

Alumno: CORTAZAR DE LA CRUZ MANUEL GIOVANNI

Matrícula:2193001368

CODIGO FUENTE

Principal.java

```
package animal;

/**
 *
 * @author giova
 */

public class Principal {

    public static void main(String[] args){

        //se crea un arreglo de animales

        Animal [] zoologico = new Animal[26];

        //instancia de las clases de animales

        zoologico[0] = new Anfibio("rana","glándulas mucosas","pulmones","oviparo","charco","verde");

        zoologico[1] = new Anfibio("sapo","glándulas mucosas","pulmones","oviparo","charco","cafe");

        zoologico[2] = new Anfibio("salamandra","carece de escamas y es húmeda","pulmones","oviparo","charco","verde");

        zoologico[3] = new Anfibio("triton","aterciopelada o verrugosa","pulmones","oviparo","charco","verde");

        zoologico[4] = new Anfibio("cecilia","glándulas mucosas","pulmones","oviparo","charco","verde");

        zoologico[5] = new Reptil("lagarto","escamas","Pulmones","oviparo","lago",true);

        zoologico[6] = new Reptil("iguana","escamas","Pulmones","oviparo","pantano",true);

        zoologico[7] = new Reptil("cocodrilo","escamas","Pulmones","oviparo","pantano",true);

        zoologico[8] = new Reptil("tortuga","escamas","Pulmones","oviparo","oceano",true);

        zoologico[9] = new Reptil("camaleon","escamas","Pulmones","oviparo","bosque",true);

        zoologico[10] = new Pez("salmon","escamas","branqueas","oviparo","oceano",2);

        zoologico[11] = new Pez("tiburon","escamas","branqueas","oviparo","oceano",2);
```

```

zoologico[12] = new Pez("atun","escamas","branqueas","oviparo","oceano",2);
zoologico[13] = new Pez("anguila","escamas","branqueas","oviparo","oceano",2);
zoologico[14] = new Pez("dorado","escamas","branqueas","oviparo","oceano",2);

zoologico[15] = new Ave("loro","plumas","pulmones","oviparo","bosque",true);
zoologico[16] = new Ave("avestrus","plumas","pulmones","oviparo","montaña",false);
zoologico[17] = new Ave("anguila","plumas","pulmones","oviparo","desierto",true);
zoologico[18] = new Ave("pinguino","plumas","pulmones","oviparo","polo sur",false);
zoologico[19] = new Ave("condor","plumas","pulmones","oviparo","montaña",true);

zoologico[20] = new Placentario("armadillo","pequeñas placas óseas","pulmones","viviparo","selva",
true,"herbivoro");

zoologico[21] = new Placentario("oso hormiguero","pelaje","pulmones","viviparo","selva",
true,"omnivoro");

zoologico[22] = new Monotrema("perry","carne","pulmones","oviparo","lagos",true, false,true);

zoologico[23] = new Monotrema("Equidna de hocico corto
","carne","pulmones","oviparo","monte",true, true,true);

zoologico[24] = new Marsupial("Topo ","aspera","pulmones","viviparo","monte",
true,"chico","herbivoro");

zoologico[25] = new Marsupial("Canguro ","pelaje suave","pulmones","viviparo","selva",
true,"grande","herbivoro");

//se imprimen todos los animales
for (int i = 0; i < zoologico.length; i++) {

    System.out.println("");

    System.out.println(i+1 + " " + zoologico[i].toString());

}

}

}

