

Resolución Práctico 3: introducción a la programación orientada a objetos

Ejercicio 1: en este ejercicio debemos crear una clase Estudiante con los atributos nombre, apellido, curso y calificación, con 3 métodos uno para mostrar la información del estudiante, otro para subir la calificación y otro para bajar la calificación, los dos últimos debían recibir un parámetro “puntos”. Luego desde el main debíamos crear un objeto estudiante con los atributos y utilizar sus métodos.

Codigo:

Parte 1: Declaracion de atributos y metodos getter y setter.

```
public class Estudiante {  
    // Declaramos atributos privados  
    private String nombre;  
    private String apellido;  
    private String curso;  
    private double calificacion;  
  
    // Declaramos getter y setter de cada atributo para poder acceder a ellos  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getApellido() {  
        return apellido;  
    }  
  
    public void setApellido(String apellido) {  
        this.apellido = apellido;  
    }  
  
    public String getCurso() {  
        return curso;  
    }  
  
    public void setCurso(String curso) {  
        this.curso = curso;  
    }  
  
    public double getCalificacion() {  
        return calificacion;  
    }  
  
    public void setCalificacion(double calificacion) {  
        this.calificacion = calificacion;  
    }  
}
```

Parte 2: creación de métodos.

```
// Metodo para ver la info de un estudiante
public void mostrarInfo(){
    System.out.println("Nombre: " + this.nombre + "\n Apellido: " + this.apellido + "\n Curso: " + this.curso + "\n Calificacion: " + this.calificacion);
}

// Metodo para subir la calificacion de un estudiante
public void subirCalificacion(double puntos){
    this.calificacion = this.calificacion + puntos;
}

// Metodo para bajar la calificacion de un estudiante
public void bajarCalificaion(double puntos){
    this.calificacion = this.calificacion - puntos;
}
```

Main:

```
public static void main(String[] args) {
    // Creamos objeto estudiante y le asignamos valores a sus atributos
    Estudiante e = new Estudiante();
    e.setNombre("Juan");
    e.setApellido("Perez");
    e.setCurso("Comision 1");
    e.setCalificacion(8.0);

    // Llamamos al metodo mostrar info
    e.mostrarInfo();

    e.subirCalificacion(2.0); // Subimos dos puntos su calificacion
    System.out.println(""); // Salto de linea para que veamos la consola mas prolija
    e.mostrarInfo(); // Verificamos que se haya subido la calificacion

    e.bajarCalificaion(1.0); // Bajamos un punto la calificacion
    System.out.println(""); // Salto de linea para que veamos la consola mas prolija
    e.mostrarInfo(); // Verificamos que se haya bajado la calificacion
}
```

Salida por pantalla:

```
Nombre: Juan
Apellido: Perez
Curso: Comision 1
Calificacion: 8.0
```

```
Nombre: Juan
Apellido: Perez
Curso: Comision 1
Calificacion: 10.0
```

```
Nombre: Juan
Apellido: Perez
Curso: Comision 1
Calificacion: 9.0
```

Ejercicio 2: en ejercicio kata debíamos crear una clase mascota con nombre, especie y edad como atributos, luego crear dos métodos uno para mostrar la información de la mascota y otro para envejecer la mascota un año.

Codigo:

Clase mascota:

```
public class Mascota {
    // Declaramos atributos privados
    private String nombre;
    private String especie;
    private int edad;

    // Declaramos getter y setter de cada atributo para poder acceder a ellos
    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getEspecie() {
        return especie;
    }

    public void setEspecie(String especie) {
        this.especie = especie;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }

    // Mostramos info de la mascota
    public void mostrarInfo() {
        System.out.println("Nombre: " + this.nombre + "\n Especie: " + this.especie + "\n Edad: " + this.edad);
    }

    // Metodo para cumplir un año
    public void cumplirAnios() {
        this.edad += 1;
    }
}
```

Main:

```
public class Main {

    public static void main(String[] args) {
        // Creamos un objeto mascota y le asignamos valores a sus atributos
        Mascota m = new Mascota();
        m.setNombre("Luna");
        m.setEspecie("Perro");
        m.setEdad(5);

        // Mostramos su informacion
        m.mostrarInfo();
        m.cumplirAnios(); // Subimos 1 año
        m.cumplirAnios(); // Subimos 1 año
        System.out.println("");
        // Mostramos info despues de que la mascota envejecio 2 años
        m.mostrarInfo();
    }
}
```

Salida por pantalla:

```
Nombre: Luna  
Especie: Perro  
Edad: 5
```

```
Nombre: Luna  
Especie: Perro  
Edad: 7
```

Ejercicio 3: en este ejercicio debíamos crear un clase libro con los atributos titulo, autor, anioPublicacion y un método para mostrar su información. Luego en el main debíamos crear un objeto libro primero seteandole un anioPublicacion invalido y mostrando la información, luego seteandole un anioPublicacion valido y mostrando la información nuevamente.

