



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FCFM

UNIDAD DE APRENDIZAJE: LABORATORIO DE DISEÑO ORIENTADO A
OBJETOS.

PRACTICA #10

PROFESOR. MIGUEL ÁNGEL SALAZAR S.

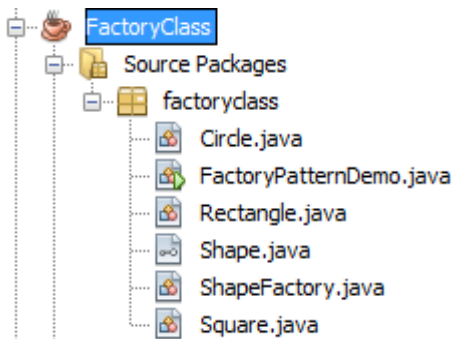
ALUMNA. VALERIA MARTÍNEZ DE LA ROSA

MATRICULA. 1678575

San Nicolás de los garza, nuevo león a 20 de abril de 2017

Estructura

Al finalizar este proyecto deberá lucir así.



Interface

Primero se hará la clase interfaz pública, llamada Shape. Esta clase tiene un método abstracto llamado drawing(), que no devuelve nada, está por ende no tiene firma porque es abstracta, lo que quiere decir que es solamente "la cabeza", en cada clase donde se implemente este método se definirá una firma diferente. En este método no declaramos el modificador de acceso porque este es default, que no afecta el declararlo o no.

```
public interface Shape {
    void drawing();
}
```

Java class

Rectangle: esta clase será pública y será implementada por la clase Shape(que quiere decir que hará uso de los métodos de esta). Se escribe @Override, que indica que se sobre-escribirá el método de la clase Shape. Después de sobre-escribir el método, en su firma se añadirá un System.out.print(" "); que imprime o muestra lo que este entre las comillas, en el print, puedes agregar \n, que hace un salto de línea para no amontonar.

```
public class Rectangle implements Shape{
    @Override
    public void drawing(){
        System.out.println("method drawing in rectangle");
    }
}
```

Se hace lo mismo en las siguientes 2 clases, llamadas Circle, y Square.

```
public class Square implements Shape{
    @Override
    public void drawing(){
        System.out.println("method drawing in square");
    }
}
```

```
public class Circle implements Shape{
    @Override
    public void drawing(){
        System.out.println("method drawing in circle");
    }
}
```

ShapeFactory: la clase es pública, y tiene un método de tipo Shape llamado getShape que tiene como parámetro una cadena de caracteres llamada shapeType. Y dentro de este método hace lo siguiente:

Compara si el shapeType es nulo, de ser así, devuelve un nulo.

Compara de nuevo si shapeType es igual a "CIRCLE", de ser así crea y devuelve un objeto Circle(), sino compara si shapeType es igual a "RECTANGLE", de ser así crea y devuelve un objeto Rectangle(), sino compara si shapeType es igual a "SQUARE", de ser así crea y devuelve un objeto Square(), sino devuelve un nulo.

```

public class ShapeFactory {
    public Shape getShape(String shapeType){
        if(shapeType == null){
            return null;
        }
        if(shapeType.equals("CIRCLE")){
            return new Circle();
        }
        else if (shapeType.equals("RECTANGLE")){
            return new Rectangle();
        }
        else if (shapeType.equals("SQUARE")){
            return new Square();
        }
        return null;
    }
}

```

FactoryPatternDemo: esta es la clase app, por lo que debe de tener un método principal, qué sea público, estático y no devuelva nada, como parámetro deberá tener un arreglo de Strings,(o de objetos).

Se crea una variable ShapeFactory, dónde se guardan únicamente objetos del mismo.

Se crea una variable Shape dónde se guarda el método getShape() con parámetro CIRCLE, y hace llamar al método drawing(), después igual pero con parámetro RECTANGLE, SQUARE.

```

public class FactoryPatternDemo {
    public static void main(String[] args){

        ShapeFactory sf = new ShapeFactory();

        Shape s1 = sf.getShape("CIRCLE");
        s1.drawing();

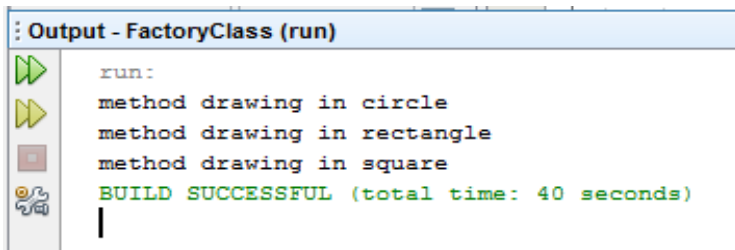
        Shape s2 = sf.getShape("RECTANGLE");
        s2.drawing();

        Shape s3 = sf.getShape("SQUARE");
        s3.drawing();

    }
}

```

Cuando corres tu programa deberá mostrar lo siguiente.



The screenshot shows the 'Output - FactoryClass (run)' window in an IDE. It contains the following text:

```

run:
method drawing in circle
method drawing in rectangle
method drawing in square
BUILD SUCCESSFUL (total time: 40 seconds)
|

```

On the left side of the output window, there are several icons: a green play button, a yellow play button, a red stop button, and a magnifying glass icon.