```

Animal.java

```
package animal;
```

```
/**
```

```
 *
```

```
 * @author giova
```

```
 */
```

```
//se crea la clase principal, la clase animal y se declaran los atributos comunes de un animal
```

```
public class Animal {
```

```
    private String tipoNombre;
```

```
    private String tipoPiel;
```

```
    private String tipoRespiracion;
```

```
    private String tipoReproduccion;
```

```
    private String tipohabitad;
```

```
    //constructor principal
```

```
    public Animal(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String tipohabitad) {
```

```
        super();
```

```
        this.tipoNombre = tipoNombre;
```

```
        this.tipoPiel = tipoPiel;
```

```
        this.tipoRespiracion = tipoRespiracion;
```

```
        this.tipoReproduccion = tipoReproduccion;
```

```
        this.tipohabitad = tipohabitad;
```

```
    }
```

```
    //getters y setters
```

```
    public String getTipoNombre() {
```

```
        return tipoNombre;
```

```
    }
```

```
public void setTipoNombre(String tipoNombre) {  
    this.tipoNombre = tipoNombre;  
}
```

```
public String getTipoPiel() {  
    return tipoPiel;  
}
```

```
public void setTipoPiel(String tipoPiel) {  
    this.tipoPiel = tipoPiel;  
}
```

```
public String getTipoRespiracion() {  
    return tipoRespiracion;  
}
```

```
public void setTipoRespiracion(String tipoRespiracion) {  
    this.tipoRespiracion = tipoRespiracion;  
}
```

```
public String getTipoReproduccion() {  
    return tipoReproduccion;  
}
```

```
public void setTipoReproduccion(String tipoReproduccion) {  
    this.tipoReproduccion = tipoReproduccion;  
}
```

```
public String getTipohabitad() {  
    return tipohabitad;  
}
```

```

    public void setTipohabitad(String tipohabitad) {
        this.tipohabitad = tipohabitad;
    }

    //se llama el toString que se reescribira en las siguientes subclases
    @Override
    public String toString() {
        return " Nombre: " + tipoNombre + " Piel: " + tipoPiel + " respiracion: "+ tipoRespiracion + "
reproduccion: "+ tipoReproduccion + " habitad: "+ tipohabitad;
    }
}

```

Anfibio.java

```

package animal;

/**
 *
 * @author giova
 */
//la subclase hereda de la clase principal
public class Anfibio extends Animal {

    //se declara la variable unica
    private String colorPiel;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del
    mismo

    public Anfibio(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String
tipohabitad, String colorPiel) {

        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad);

        this.colorPiel = colorPiel;
    }
}

```

```

    }

    public String getColorPiel() {
        return colorPiel;
    }

    public void setColorPiel(String colorPiel) {
        this.colorPiel = colorPiel;
    }

    //se reescribe el toString con una modificacion
    @Override
    public String toString() {
        return "El color de piel es "+colorPiel+ super.toString(); //To change body of generated methods, choose
Tools | Templates.
    }
}

```

Ave.java

```

package animal;

/**
 *
 * @author giova
 */
//la subclase hereda de la clase principal
public class Ave extends Animal {
    //se declara la variable unica
    private boolean vuela;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del
    mismo

    public Ave(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String
tipohabitad, boolean vuela) {
        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad);
    }
}

```

```

        this.vuela = vuela;
    }

    public boolean isVuela() {
        return vuela;
    }

    public void setVuela(boolean vuela) {
        this.vuela = vuela;
    }

    //se reescribe el toString con una modificacion
    @Override
    public String toString() {
        if(isVuela())
            return "Se trata de un ave que vuela " + super.toString();
        else
            return "Se trata de un ave que No vuela " + super.toString();
    }
}

```

Mamifero.java

```

package animal;

/**
 *
 * @author giova
 */
//la subclase hereda de la clase principal

public class Mamifero extends Animal {

```

```

//se declara la variable unica
private boolean terrestre;

//se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del
mismo

public Mamifero(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion,
String tipohabitad, boolean terrestre) {

    super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad);

    this.terrestre = terrestre;
}

public boolean isTerrestre() {

    return terrestre;
}

public void setTerrestre(boolean terrestre) {

    this.terrestre = terrestre;
}

//se reescribe el toString con una modificacion
@Override
public String toString() {

    if(isTerrestre())

        return " Es un terrstre "+super.toString();

    else

        return " No es terrstre "+super.toString();

}
}

```

Pez.java

```

package animal;

```

```

/**

```



```

*
* @author giova
*/
//la subclase hereda de la clase principal

public class Pez extends Animal{
    //se declara la variable unica
    private int numOjos;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del mismo
    public Pez(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String tipohabitad, int numOjos) {
        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad);
        this.numOjos = numOjos;
    }

    public int getNumOjos() {
        return numOjos;
    }

    public void setNumOjos(int numOjos) {
        this.numOjos = numOjos;
    }

    //se reescribe el toString con una modificacion
    @Override
    public String toString() {
        return "Se trata un pez con " + numOjos + " ojos " + super.toString(); //To change body of generated methods, choose Tools | Templates.
    }
}

