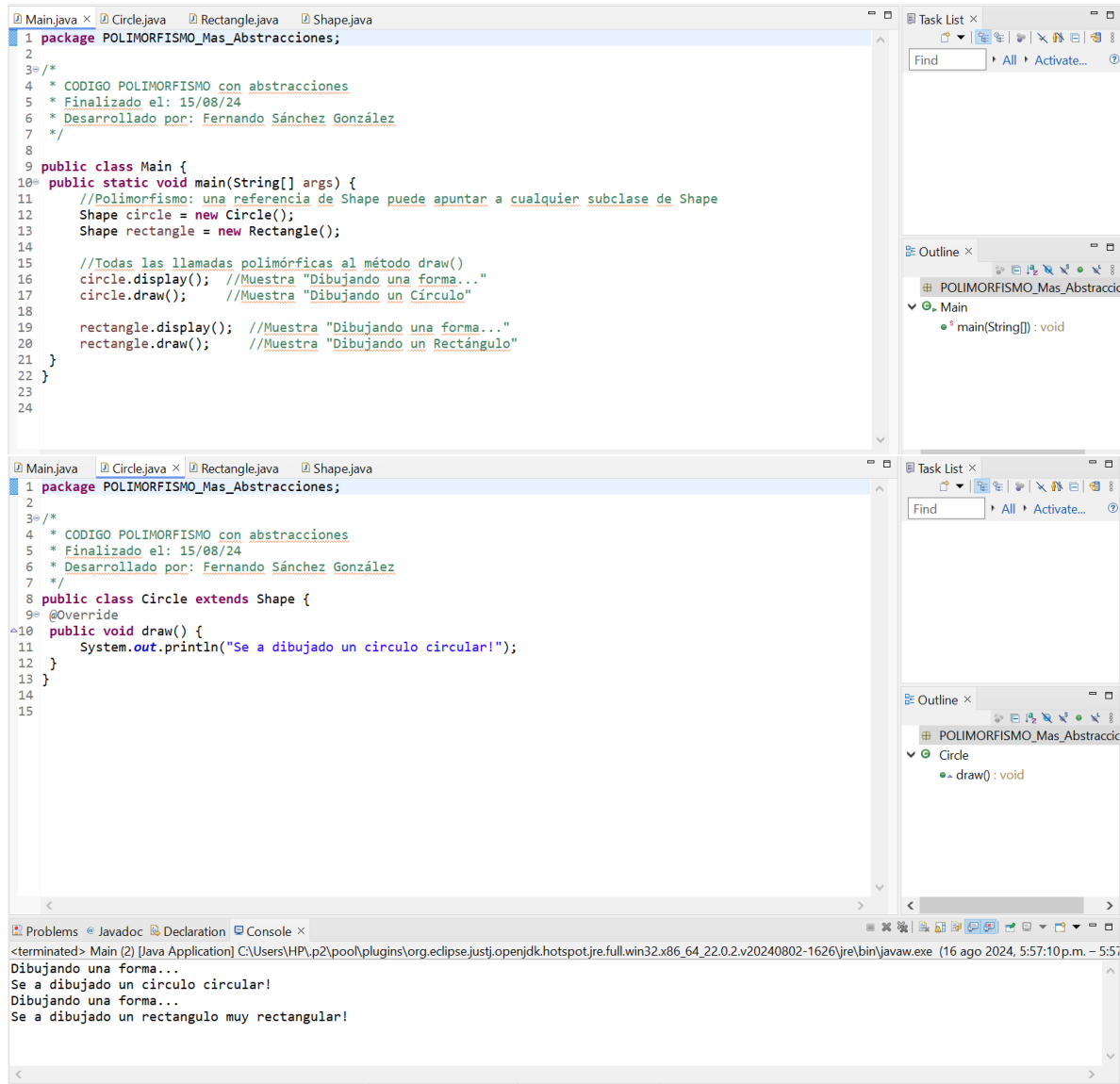


Concepto de Polimorfismo:

