



PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA 4 POLIMORFISMO

ACTIVIDAD 4.1

GRUPO: 2g2A

NOMBRE DE LOS ALUMNOS:

- ENRIQUEZ MONTALVO RIGOBERTO
- GIL RODRIGUEZ JONATHAN
- GUTIERREZ CRUZ ÁNGEL DE JESÚS
- MORALES VAZQUEZ JUAN DIEGO
- SOSA FIGUEROA BENJAMÍN DE JESÚS

PROFESORA: PATRICIA QUITL GONZALEZ

FECHA DE ENTREGA: 08/05/2023

HORA DE CLASE: 11:00-12:00

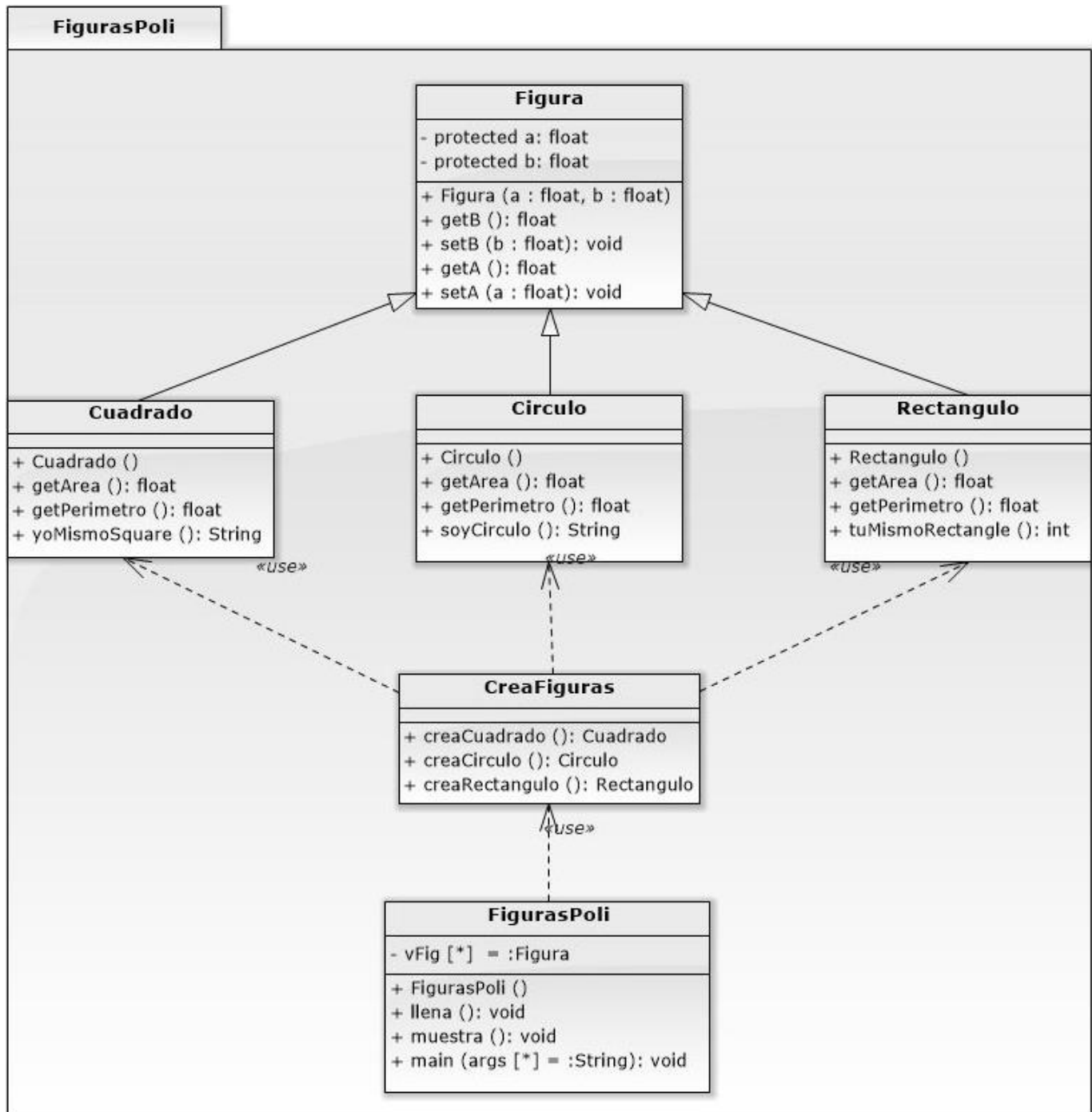


PROBLEMA 1. FIGURAS POLIMORFISMO

Definición del problema

Mostrar perímetro y área de un Cuadrado, Rectángulo y Circulo, aplicando POLIMORFISMO guardando los objetos creados en un mismo vector.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

```
run:
se genero: 2
```

?

cual es la medida base del rectangulo

5

OK

Cancel

```
run:
se genero: 2
```

?

cual es la medida altura del rectangulo

7

OK

Cancel

```
run:
se genero: 2

se genero: 3
```

?