```

Reptil.java

```
package animal;
```

```

/**
 *
 * @author giova
 */
//la subclase hereda de la clase principal

public class Reptil extends Animal {

    //se declara la variable unica

    private boolean tienePatas;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del mismo

    public Reptil(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String tipohabitad, boolean tienePatas) {

        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad);

        this.tienePatas = tienePatas;

    }

    public boolean isTienePatas() {

        return tienePatas;

    }

    public void setTienePatas(boolean tienePatas) {

        this.tienePatas = tienePatas;

    }

    //se reescribe el toString con una modificacion

    @Override

    public String toString() {

        if(isTienePatas())

            return "Se trata de un reptil con patas " + super.toString(); //To change body of generated methods,
            choose Tools | Templates

        else

```

```
        return "Se trata de un reptil sin patas " + super.toString(); //To change body of generated methods,  
        choose Tools | Templates
```

```
    }  
}
```

Marsupial.java

```
package animal;
```

```
/**
```

```
 *
```

```
 * @author giova
```

```
 */
```

```
//la subclase hereda de la clase principal
```

```
public class Marsupial extends Mamifero {
```

```
    //se declara la variable unica
```

```
    private String tamano;
```

```
    private String tipoalimentacion;
```

```
    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del  
    mismo
```

```
    public Marsupial(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion,  
    String tipohabitad, boolean terrestre, String tamano, String tipoalimentacion) {
```

```
        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad, terrestre);
```

```
        this.tamano = tamano;
```

```
        this.tipoalimentacion = tipoalimentacion;
```

```
    }
```

```
    public String getTamano() {
```

```
        return tamano;
```

```
    }
```

```
    public void setTamano(String tamano) {
```

```

        this.tamano = tamano;
    }
//se reescribe el toString con una modificacion

@Override
public String toString() {
    return "Es un marsupial muy " + tamano + " y es "+ tipoalimentacion + super.toString(); //To change
body of generated methods, choose Tools | Templates.
}
}

```

Monotrema.java

```

package animal;

/**
 *
 * @author giova
 */
//la subclase hereda de la clase principal

public class Monotrema extends Mamifero {
    //se declara la variable unica
    private boolean espinas;
    private boolean ornitorinco;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del
    mismo
    public Monotrema(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion,
String tipohabitad, boolean terrestre, boolean espinas, boolean ornitorinco) {
        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad, terrestre);
        this.espinas = espinas;
        this.ornitorinco = ornitorinco;
    }
}

```

```

public boolean isEspinas() {
    return espinas;
}

public void setEspinas(boolean espinas) {
    this.espinas = espinas;
}

public boolean isOrnitorinco() {
    return ornitorinco;
}

public void setOrnitorinco(boolean ornitorinco) {
    this.ornitorinco = ornitorinco;
}

//se reescribe el toString con una modificacion
@Override
public String toString() {
    if (isEspinas()) {
        return "Tiene espinas "+super.toString();
    }else{
        if (isOrnitorinco()) {
            return "Es un ornitorinco, no hace gran cosa " + super.toString() + " ¡Es el agente P!" ;
        }else{
            return "Naaa, no tiene espinas en todo su cuerpo " + super.toString();
        }
    }
}
}
}

```

Placentario.java

```
package animal;

/**
 * @author giova
 */
//la subclase hereda de la clase principal

public class Placentario extends Mamifero {

    //se declara la variable unica
    private String tipoalimentacion;

    //se hace referencia del constructor principal y de la variable declarada y se declara el el constructor del mismo
    public Placentario(String tipoNombre, String tipoPiel, String tipoRespiracion, String tipoReproduccion, String tipohabitad, boolean terrestre, String tipoalimentacion) {

        super(tipoNombre, tipoPiel, tipoRespiracion, tipoReproduccion, tipohabitad, terrestre);

        this.tipoalimentacion = tipoalimentacion;
    }

    public String getTipoalimentacion() {

        return tipoalimentacion;
    }

    public void setTipoalimentacion(String tipoalimentacion) {

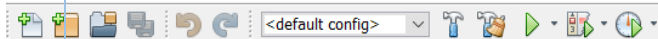
        this.tipoalimentacion = tipoalimentacion;
    }

    //se reescribe el toString con una modificacion
    @Override
    public String toString() {

        return "Por la alimentacion es un animal " + tipoalimentacion + super.toString(); //To change body of generated methods, choose Tools | Templates.
    }
}
```

