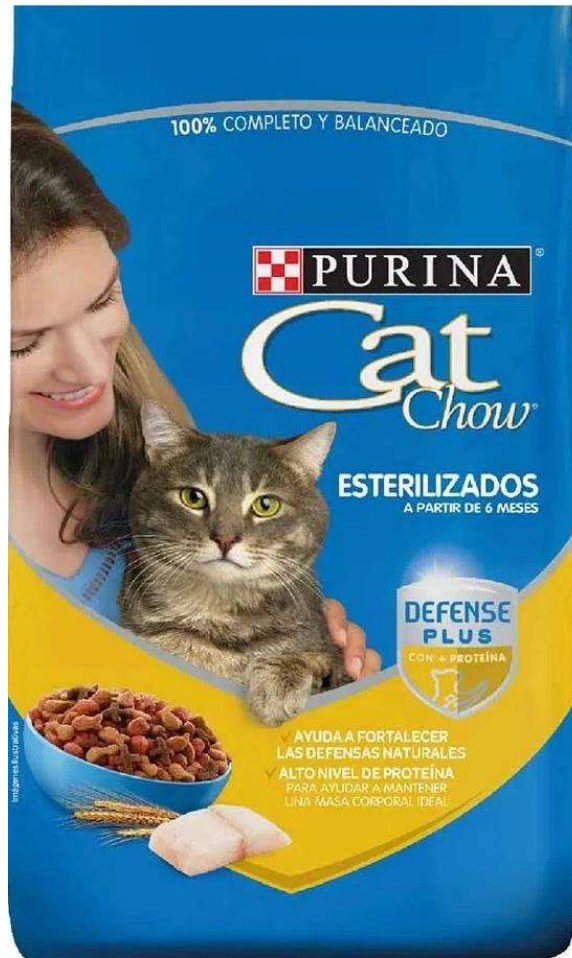


Los Cat Chow – Practica 04



Fuentes Anica Cesar Candelario –
321242927

Solares Ramos Luis Mario –
321057110

Suárez Barrera Mauricio – 313226131

Patrones de diseño:

- Proxy: Este patrón de diseño se utilizó para la problemática de una sola impresora dentro de la empresa, en la que la impresora la tomamos como el “server” y al cliente lo tomamos como los como los empleados, usando una cola como estructura de dato para la lista de solicitudes de impresiones que llegan, sin darle importancia a algún empleado en particular.
- Singleton: Este patrón de diseño nos asegura que una clase (en nuestro caso “Print”) solo tenga una instancia (es decir una sola impresora), ya que como podemos observar en la práctica nos dice que solo hay una impresora en toda la empresa, así todos pueden acceder a esta impresora sin importar el área del usuario.

En nuestro código se presenta que al no ser un trabajador del área de mercadotecnia solo se pueda imprimir a blanco y negro, es decir cualquier trabajador que no sea del piso 4 no puede imprimir a color, con una excepción, en el piso 4 tenemos dos áreas “Mercadotecnia” y “Desarrollo”, teniendo a Desarrollo como un área de las dos que se encuentran en ese piso que no puede imprimir a color.

Estructura de los directorios:

En el directorio src se encuentran los archivos .java, el cual se divide en dos carpetas “singleton” “ui”.

A la misma altura se encuentran los dos diagramas, el de clases y el de casos de uso.

Compilación y ejecución del código:

Compila con:

1. javac -d build -cp build .\src\Client.java
2. javac -d build -cp build .\src\singleton\RequestPrint.java
3. javac -d build -cp build .\src\singleton\Print.java
4. javac -d build -cp build .\src\Server.java

Ejecuta con:

En una terminal despues de compilar escriba `<java -cp build src/ui/SimulationE.java server>` y en la segunda terminal `< java -cp build src/ui/SimulationE.java client>`

Notas del código: A la hora de ejecutar `java -cp build src/ui/SimulationE.java server` y en otra terminal `java -cp build src/ui/SimulationE.java client` aparece en la primera terminal:

```
Java.io.EOFException                                     at
java.base/java.io.ObjectInputStream$BlockDataInputStream.peekByte(ObjectInp
utStream.java:3241)
```

```
    at
java.base/java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1701)
```

```
    at
java.base/java.io.ObjectInputStream.readObject(ObjectInputStream.java:528)
```

```
    at
java.base/java.io.ObjectInputStream.readObject(ObjectInputStream.java:486)
```

```
        at src.Server.receiveRequest(Server.java:64)
```

```
        at src.ui.SimulationE.startServer(SimulationE.java:36)
```

```
        at src.ui.SimulationE.main(SimulationE.java:19)
```

```
    at
java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethod
HandleAccessor.java:103)
```

```
        at java.base/java.lang.reflect.Method.invoke(Method.java:580)
```

```
    at
jdk.compiler/com.sun.tools.javac.launcher.SourceLauncher.execute(SourceLaun
cher.java:268)
```

```
    at
jdk.compiler/com.sun.tools.javac.launcher.SourceLauncher.run(SourceLauncher
.java:153)
```

```
    at
jdk.compiler/com.sun.tools.javac.launcher.SourceLauncher.main(SourceLaunch
er.java:78)
```

Verificando el tipo de impresion

Exception in thread "main" java.lang.NullPointerException: Cannot invoke
"src.ProxyPrinter.allowColorPrints()" because "<local4>" is null

```
    at src.ui.SimulationE.startServer(SimulationE.java:38)
```

```
at src.ui.SimulationE.main(SimulationE.java:19)
```

Segunda nota: El servidor si recibe las cosas pero no las manda.