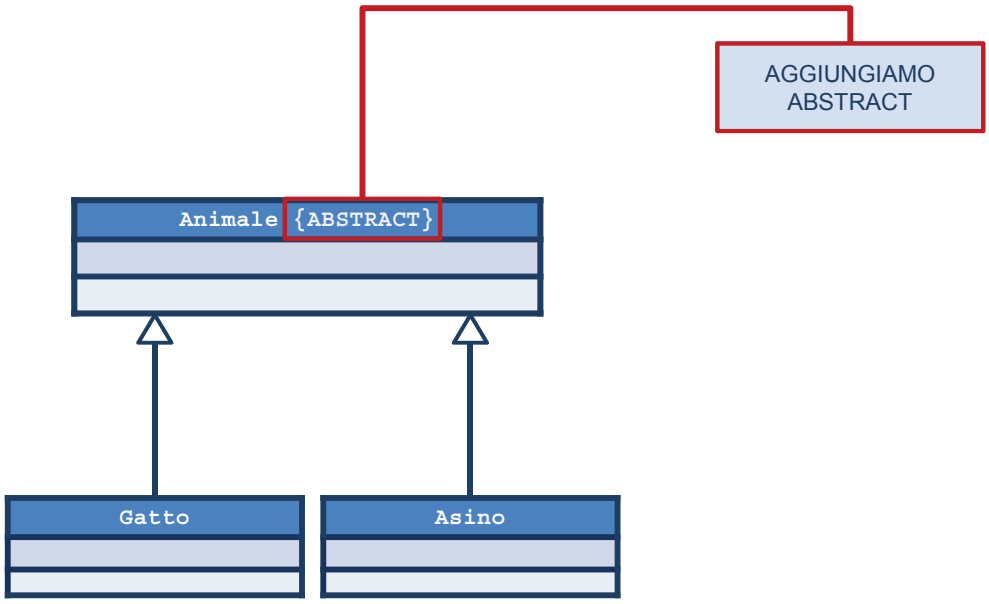
Una classe astratta è una classe che mappa un concetto astratto del mondo reale

Cioè una classe che non ha senso istanziare
ma che ha senso avere a disposizione per non duplicare codice



ALESSANDRO URSOMANDO

CLASSI ASTRATTE IN UML

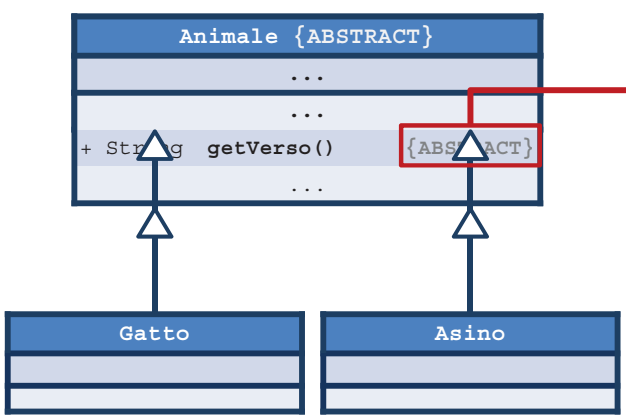


ALESSANDRO URSOMANDO

METODI ASTRATTI

Per definizione una classe astratta ha almeno un metodo astratto..

..del quale si specifica solo la firma: non c'è l'implementazione



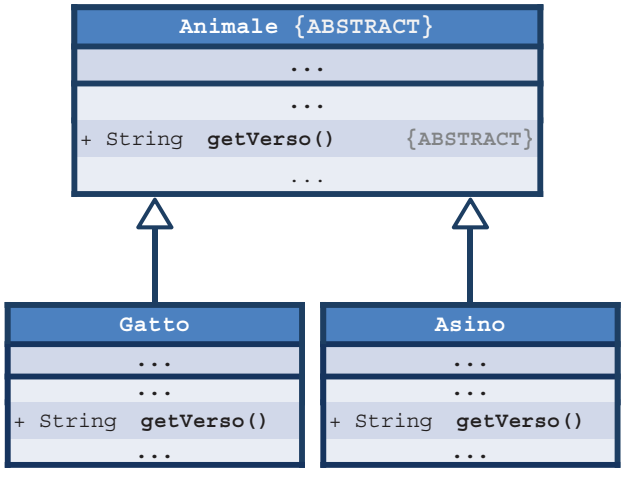

```
public class abstract Animale {
   ...
   public abstract String getVerso();
   ...
}
```

SPECIFICA UML
METODO ASTRATTO

METODI ASTRATTI

I figli di una classe astratta faranno l'override dei metodi astratti..

