



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS



Laboratorio de Diseño Orientado a Objetos

Armando Rodrigo Botello Alanis

Grupo: 009

Matrícula: 1722884

Práctica de Laboratorio #10

Patrón Creacional Factory

Maestro: Miguel Salazar

Día: 23 de abril de 2017

Contenido

Objetivo 3

Pasos..... 3

Objetivo

Al finalizar las actividades de este laboratorio, deberás ser capaz de comprender para que sirve un patrón de Diseño Factory

Pasos

El patrón Factory es un patrón de diseño creacional y que sirve para construir una jerarquía de clases.

Paso 1. Iniciando con un Interface Shape

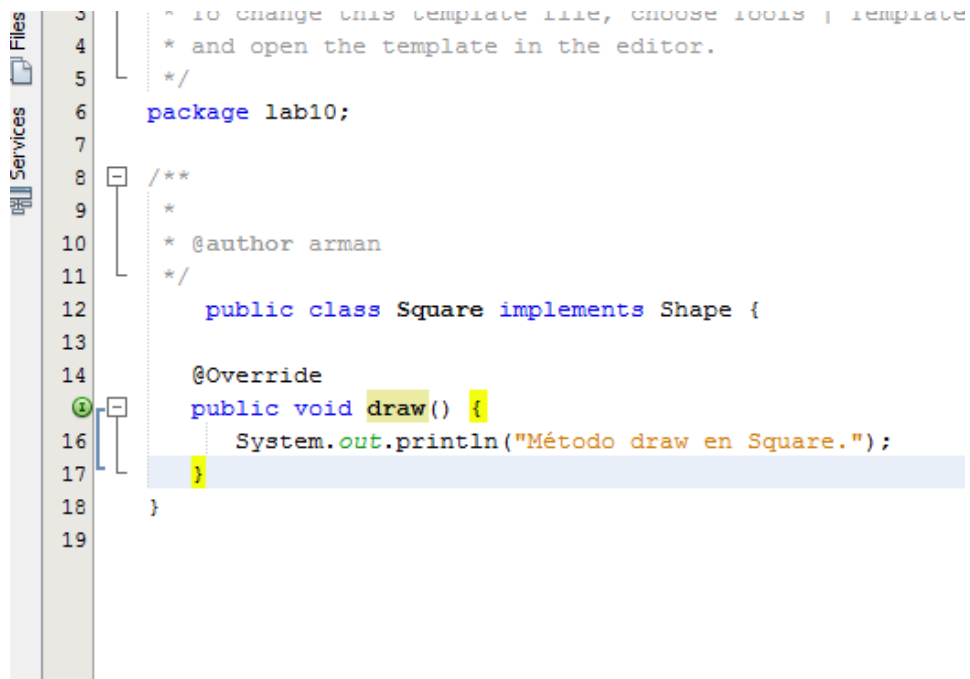
```
1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package lab10;
7
8   /**
9    *
10   * @author arman
11   */
12
13   public interface Shape {
14       void draw();
15   }
```

Paso 2. Creando cada figura (La subcategoría) deseada

Iniciando con Rectángulo:

```
6   package lab10;
7
8   /**
9    *
10   * @author arman
11   */
12   public class Rectangle implements Shape {
13
14       @Override
15       public void draw() {
16           System.out.println("Método draw en Rectangle");
17       }
18   }
```

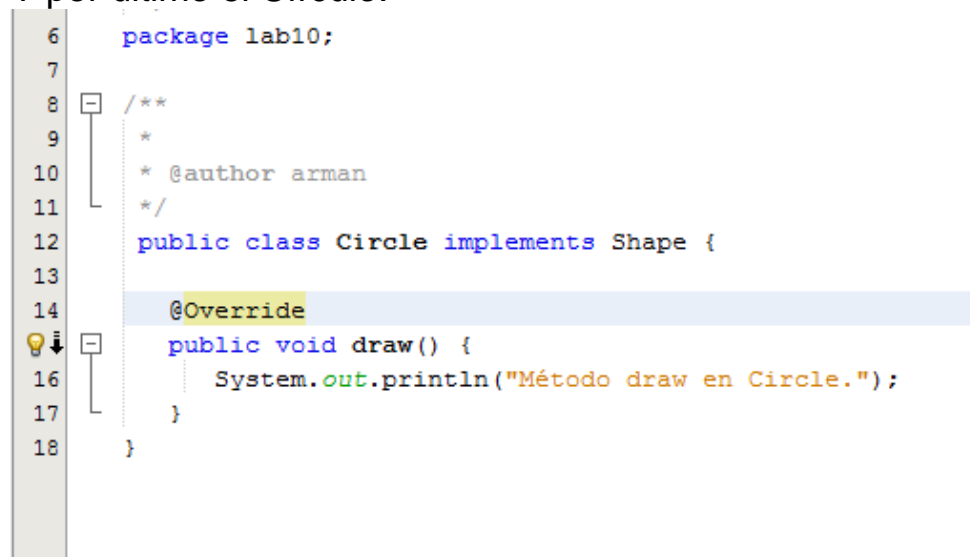
Después con el Cuadrado:



