



Ejercicio 12

Desarrollar la clase `ImpresoraMonocromatica` . Sus atributos serán si está o no encendida, la cantidad de hojas actualmente en su bandeja y el nivel de tinta negra. Inicialmente, toda impresora está apagada, sin hojas y con nivel de tinta en **100**.

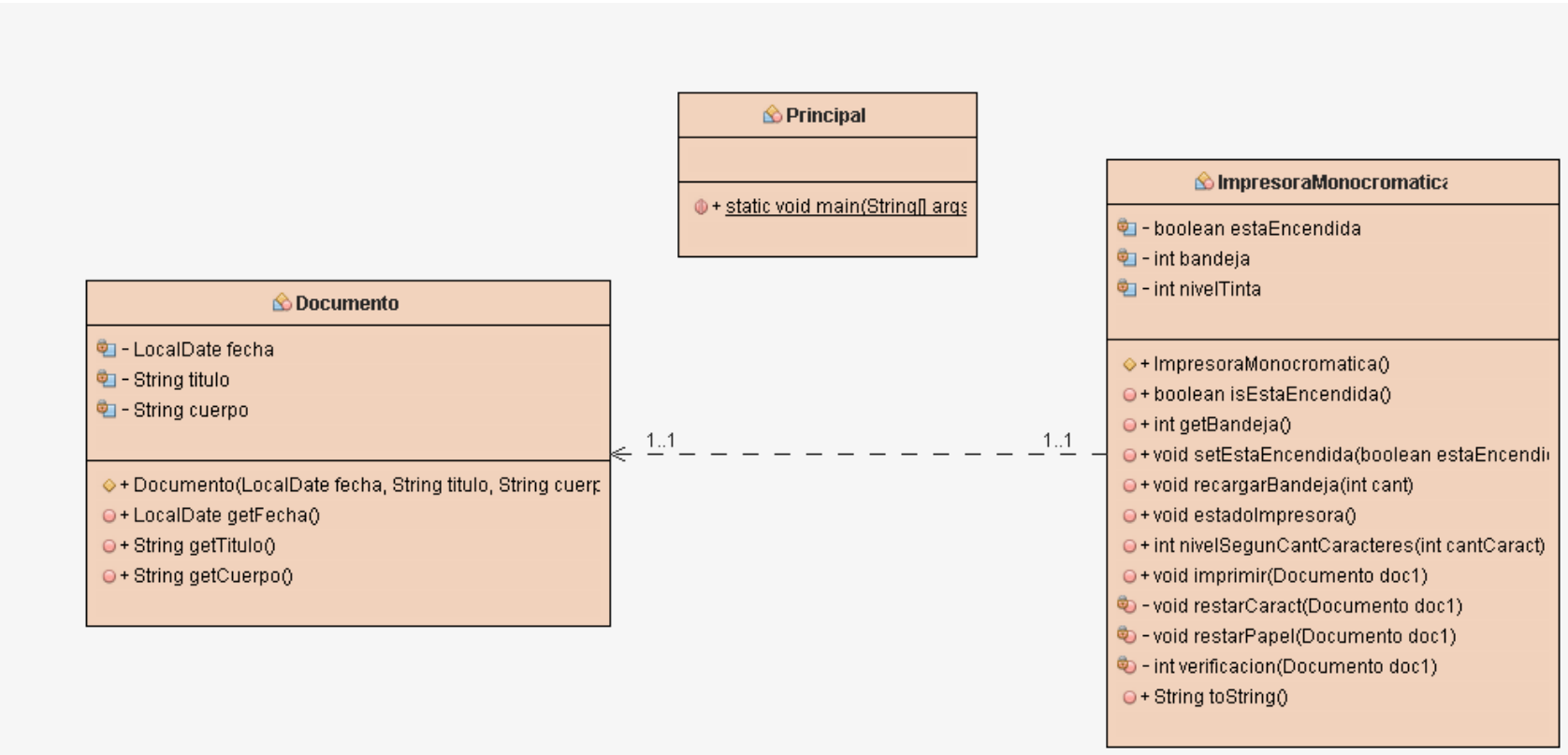
Debe responder a los siguientes métodos:

- `int nivelSegunCantCaracteres(int) {...}`
Devuelve cuánta cantidad de tinta debería usarse según la cantidad de caracteres recibida por parámetro.
- `void recargarBandeja(int) {...}`
Suma a la bandeja una cantidad de hojas. El máximo de la bandeja es de 35 hojas. Se debe verificar no excederse de tal valor. Si el parámetro es negativo, la bandeja se deja como está.
- `void imprimir(Documento) {...}`
Imprime en consola la fecha del documento (Modela la clase correspondiente), junto a su título y cuerpo, en el siguiente formato:

Fecha	**Titulo**
Cuerpo	

Al hacerlo, se descuenta 1 punto de nivel de tinta por cada 50 caracteres del cuerpo impresos y se resta 1 hoja de la cantidad en bandeja por cada 20 caracteres del cuerpo impreso. Se debe verificar antes de imprimir que se cuente con nivel de tinta y cantidad de hojas suficientes.

