

Nombre y apellido alumno: Gomez Hertler Lisandro

DNI: 46461232

Nombre y Apellido alumno: Firmapaz Gabriel Andrés

DNI: 43205785

Profesores: Yanina Medina, Andrea Airaldi y Matías Mascazzini

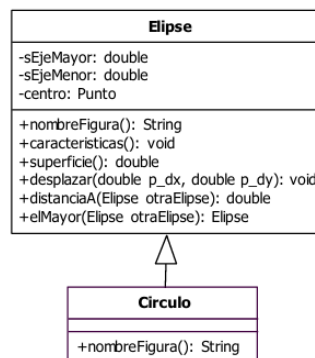
Grupo de Laboratorio: Grupo 2

TP: 5

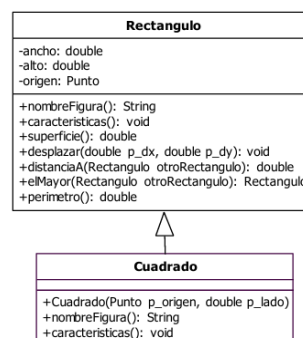
Fecha de entrega: 20/10/2024

Herencia: Se refiere al mecanismo de compartir atributos y métodos utilizando la relación de generalización.

Generalización: La **generalización** es el proceso de abstraer comportamientos y atributos comunes de varias clases en una clase más general o superior. En otras palabras, implica crear una **clase padre (superclase)** que contenga las características comunes a varias clases hijas, Ejemplo (subclases). Por ejemplo:



Especialización: La **especialización** es el proceso inverso a la generalización. En lugar de abstraer comportamientos comunes, se agregan **características o comportamientos específicos** a una subclase que extiende una clase más general. Por ejemplo:



También se hizo uso de la pseudo variable super que es muy útil cuando desde la subclase necesitas acceder o sobrescribir funcionalidades de la superclase.

En caso de Polimorfismo vimos el ejemplo de la clase ArtefactoHogar en donde su método imprimir() imprime una determinada línea de texto pero, en la clase Heladera (que hereda los métodos de ArtefactoHogar) por ejemplo, el método del mismo nombre responde al mismo mensaje sin problemas pero, estando este método modificado.

Imprimir() de ArtefactoHogar:

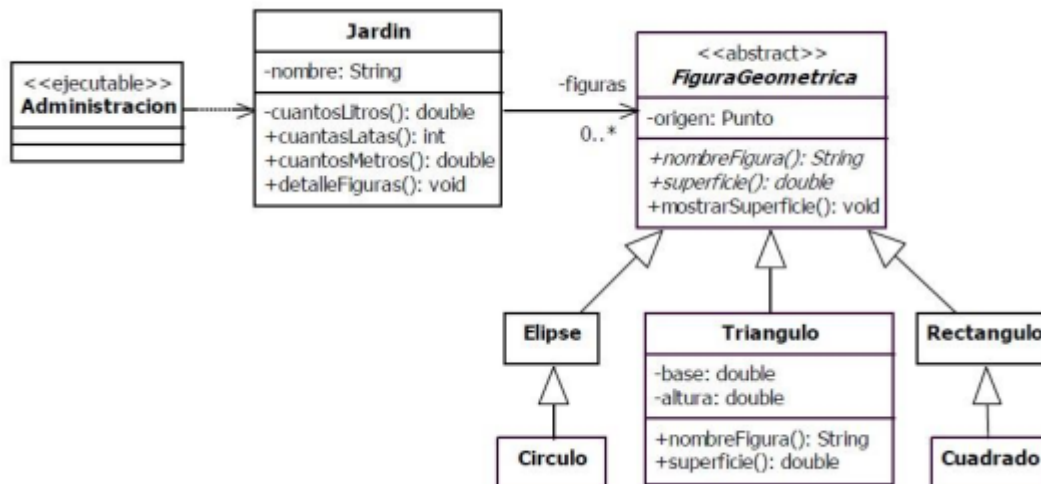
```
64 public void imprimir(){
65     System.out.println("Marca: "+this.getMarca()+" - Precio: $" +this.getPrecio()+" - Stock: "+this.getStock());
66 }
67
```

Imprimir() de Heladera:

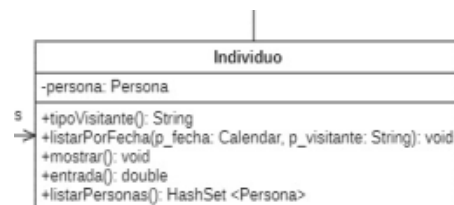
```
55 */
56 public void imprimir(){
57     System.out.println("****Heladera****");
58     super.imprimir();
59     System.out.println("Pies: "+this.getPies());
60     System.out.println("Puertas: "+this.getPuertas());
61     System.out.println("Compresor: "+ this.getCompresor());
62 }
```

Ambos responden al mismo mensaje pero cada uno posee su propio comportamiento.

Polimorfismo y desarrollo **con** reutilización lo usamos en el caso del ejercicio 7 del tp5, en donde ya **desarrollamos y probamos** en los puntos 3 y 4 del tp5 (elipse junto con círculo y rectángulo junto con cuadrado), y estos son utilizados posteriormente en la clase Jardin.



Polimorfismo y desarrollo **para** reutilización lo usamos en el caso del ejercicio 10 del tp5, en donde la clase Persona ya se desarrolló en tps anteriores (mención especial a que esta clase se reutiliza en varias ocasiones en este tp).



Las ventajas que vimos son: que al utilizar objetos abstractos podemos pasarles objetos que posean las mismas características pero que funcionan de distinta forma, ejemplo, ArtefactoHogar con Heladera o Cocina. O también, que cuando codificamos una clase padre, esta posea un protocolo que posteriormente se utilizara en las clases hijas, pero teniendo la posibilidad de transformar internamente los métodos del protocolo sin generar cambios “gigantes” en la implementación, sino simplemente en sumar otras funciones, ejemplo: el método imprimir() de ArtefactoHogar y Cocina.