Codigo:

Clase libro:

```
public class Libro {  
    // Declaramos atributos privados  
    private String titulo;  
    private String autor;  
    private int anioPublicacion;  
  
    // Declaramos getter y setter para cada atributo  
    public String getTitulo() {  
        return titulo;  
    }  
  
    public void setTitulo(String titulo) {  
        this.titulo = titulo;  
    }  
  
    public String getAutor() {  
        return autor;  
    }  
  
    public void setAutor(String autor) {  
        this.autor = autor;  
    }  
  
    public int getAnioPublicacion() {  
        return anioPublicacion;  
    }  
  
    public void setAnioPublicacion(int anioPublicacion) {  
        int anioActual = 2025;  
        if (anioPublicacion > 0 && anioPublicacion <= anioActual) {  
            this.anioPublicacion = anioPublicacion;  
        }  
    }  
  
    // Metodo para mostrar info de un libro  
    public void mostrarInfo() {  
        System.out.println("Titulo: " + this.titulo + "\n Autor: " + this.autor + "\n Año de publicacion: " + this.anioPublicacion);  
    }  
}
```

Main:

```
public class Main {  
  
    public static void main(String[] args) {  
        // Creamos un objeto libro  
        Libro l = new Libro();  
  
        // Le asignamos valores a cada atributo  
        l.setTitulo("Cien años de soledad");  
        l.setAutor("Gabriel García Márquez");  
        l.setAñoPublicacion(2026); // Seteamos un año no valido  
  
        // Mostramos la informacion habiendo seteado un año invalido  
        l.mostrarInfo();  
        System.out.println(""); // Salto de linea para ver mas prolija la consola  
        l.setAñoPublicacion(1967); // Seteamos un año valido  
        l.mostrarInfo(); // Mostramos la informacion con todos los datos cargados correctamente  
    }  
}
```

Salida por pantalla:

```
Titulo: Cien años de soledad  
Autor: Gabriel García Márquez  
Año de publicacion: 0
```

```
Titulo: Cien años de soledad  
Autor: Gabriel García Márquez  
Año de publicacion: 1967
```

Ejercicio 4: en este ejercicio debíamos crear una clase gallina con 3 atributos (idGallina, edad y huevosPuestos), y 3 metodos(ponerHuevo, envejecer y mostrar estado). Luego desde el main debíamos crear dos objetos gallina, y realizar las acciones de los metodos mostrando su estado.

Codigo:

Clase Gallina:

```
public class Gallina {  
    // Declaramos atributos privados  
    private String idGallina;  
    private int edad;  
    private int huevosPuestos;  
  
    // Declaramos los metodosos getter y setter  
    public String getIdGallina() {  
        return idGallina;  
    }  
  
    public void setIdGallina(String idGallina) {  
        this.idGallina = idGallina;  
    }  
  
    public int getEdad() {  
        return edad;  
    }  
  
    public void setEdad(int edad) {  
        this.edad = edad;  
    }  
  
    public int getHuevosPuestos() {  
        return huevosPuestos;  
    }  
  
    public void setHuevosPuestos(int huevosPuestos) {  
        this.huevosPuestos = huevosPuestos;  
    }  
  
    // Metodo para que la gallina ponga un huevo  
    public void ponerHuevo(){  
        this.huevosPuestos += 1;  
    }  
  
    // Metodo para que la gallina envejezca un año  
    public void envejecer() {  
        this.edad += 1;  
    }  
  
    // Mostramos el estado de la gallina  
    public void mostrarEstado() {  
        System.out.println("ID: " + this.idGallina + ", huevos puestos: " + this.huevosPuestos + ", edad: " + this.edad);  
    }  
}
```