Diagrama UML:



Proyecto Java:

```
package ejercicio_12;

public class ImpresoraMonocromatica {
    private boolean estaEncendida;
    private int bandeja;
    private int nivelTinta;

    //CONSTRUCTOR
    public ImpresoraMonocromatica() {
        this.estaEncendida=false;
        this.bandeja=0; // la bandeja empieza con 0 hojas
        this.nivelTinta=100;
    }
    //GETTERS
```

```

    public boolean isEstaEncendida() {
        return estaEncendida;
    }
    public int getBandeja() {
        return bandeja;
    }
    //SETTERS
    public void setEstaEncendida(boolean estaEncendida) {
        this.estaEncendida = estaEncendida;
    }
    public void recargarBandeja(int cant){
        if(bandeja<=35 && cant > 0 && cant <= 35){
            bandeja = bandeja + cant;
        }
    }
    // METODOS
    public void estadoImpresora(){
        System.out.println(toString());
    }

    //se consume 1 punto de tinta cada 50 caracteres
    public int nivelSegunCantCaracteres(int cantCaract){
        return cantCaract/50;
    }

    public void imprimir(Documento doc1) throws Exception {
        if(verificacion(doc1)==1){
            restarCaract(doc1);
            restarPapel(doc1);
            // Imprime la fecha
            System.out.print("\n"+doc1.getFecha() + "\t");
            // Imprime el titulo
            System.out.println(doc1.getTitulo());
            // Imprime el cuerpo
            System.out.println(doc1.getCuerpo());
        }
    }

    private void restarCaract(Documento doc1){
        int cantCaract = doc1.getCuerpo().length();
        nivelTinta -= nivelSegunCantCaracteres(cantCaract);
    }

    private void restarPapel(Documento doc1){
        int cantCaract = doc1.getCuerpo().length();
        bandeja-=cantCaract/20;
    }

    private int verificacion(Documento doc1) throws Exception {
        int cantCaract = doc1.getCuerpo().length();
        if (estaEncendida == true) {
            if (nivelTinta >= nivelSegunCantCaracteres(cantCaract)) {
                if (bandeja > 0) {
                    return 1;
                } else {
                    throw new Exception("La bandeja de la impresora está vacía.");
                }
            } else {
                throw new Exception("El nivel de tinta no es suficiente para imprimir este documento.");
            }
        } else {
            throw new Exception("La impresora está apagada.");
        }
    }
}

@Override
public String toString() {
    return "ImpresoraMonocromatica{" + "estaEncendida=" + estaEncendida + ", bandeja=" + bandeja + ", nivelTinta=" + nivelTinta + '}';
}
}

```

```

package ejercicio_12;

import java.time.LocalDate;

public class Documento {
    private LocalDate fecha;
    private String titulo;
    private String cuerpo;

    //CONSTRUCTOR
    public Documento(LocalDate fecha, String titulo, String cuerpo) {
        this.fecha = fecha;
        this.titulo = titulo;
        this.cuerpo = cuerpo;
    }

    public LocalDate getFecha() {
        return fecha;
    }

    public String getTitulo() {
        return titulo;
    }
}

```

```
    }

    public String getCuerpo() {
        return cuerpo;
    }
}
```

```
package ejercicio_12;

import java.time.LocalDate;

public class Principal {

    public static void main(String[] args) throws Exception {
        ImpresoraMonocromatica impresora= new ImpresoraMonocromatica();
        //veo el estado de la impresora
        impresora.estadoImpresora();
        //prendo impresora
        impresora.setEstaEncendida(true);
        //cargo papel
        impresora.recargarBandeja(35); // recargo hojas al tope
        System.out.println("bandeja: "+ impresora.getBandeja());
        // verificar cant de tinta según cant de caracteres
        System.out.println("nivel de tinta que debería usarse " + impresora.nivelSegunCantCaracteres(100));

        // creo el documento a imprimir
        String titulo = "Messi Goat";
        String campo = "Messi es el mejor futbolista de todos los tiempos.";
        Documento doc1= new Documento(LocalDate.now(),titulo,campo);
        impresora.imprimir(doc1);

        //veo el estado de la impresora (post impresion)
        impresora.estadoImpresora();
    }
}
```



Explicación del programa:

El proyecto consiste en la implementación de una impresora monocromática en Java. La clase `ImpresoraMonocromatica` cuenta con atributos como `estaEncendida`, `bandeja` y `nivelTinta`. También se han definido algunos métodos como `recargarBandeja`, `nivelSegunCantCaracteres`, `restarCaract`, `restarPapel`, `verificacion`, `estadoImpresora` e `imprimir`.

La clase `Documento` define los atributos de un documento como `fecha`, `titulo` y `cuerpo`.

La clase `Principal` permite probar el funcionamiento de la impresora, donde se enciende la impresora, se carga papel, se crea un documento y se imprime en la impresora. Además, muestra el estado de la impresora antes y después de la impresión.

El proyecto está diseñado para manejar la impresión de un solo documento a la vez, pero se puede crear múltiples documentos y usar el método `imprimir` para imprimirlos en la impresora.