The screenshot shows an IDE with a file explorer on the left containing 'Files' and 'Services' tabs. The main editor displays a Java file with the following code:

```
3  ~ to change this template file, choose tools | template
4  * and open the template in the editor.
5  */
6  package lab10;
7
8  /**
9   *
10  * @author arman
11  */
12  public class Square implements Shape {
13
14      @Override
15      public void draw() {
16          System.out.println("Método draw en Square.");
17      }
18  }
19
```

Y por último el Círculo:

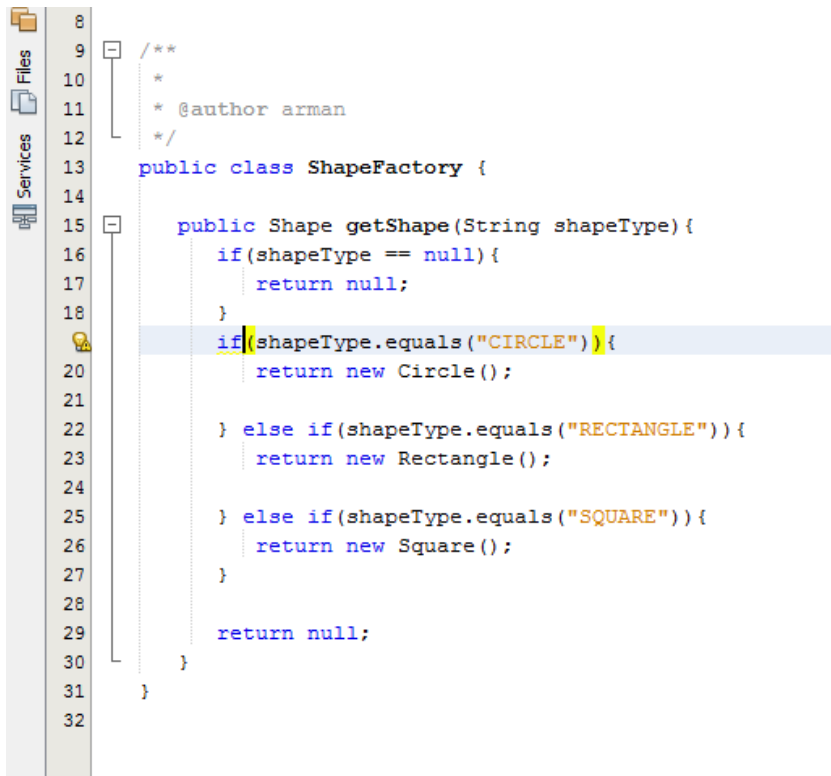


The screenshot shows an IDE with a file explorer on the left containing 'Files' and 'Services' tabs. The main editor displays a Java file with the following code:

```
6  package lab10;
7
8  /**
9   *
10  * @author arman
11  */
12  public class Circle implements Shape {
13
14      @Override
15      public void draw() {
16          System.out.println("Método draw en Circle.");
17      }
18  }
```

Paso 3. Crear el Factory

Aquí se crea la clase factory para la construcción de los objetos de las clases concretas en base a una decisión.



```
8
9  /**
10   *
11   * @author arman
12   */
13  public class ShapeFactory {
14
15      public Shape getShape(String shapeType){
16          if(shapeType == null){
17              return null;
18          }
19          if(shapeType.equals("CIRCLE")){
20              return new Circle();
21          }
22          else if(shapeType.equals("RECTANGLE")){
23              return new Rectangle();
24          }
25          else if(shapeType.equals("SQUARE")){
26              return new Square();
27          }
28
29          return null;
30      }
31  }
32
```

Paso 4. El cliente interactúa con ShapeFactory



```
8
9  /**
10   *
11   * @author arman
12   */
13  public class ShapeFactory {
14
15      public Shape getShape(String shapeType){
16          if(shapeType == null){
17              return null;
18          }
19          if(shapeType.equals("CIRCLE")){
20              return new Circle();
21          }
22          else if(shapeType.equals("RECTANGLE")){
23              return new Rectangle();
24          }
25          else if(shapeType.equals("SQUARE")){
26              return new Square();
27          }
28
29          return null;
30      }
31  }
32
```