Cual es la medida del radio del circulo

8

OK

Cancel

```
run:
se genero: 2

se genero: 3

se genero: 3
```

?

Cual es la medida del radio del circulo

4

OK

Cancel

```
run:
se genero: 2

se genero: 3

se genero: 3

se genero: 3
```

?

Cual es la medida del radio del circulo

11

OK

Cancel

```
run:
se genero: 2

se genero: 3

se genero: 3

se genero: 3

se genero: 2
```

?

cual es la medida base del rectangulo

20

OK

Cancel

```

run:
se genero: 2

se genero: 3

se genero: 3

se genero: 3

se genero: 2

se genero: 2

```

Input X

? cual es la medida altura del rectangulo

13

OK Cancel

```

run:
se genero: 2

se genero: 3

se genero: 3

se genero: 3

se genero: 2

Tu eres rectangulo
Los datos
35.0 24.0

Los datos
El es un circulo
201.0624 50.264

Los datos
El es un circulo
50.2656 25.132

Los datos
El es un circulo
380.13358 69.113

Tu eres rectangulo
Los datos
260.0 66.0

BUILD SUCCESSFUL (total time: 6 minutes 37 seconds)

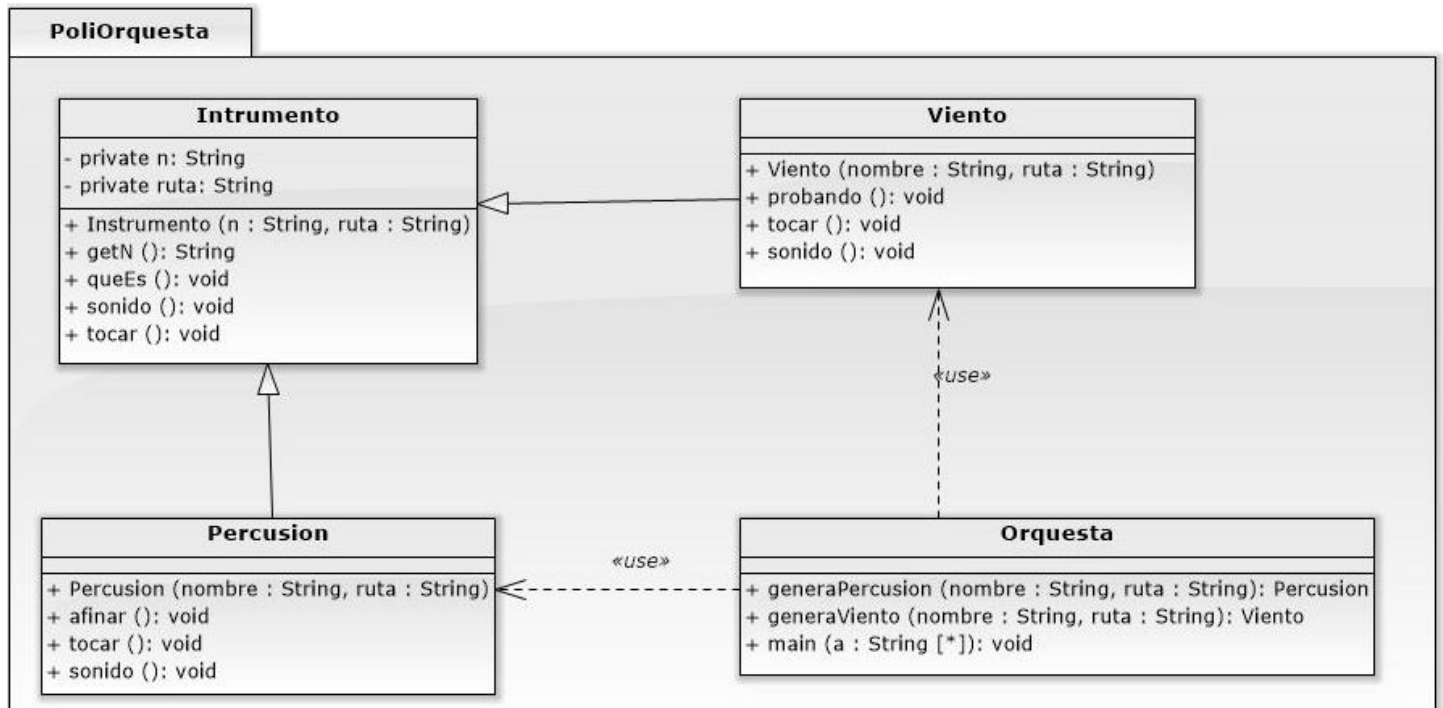
```

