

Programación en Lenguaje Java

Problema 8.2. Cuestiones herencia y polimorfismo



Michael González Harbour

Mario Aldea Rivas

Departamento de Matemáticas,
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Problema 8.2. Cuestiones herencia y polimorfismo

Objetivos

- Afianzar los conceptos relacionados con la herencia, el polimorfismo, los constructores y los métodos redefinidos.

Cuestión 1. Repaso herencia, constructores y métodos redefinidos.

Se dispone de la siguiente jerarquía de clases:

```
public class Abuelo {
    private int num;

    public Abuelo() {
        num = 10;
    }

    public Abuelo(int num) {
        this.num = num;
    }

    public int num() {
        return num;
    }

    public int suma() {
        return num + 1;
    }

    public int resta() {
        return num - 1;
    }
}

public class Padre extends Abuelo {
    public Padre() {
    }

    public Padre(int num) {
        super(num);
    }

    @Override
    public int suma() {
        return super.suma() + 1;
    }
}

public class Hijo extends Padre {
    @Override
    public int suma() {
        return super.suma() + 1;
    }

    @Override
    public int resta() {
        return super.resta() - 1;
    }
}
```

Indicar la salida por pantalla que produciría el programa mostrado a continuación:

```
public class Cuest1Main {
    public static void main(String[] args) {
        Abuelo a = new Abuelo();
        Padre p1 = new Padre();
        Padre p2 = new Padre(3);
        Hijo h = new Hijo();

        System.out.println("a.num:" + a.num());
        System.out.println("p1.num:" + p1.num());
        System.out.println("p2.num:" + p2.num());
        System.out.println("h.num:" + h.num());

        System.out.println("h.suma():" + h.suma());
        System.out.println("h.resta():" + h.resta());
    }
}
```

Cuestión 2. Repaso herencia y polimorfismo.

Se dispone de las siguientes clases:

```
public class Padre {
    private final int numero;

    public Padre(int numero) {
        this.numero = numero;
    }

    public int suma() {
        return numero + 1;
    }

    public int getNumero() {
        return numero;
    }
}

public class Hijo extends Padre {

    public Hijo(int numero) {
        super(numero);
    }

    @Override
    public int suma() {
        return super.suma() + 1;
    }

    public int resta() {
        return getNumero() - 1;
    }
}
```

En el programa mostrado a continuación:

- La línea correspondiente a la creación de uno de los objetos es incorrecta y produciría un error de compilación. Indicar cual es esa línea.
- Indicar la salida por consola que se produciría si se ejecuta el programa (una vez eliminadas todas las líneas que utilizaban el objeto incorrecto).

```
public static void main(String[] args) {
    Padre p1 = new Padre(2);
    Padre p2 = new Hijo(2);
    Hijo h1 = new Padre(2);
    Hijo h2 = new Hijo(2);

    System.out.println("p1.suma():" + p1.suma());
    System.out.println("p2.suma():" + p2.suma());
}
```

```
        System.out.println("h1.suma():" + h1.suma());
        System.out.println("h2.suma():" + h2.suma());
    }
```

Cuestión 3. Repaso herencia y polimorfismo.

El programa mostrado a continuación tiene una línea errónea que provocaría un error de compilación:

- Indica cual es esa línea y corrígela.
- ¿Qué salida por consola generaría el programa una vez corregido?

```
public static void main(String[] args) {
    Hijo h = new Hijo(2);
    Padre p = new Hijo(2);

    System.out.println("h.resta():" + h.resta());
    System.out.println("p.resta():" + p.resta());
}
```

(El programa utiliza las clases Padre e Hijo de la cuestión 2).

Cuestión 4. Repaso herencia y polimorfismo.

Indica la salida por consola que produciría el programa mostrado a continuación (las clases Padre e Hijo son las de la cuestión 2):

```
public static void main(String[] args) {
    ArrayList<Padre> personas = new ArrayList<Padre>();
    Hijo h = new Hijo(2);
    Padre p = new Hijo(2);
    personas.add(h);
    personas.add(p);
    personas.add(new Padre(2));
    personas.add(new Hijo(2));

    for(Padre persona: personas) {
        System.out.println("suma:" + persona.suma());
    }
}
```

Cuestión 5. Repaso herencia y polimorfismo.

El lazo `for` que aparece en el programa mostrado a continuación (usa las clases Padre e Hijo de la cuestión 2) es incorrecto. Corrígelo para que cumpla la función descrita en su comentario:

```
public static void main(String[] args) {
    ArrayList<Padre> personas = new ArrayList<Padre>();
    Padre p = new Hijo(2);

    personas.add(p);
    personas.add(new Padre(2));
    personas.add(new Hijo(2));

    // TODO: mostrar la "resta" de cada uno de los objetos de
    // clase Hijo en el ArrayList
}
```

```
        for(Padre persona: personas) {
            System.out.println("resta:" + persona.resta());
        }
    }
}
```

Cuestión 6. Repaso herencia y polimorfismo.

Se dispone de la siguiente jerarquía de clases:

```
public class Abuelo {
    String s;

    public Abuelo() {
        s = "Sin nombre";
    }

    public Abuelo(String s) {
        this.s = s;
    }

    public String toString() {
        return " s:" + s;
    }
}

public class Padre extends Abuelo {
    public Padre() {
    }

    public Padre(String s) {
        super(s);
    }

    public String toString(int i) {
        return "padre" + super.toString();
    }
}

public class Hijo extends Padre{
    private int i;

    public Hijo(int i) {
        this.i = i;
    }

    public String toString() {
        return super.toString() + ",i:" + i;
    }
}
```

En el programa mostrado a continuación:

- Hay una línea que produce un error de compilación. Indicar cual es esa línea.
- Indicar la salida por consola que se produciría si se ejecuta el programa (con la línea incorrecta comentada).

```
public class Herencia {  
  
    public static void main(String[] args) {  
        Hijo h = new Hijo(1);  
        Padre p = new Padre("padre");  
        Abuelo a1 = new Abuelo("abuelo");  
        Padre p2 = new Abuelo("abuelo2");  
        Abuelo a2 = new Hijo(2);  
  
        System.out.println("h:" + h);  
        System.out.println("p:" + p);  
        System.out.println("a1:" + a1);  
        System.out.println("a2:" + a2);  
        System.out.println("h:" + (Abuelo)h);  
    }  
}
```