..e ne forniranno l'implementazione..




```
public class abstract Animale {
   ...
   public abstract String getVerso();
   ...
}
```

```
public class Gatto extends Animale {
   ...
   public String getVerso() { return "miagola"; }
   ...
}
```

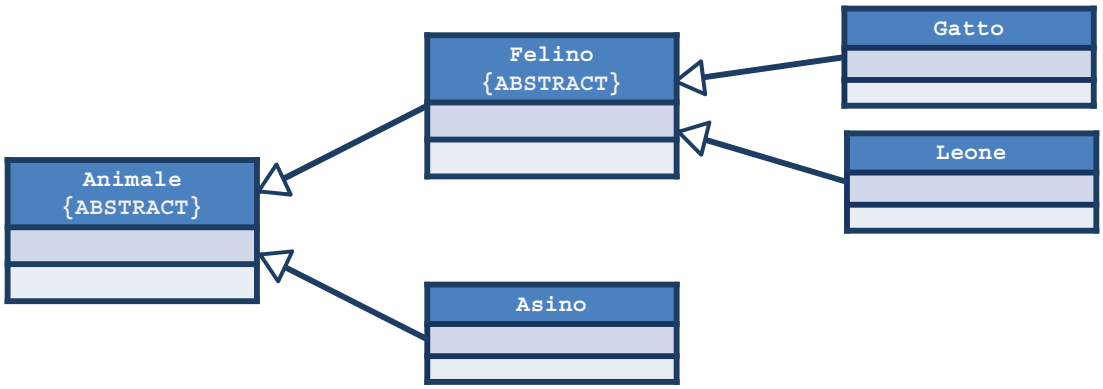
```
public class Asino extends Animale {
   ...
   public String getVerso() { return "raglia"; }
   ...
}
```

METODI ASTRATTI

I figli di una classe astratta faranno l'override dei metodi astratti..

..e ne forniranno l'implementazione..

..oppure saranno astratti a loro volta!



CONSIDERAZIONI FINALI

Perché hanno inventato le classi astratte?
Come si affrontavano situazioni tipo questa prima delle classi astratte?

```
public class Animale {
   // gli attributi
   ...
   protected String verso;
   ...
   // il costruttore
   public Animale() {
      ...
      verso = "verso non presente";
      ...
   }
   // gli altri metodi
   ...
   public String getVerso() {
      return verso;
   }
   ...
}
```

```
public class Gatto extends Animale {
   ...
   public Gatto() {
      ...
      verso = "miagola";
      ...
   }
   ...
}
```

```
public class Asino extends Animale {
   ...
   public Asino() {
      ...
      verso = "raglia";
      ...
   }
   ...
}
```

Si, ok, era un po' brutto..

CLASSI ASTRATTE E INTERFACCE

Una classe astratta che ha solo metodi astratti
può essere vista come **un'interfaccia**

Un'interfaccia è un contenitore
simile a una classe
ma che presenta qualche limitazione:

1) Tutti gli attributi (se ci sono) sono per definizione **public static** e **final**
2) tutti i metodi (se ci sono) sono per definizione **abstract** e **public**

LE INTERFACCE

Consideriamo l'insieme delle figure piane
(circonferenze, rettangoli, quadrati, trapezi..)
qualunque siano le loro caratteristiche di certo avranno un perimetro e un'area

```
public abstract class Figura {
   public abstract double getArea();
   public abstract double getPerimetro();
}
```

```
public interface Figura {
   double getArea();
   double getPerimetro();
}
```

L'interfaccia
non è una classe

LE INTERFACCE

Consideriamo l'insieme delle figure piane
(circonferenze, rettangoli, quadrati, trapezi..)
qualunque siano le loro caratteristiche di certo avranno un perimetro e un'area

```
public abstract class Figura {
   public abstract double getArea();
   public abstract double getPerimetro();
}
```

```
public class Rettangolo extends Figura {
   ...
}
```

```
public interface Figura {
   double getArea();
   double getPerimetro();
}
```

```
public class Rettangolo implements Figura {
   ...
}
```

Una classe implementa una interfaccia

LE INTERFACCE IN UML

Consideriamo l'insieme delle figure piane
(circonferenze, rettangoli, quadrati, trapezi..)
qualunque siano le loro caratteristiche di certo avranno un perimetro e un'area

```
public abstract class Figura {
   public abstract double getArea();
   public abstract double getPerimetro();
}
```

```
public class Rettangolo extends Figura {
   ...
}
```

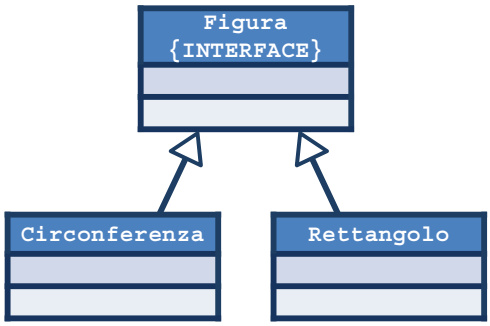
```
public interface Figura {
   double getArea();
   double getPerimetro();
}
```

```
public class Rettangolo implements Figura {
   ...
}
```

```
public class Circonferenza implements Figura
{
   ...
}
```

LE INTERFACCE IN UML

Consideriamo l'insieme delle figure piane
(circonferenze, rettangoli, quadrati, trapezi..)
qualunque siano le loro caratteristiche di certo avranno un perimetro e un'area




```
public interface Figura {
   double getArea();
   double getPerimetro();
}
```

```
public class Rettangolo implements Figura {
   ...
}
```

```
public class Circonferenza implements Figura
{
   ...
}
```


L'EREDITARIETÀ MULTIPLA

Una classe può **implementare** una o più interfacce (oltre che **estendere** una classe).