Main:

```
public class Main {  
  
    public static void main(String[] args) {  
        // Creamos un objeto gallina y le asignamos valores a sus atributos  
        Gallina g = new Gallina();  
        g.setIdGallina("abc123");  
        g.setEdad(1);  
        g.setHuevosPuestos(0);  
  
        Gallina g2 = new Gallina();  
        g2.setIdGallina("abc124");  
        g2.setEdad(2);  
        g2.setHuevosPuestos(5);  
  
        // Mostramos el estado inicial de las gallinas  
        g.mostrarEstado();  
        g2.mostrarEstado();  
  
        // Llamamos 3 veces al metodo ponerHuevo de cada gallina  
        g.ponerHuevo();  
        g.ponerHuevo();  
        g.ponerHuevo();  
  
        g2.ponerHuevo();  
        g2.ponerHuevo();  
        g2.ponerHuevo();  
  
        // Llamamos una vez al metodo envejecer de cada gallina  
        g.envejecer();  
        g2.envejecer();  
  
        // Mostramos el nuevo estado de las gallinas  
        g.mostrarEstado();  
        g2.mostrarEstado();  
    }  
}
```

Salida por pantalla:

```
ID: abc123, huevos puestos: 0, edad: 1  
ID: abc124, huevos puestos: 5, edad: 2  
ID: abc123, huevos puestos: 3, edad: 2  
ID: abc124, huevos puestos: 8, edad: 3
```

Ejercicio 5: En este ejercicio debíamos crear una clase NaveEspacial con 2 atributos (nombre y combustible), y 4 métodos (despegar, avanzar, recargarCombustible y mostrarEstado). Luego desde el main debíamos crear un objeto nave con 50 unidades de combustible, intentar avanzar sin recargar, luego recargar y avanzar, y finalmente mostrar el estado de la nave.

Codigo:

Clase nave espacial:

Atributos y metodos getter y setter:

```
public class NaveEspacial {  
    // Declaramos atributos privados y metodos getter y setter  
    private String nombre;  
    private int combustible;  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public int getCombustible() {  
        return combustible;  
    }  
  
    public void setCombustible(int combustible) {  
        this.combustible = combustible;  
    }  
}
```

Metodos despegar, avanzar, recargar combustible y mostrar estado:

```
// Metodo despegar  
public void despegar(){  
    if(this.combustible >= 5) { // Verificamos que la nave tenga suficiente combustible para despegar  
        System.out.println("La nave despegó...");  
        this.combustible -= 5; // Si despegó se le restan las unidades necesarias para su despegue  
    } else {  
        System.out.println("No hay suficiente combustible");  
    }  
}  
  
// Metodo avanzar, recibe una distancia como parametro  
public void avanzar(int distancia) {  
    if((distancia * 2) <= this.combustible) { // Cada km consumo 2 unidades de combustible verificamos que la nave tenga suficiente para la distancia  
        System.out.println("La nave avanza " + distancia + " kms");  
        this.combustible -= (distancia * 2); // En el caso que haya podido avanzar se le resta el combustible dedicado a los kms  
    } else {  
        System.out.println("No hay suficiente combustible");  
    }  
}  
  
// Metodo recarga de combustible, recibe cantidad como parametro  
public void recargarCombustible(int cantidad){  
    int maximo = 100; // Declaramos un maximo  
    if((cantidad + this.combustible) <= maximo) { // Verificamos que el combustible no supere al maximo de la nave  
        this.combustible += cantidad; // Si no lo supera se realiza la recarga  
    } else {  
        System.out.println("Ha ingresado demasiado combustible");  
    }  
}  
  
// Metodo para mostrar el estado de la nave  
public void mostrarEstado(){  
    System.out.println("Nombre: " + this.nombre + ", combustible: " + this.combustible);  
}
```


Main:

```
public class Main {  
  
    public static void main(String[] args) {  
        NaveEspacial nave = new NaveEspacial();  
        nave.setNombre("Exploradora 1");  
        nave.setCombustible(50);  
  
        System.out.println("Estado inicial:");  
        nave.mostrarEstado();  
        nave.despegar();  
        nave.mostrarEstado();  
  
        System.out.println("Intentando avanzar 30 km:");  
        nave.avanzar(30);  
  
        System.out.println("Recargando 40 unidades de combustible:");  
        nave.recargarCombustible(30);  
  
        System.out.println("Intentando avanzar 20 km:");  
        nave.avanzar(20);  
  
        System.out.println("Estado final:");  
        nave.mostrarEstado();  
    }  
}
```

Salida por pantalla:

```
Estado inicial:  
Nombre: Exploradora 1, combustible: 50  
La nave despegó...  
Nombre: Exploradora 1, combustible: 45  
Intentando avanzar 30 km:  
No hay suficiente combustible  
Recargando 40 unidades de combustible:  
Intentando avanzar 20 km:  
La nave avanzó 20 kms  
Estado final:  
Nombre: Exploradora 1, combustible: 35
```