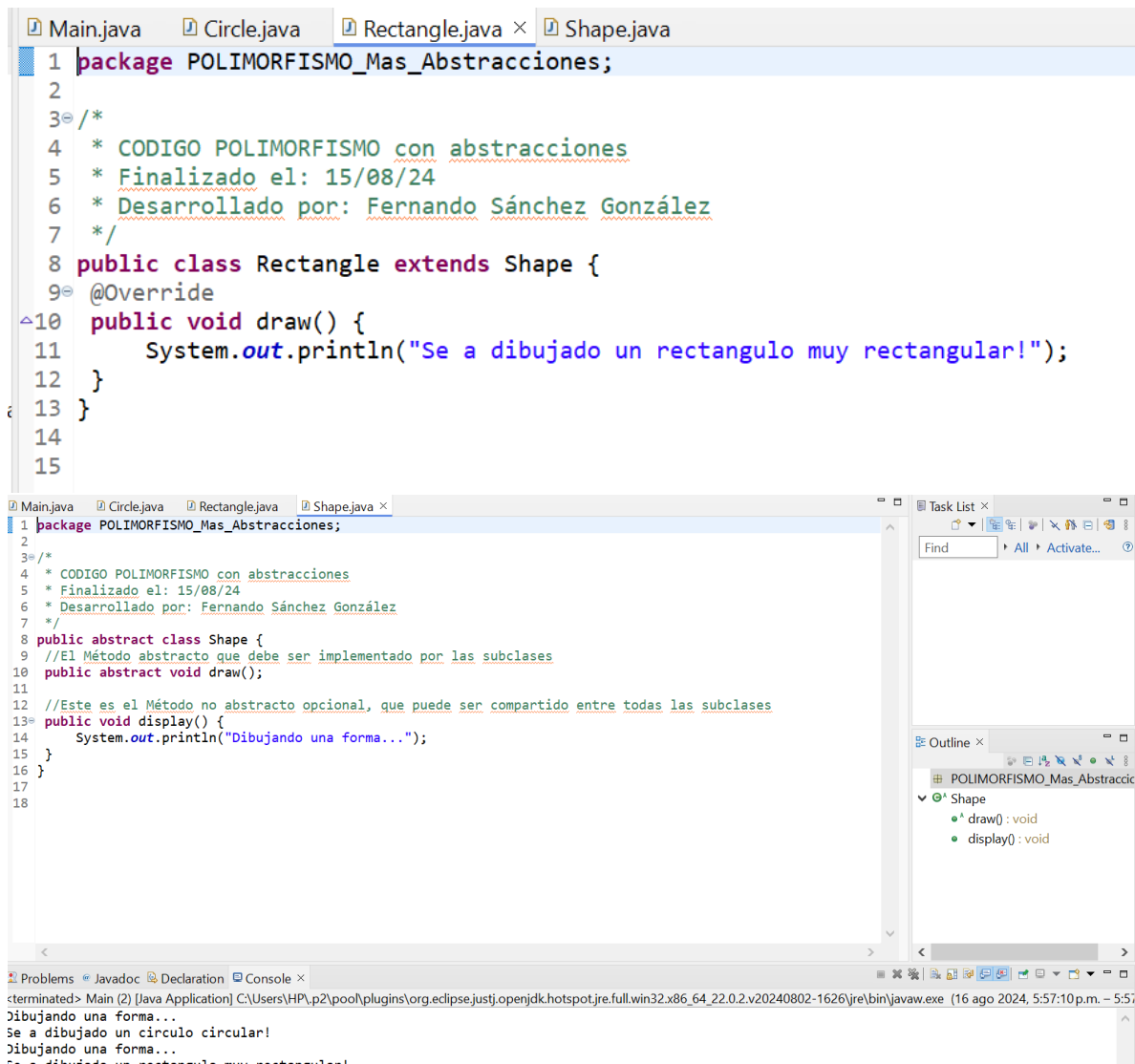


```
1 package POLIMORFISMO_Mas_Abstracciones;
2
3 /**
4  * CODIGO POLIMORFISMO con abstracciones
5  * Finalizado el: 15/08/24
6  * Desarrollado por: Fernando Sánchez González
7  */
8
9 public class Main {
10     public static void main(String[] args) {
11         //Polimorfismo: una referencia de Shape puede apuntar a cualquier subclase de Shape
12         Shape circle = new Circle();
13         Shape rectangle = new Rectangle();
14
15         //Todas las llamadas polimórficas al método draw()
16         circle.display(); //Muestra "Dibujando una forma..."
17         circle.draw();     //Muestra "Dibujando un Círculo"
18
19         rectangle.display(); //Muestra "Dibujando una forma..."
20         rectangle.draw();    //Muestra "Dibujando un Rectángulo"
21     }
22 }
23
24
```

```
1 package POLIMORFISMO_Mas_Abstracciones;
2
3 /**
4  * CODIGO POLIMORFISMO con abstracciones
5  * Finalizado el: 15/08/24
6  * Desarrollado por: Fernando Sánchez González
7  */
8 public class Circle extends Shape {
9     @Override
10     public void draw() {
11         System.out.println("Se a dibujado un circulo circular!");
12     }
13 }
14
15
```

Problems | Javadoc | Declaration | Console

```
<terminated> Main (2) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (16 ago 2024, 5:57:10 p.m. - 5:57:10 p.m.)
Dibujando una forma...
Se a dibujado un circulo circular!
Dibujando una forma...
Se a dibujado un rectangulo muy rectangular!
```



```
1 package POLIMORFISMO_Mas_Abstracciones;
2
3 /*
4  * CODIGO POLIMORFISMO con abstracciones
5  * Finalizado el: 15/08/24
6  * Desarrollado por: Fernando Sánchez González
7  */
8 public class Rectangle extends Shape {
9     @Override
10    public void draw() {
11        System.out.println("Se a dibujado un rectangulo muy rectangular!");
12    }
13 }
14
15
```

```
1 package POLIMORFISMO_Mas_Abstracciones;
2
3 /*
4  * CODIGO POLIMORFISMO con abstracciones
5  * Finalizado el: 15/08/24
6  * Desarrollado por: Fernando Sánchez González
7  */
8 public abstract class Shape {
9     //El Método abstracto que debe ser implementado por las subclases
10    public abstract void draw();
11
12    //Este es el Método no abstracto opcional, que puede ser compartido entre todas las subclases
13    public void display() {
14        System.out.println("Dibujando una forma...");
15    }
16 }
17
18
```

Problems | Javadoc | Declaration | Console x

```
<terminated> Main (2) [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (16 ago 2024, 5:57:10 p.m. - 5:57:10 p.m.)
Dibujando una forma...
Se a dibujado un circulo circular!
Dibujando una forma...
Se a dibujado un rectangulo muy rectangular!
```

Explicación de los .java del Polimorfismo

1. Clase Abstracta (Shape):

- Define el método abstracto `draw()`, que obliga a las subclases a proporcionar su propia implementación.
- También incluye un método no abstracto `display()` que es compartido por todas las subclases.

2. Subclases (Circle, Rectangle):

- Ambas subclases extienden `Shape` y proporcionan implementaciones específicas del método `draw()`.

3. Polimorfismo en Acción (Main):

- En la clase `Main`, se utilizan referencias de tipo `Shape` para apuntar a objetos de tipo `Circle` y `Rectangle`.

- Gracias al polimorfismo, podemos llamar al método `draw()` en las referencias de tipo `Shape`, y la implementación correcta será invocada dependiendo del tipo real del objeto.