PROBLEMA 2. ORQUESTA POLIMORFISMO

Definición del problema

Programa que realice el sonido de un timbal, una trompeta y un platillo, los cuales pueden ser de Viento o de Percusión, estos heredarán de la clase Instrumento y se ejecutaran en la clase Orquesta por medio del POLIMORFISMO, guardando los objetos creados en un mismo vector, además de reproducir su respectivo sonido.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

```

run:
Instrumento      Timbal
                Toca   DO

Afinando.....
Afinado

Instrumento      Trompeta
                Toca   RE

Probando.....  listo

Instrumento      Platillo
                Toca   DO

Afinando.....
Afinado

BUILD SUCCESSFUL (total time: 14 seconds)
  
```

Conclusiones

Enríquez Montalvo Rigoberto: Durante esta unidad he reforzado mis conocimientos referentes a la "herencia" y las clases abstractas. Lo me ha sido de utilidad para comprender el tema del "Polimorfismo" y la diferencia que hay entre un vector común a uno del tipo polimórfico, puesto que este último se destaca en ser un vector capaz de almacenar distintos objetos y no ser de una sola categoría.

Gil Rodriguez Jonathan: En este tema he aprendido como funciona el polimorfismo en java, el cual es uno de los cuatro pilares de la programación orientada a objetos y nos ayuda a que un objeto se pueda comportar de distintas formas, según sea nuestra conveniencia, de ahí la palabra. Aprendí que para aplicar el polimorfismo debe existir la herencia, por lo menos un método abstracto (por consiguiente, una clase abstracta, la cual debe ser la clase padre) y la sobreescritura de este método abstracto. Estas condiciones nos permiten aplicar polimorfismo por medio de los métodos sobrescritos, al crear un vector de objetos del tipo de la clase padre y en cualquier posición del vector instanciar un objeto de cualquier clase hija, lo que nos permite tener un vector polimórfico. Sin duda alguna el polimorfismo nos permite optimizar nuestro código, además de tener un mejor control sobre este.

Gutiérrez Cruz Ángel de Jesús: En este tema aprendí a ocupar los métodos abstractos ya que, aunque los conocía, me generaban confusión respecto a su funcionamiento, igual aprendí a guardar en un vector diferentes tipos de datos pero que comparten entre si una clase padre en común, lo cual considero que será de mucha ayuda en un futuro para poder optimizar cualquier código.

Morales Vázquez Juan Diego: En este tema vimos lo que es el polimorfismo, que es otro pilar del paradigma orientado a objetos donde usamos el tema pasado, la herencia, este tema me resulta algo peculiar pues hace que un objeto puede tomar diferentes comportamientos en tiempo de ejecución por lo que resulta muy conveniente en ciertos casos como lo es el manejo de un vector donde se pueden almacenar dos objetos diferentes pero que están relacionados con una misma clase padre, donde esta es una clase abstracta donde posee al menos un método abstract que define el encabezado pero no implementa el comportamiento pues eso corresponde a la clase hija, como en el ejemplo de manejar un solo CRUD para un vector

polimórfico, lo que facilita la implementación de ciertos códigos donde se es repetitivo, esto cuidando las características de cada clase haciendo distinciones de esta a través del `instanceof` o el `getClass`.

Sosa Figueroa Benjamín de Jesús: Aprendí que, en el polimorfismo en Java, puedo utilizar una misma interfaz o clase abstracta para representar diferentes objetos que comparten ciertas características. Principalmente se nota más en un vector ya que ahora puede tener un comportamiento múltiple ya que se podrá comportar de diferentes formas, esto logrando la optimización de código y una mejor presentación al mismo, además de ver la forma de hacer un pequeño compilador de código para un problema con ecuaciones, apoyándonos de las clases que nos proporciona Java `Trim` y `StringTokenizer` que nos ayudan mucho manejando cadenas, aun así lo que se me complicó más es el aprender manejar las clases abstractas ya que se manejan de una forma bastante diferente a la tradicional, desde la sintaxis, así como la invocación de los métodos.