RESULTADO//IMPRESIÓN DEL PROGRAMA

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help



Output - Animal (run)

run:

```
1 El color de piel es verde Nombre: rana Piel: glándulas mucosas respiracion: pulmones reproduccion: oviparo habitad: charco
2 El color de piel es cafe Nombre: sapo Piel: glándulas mucosas. respiracion: pulmones reproduccion: oviparo habitad: charco
3 El color de piel es verde Nombre: salamandra Piel: carece de escamas y es húmeda respiracion: pulmones reproduccion: oviparo habitad: charco
4 El color de piel es verde Nombre: triton Piel: aterciopelada o verrugosa respiracion: pulmones reproduccion: oviparo habitad: charco
5 El color de piel es verde Nombre: cecilia Piel: glándulas mucosas respiracion: pulmones reproduccion: oviparo habitad: charco
6 Se trata de un reptil con patas Nombre: lagarto Piel: escamas respiracion: Pulmones reproduccion: oviparo habitad: lago
7 Se trata de un reptil con patas Nombre: iguana Piel: escamas respiracion: Pulmones reproduccion: oviparo habitad: pantano
8 Se trata de un reptil con patas Nombre: cocodrilo Piel: escamas respiracion: Pulmones reproduccion: oviparo habitad: pantano
9 Se trata de un reptil con patas Nombre: tortuga Piel: escamas respiracion: Pulmones reproduccion: oviparo habitad: oceano
10 Se trata de un reptil con patas Nombre: camaleon Piel: escamas respiracion: Pulmones reproduccion: oviparo habitad: bosque
11 Se trata un pez con 2 ojos Nombre: salmon Piel: escamas respiracion: branqueas reproduccion: oviparo habitad: oceano
12 Se trata un pez con 2 ojos Nombre: tiburon Piel: escamas respiracion: branqueas reproduccion: oviparo habitad: oceano
13 Se trata un pez con 2 ojos Nombre: atun Piel: escamas respiracion: branqueas reproduccion: oviparo habitad: oceano
14 Se trata un pez con 2 ojos Nombre: anguila Piel: escamas respiracion: branqueas reproduccion: oviparo habitad: oceano
15 Se trata un pez con 2 ojos Nombre: dorado Piel: escamas respiracion: branqueas reproduccion: oviparo habitad: oceano
16 Se trata de un ave que vuela Nombre: loro Piel: plumas respiracion: pulmones reproduccion: oviparo habitad: bosque
17 Se trata de un ave que No vuela Nombre: avestruz Piel: plumas respiracion: pulmones reproduccion: oviparo habitad: montaña
18 Se trata de un ave que vuela Nombre: anguila Piel: plumas respiracion: pulmones reproduccion: oviparo habitad: desierto
19 Se trata de un ave que No vuela Nombre: pinguino Piel: plumas respiracion: pulmones reproduccion: oviparo habitad: polo sur
20 Se trata de un ave que vuela Nombre: condor Piel: plumas respiracion: pulmones reproduccion: oviparo habitad: montaña
21 Por la alimentacion es un animal herbivoro Es un terrstre Nombre: armadillo Piel: pequeñas placas óseas respiracion: pulmones reproduccion: viviparo habitad: selva
22 Por la alimentacion es un animal omnivoro Es un terrstre Nombre: oso hormiguero Piel: pelaje respiracion: pulmones reproduccion: viviparo habitad: selva
23 Es un ornitorinco, no hace gran cosa Es un terrstre Nombre: perry Piel: carne respiracion: pulmones reproduccion: oviparo habitad: lagos ¡Es el agente P!
24 Tiene espinas Es un terrstre Nombre: Equidna de hocico corto Piel: carne respiracion: pulmones reproduccion: oviparo habitad: monte
25 Es un marsupial muy chico y es herbivoro Es un terrstre Nombre: Topo Piel: aspera respiracion: pulmones reproduccion: viviparo habitad: monte
26 Es un marsupial muy grande y es herbivoro Es un terrstre Nombre: Canguro Piel: pelaje suave respiracion: pulmones reproduccion: viviparo habitad: selva
BUILD SUCCESSFUL (total time: 0 seconds)
```

DIAGRAMAS DE CLASES:

