POLYMORPHISM and ABSTRACT CLASSES (NetBeans screenshots)


```java
package ec.edu.espe.zoo.view;

import ec.edu.espe.zoo.model.Animal;
import ec.edu.espe.zoo.model.Giraffe;
import ec.edu.espe.zoo.model.Mammal;
import ec.edu.espe.zoo.model.Platypus;
import ec.edu.espe.zoo.model.Vertebrate;
import java.util.ArrayList;
import java.util.Date;

/**
 *
 */
// Vertebrate is abstract; cannot be instantiated
// Variable animal is never read
// (Alt-Enter shows hints)
                                    // ----> POLYMORPHISM by Carlos Hernandez");
        Animal animal=new Vertebrate();
        Animal animal=new Mammal(...);
        System.out.println("animal--->"+animal);
        System.out.println("animal type--->"+animal.getClass().getSimpleName());

        ArrayList<Animal>animals= new ArrayList<>();
        animals.add( animal);

        animal=new Platypus(...,"Platypus Scientific",new Date(),...);
        System.out.println("Animal--->"+animal);
        System.out.println("animal type--->"+animal.getClass());
        animals.add( animal);

        animal=new Giraffe(...,"Girafus", new Date(),..., "big spine",...);
        System.out.println("Animal--->"+animal);
        System.out.println("animal type--->"+animal.getClass());
```



```java
package ec.edu.espe.zoo.view;

import ec.edu.espe.zoo.model.Animal;
import ec.edu.espe.zoo.model.Giraffe;
import ec.edu.espe.zoo.model.Mammal;
import ec.edu.espe.zoo.model.Platypus;
import java.util.ArrayList;
import java.util.Date;

/**
 * @author Carlos
 */
public class Zoo {
    public static void main(String[] args) {
        System.out.println(" ----> ABSTRACT CLASSES by Carlos Hernandez");
        Animal animal=new Mammal(...,"", new Date(),...,"",...);
        System.out.println("animal--->"+animal);
        System.out.println("animal type--->"+animal.getClass().getSimpleName());

        ArrayList<Animal>animals= new ArrayList<>();
        animals.add( animal);

        animal=new Platypus(...,true,...,"Platypus Scientific",new Date(),...);
```

Output - Zoo_Abstraction (run):
```
run:
----> ABSTRACT CLASSES by Carlos Hernandez
animal--->Mammal(Vertebrate)
 - Animal(id=0, scientificName=, bornDate=Mon Jan 15 00:56:25 ECT 2024, cageId=0;spine=, numberOfBones=0;numberOfMammaryGlands=0)
animal type--->Mammal
animal--->Platypus(Mammal(Vertebrate)
 - Animal(id=0, scientificName=Platypus Scientific, bornDate=Mon Jan 15 09:56:25 ECT 2024, cageId=0;spine=spine small, numberOfBones=0;numberOfMammaryGlands=0)
animal type--->class ec.edu.espe.zoo.model.Platypus
animal--->Giraffe(Mammal(Vertebrate)
 - Animal(id=0, scientificName=Girafus, bornDate=Mon Jan 15 09:56:25 ECT 2024, cageId=0;spine=big spine, numberOfBones=0;numberOfMammaryGlands=0)
animal type--->class ec.edu.espe.zoo.model.Giraffe
```



```java
package ec.edu.espe.zoo.view;

import ec.edu.espe.zoo.model.Animal;
import ec.edu.espe.zoo.model.Giraffe;
import ec.edu.espe.zoo.model.Mammal;
import ec.edu.espe.zoo.model.Platypus;
import java.util.ArrayList;
import java.util.Date;

/**
 *
 * @author Carlos
 */
public class Zoo {
    public static void main(...
// Mammal is abstract; cannot be instantiated
// (Alt-Enter shows hints)
        System.out.println(... Hernandez");
        Animal animal=new Mammal(0, 0, "", new Date(), 0, "", 0);
        System.out.println("animal--->"+animal);
        System.out.println("animal type--->"+animal.getClass().getSimpleName());

        ArrayList<Animal>animals= new ArrayList<>();
        animals.add( animal);
```

```
run:
---> ABSTRACT CLASSES by Carlos Hernandez
animal--->Giraffe{Mammal{Vertebrate{
- Animal{id=0, scientificName=gir, bornDate=Mon Jan 15 10:00:00 ECT 2024, cageId=0}spine=heavy, numberOfBones=0}number
animal type--->Giraffe
```

Project tree:

- Exam1
- Exam1P [main]
- IcreCreamMain [main]
  - Source Packages
    - ec.edu.espe.icecream.model
    - ec.edu.espe.icecream.utils
    - ec.edu.espe.icecream.view
  - Libraries
- IcreCreamMain2
- Person
- Pruebas
- Taxes
- Testsilbings
- Zoo
- Zoo_Abstarction
  - Source Packages
    - ec.edu.espe.zoo.controller
    - ec.edu.espe.zoo.model
      - Animal.java
      - Giraffe.java
      - Invertebrate.java
      - Mammal.java
      - Platypus.java
      - Vertebrate.java

Editor (Platypus.java):

```java
6  /**
7
8  Platypus is not abstract and does not override abstract method feed() in Animal
   ----
   (Alt-Enter shows hints)
9
10 public class Platypus extends Mammal {
11     private boolean poisoness;
12     private int poisonGlands;
13
14     public Platypus(boolean poisoness, int poisonGlands, :
15         super(numberOfMammaryGlands, id, scientificName, :
16         this.poisoness = poisoness;
17         this.poisonGlands = poisonGlands;
18     }
19
20     @Override
21     public String toString() {
22         return "Platypus{" +super.toString() + "poisoness
23     }
24
25     /**
26      * @return the poisoness
27      */
28     public boolean isPoisoness() {
29         return poisoness;
```

Lower project tree:

- Source Packages
  - ec.edu.espe.zoo.controller
  - ec.edu.espe.zoo.model
    - Animal.java
    - Giraffe.java
    - Mammal.java
    - Platypus.java
    - Vertebrate.java
  - ec.edu.espe.zoo.view
    - Zoo.java
- Test Packages
- Libraries
- Test Libraries

Lower editor:

```java
32     animal = new Giraffe(numberOfMammaryGlands: 0, id: 0, scientificName: "Girafus", new Date(), cage
33     System.out.println("Animal--->" + animal);
34     System.out.println("animal type--->" + animal.getClass());
```

```
animal type--->class ec.edu.espe.zoo.model.Platypus

Feeding Platypus with small fish and wores

Animal--->Giraffe{Mammal{Vertebrate{
- Animal{id=0, scientificName=Girafus, bornDate=Mon Jan 15 10:33:55 ECT 2024, cageId=0}spine=big spine, r
animal type--->class ec.edu.espe.zoo.model.Giraffe
other giraffe -->Giraffe{Mammal{Vertebrate{
- Animal{id=2, scientificName=Giraffa, bornDate=Thu Feb 01 00:00:00 ECT 3923, cageId=3}spine=long, number
brushing the neck of the giraffe
Feeding giraffe with grass
assign cage number -->25
```