



PROGRAMACIÓN ORIENTADA A OBJETOS

TEMA 3 HERENCIA

ACTIVIDAD 3.1

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: 14/04/2023

HORA DE CLASE: 11:00-12:00

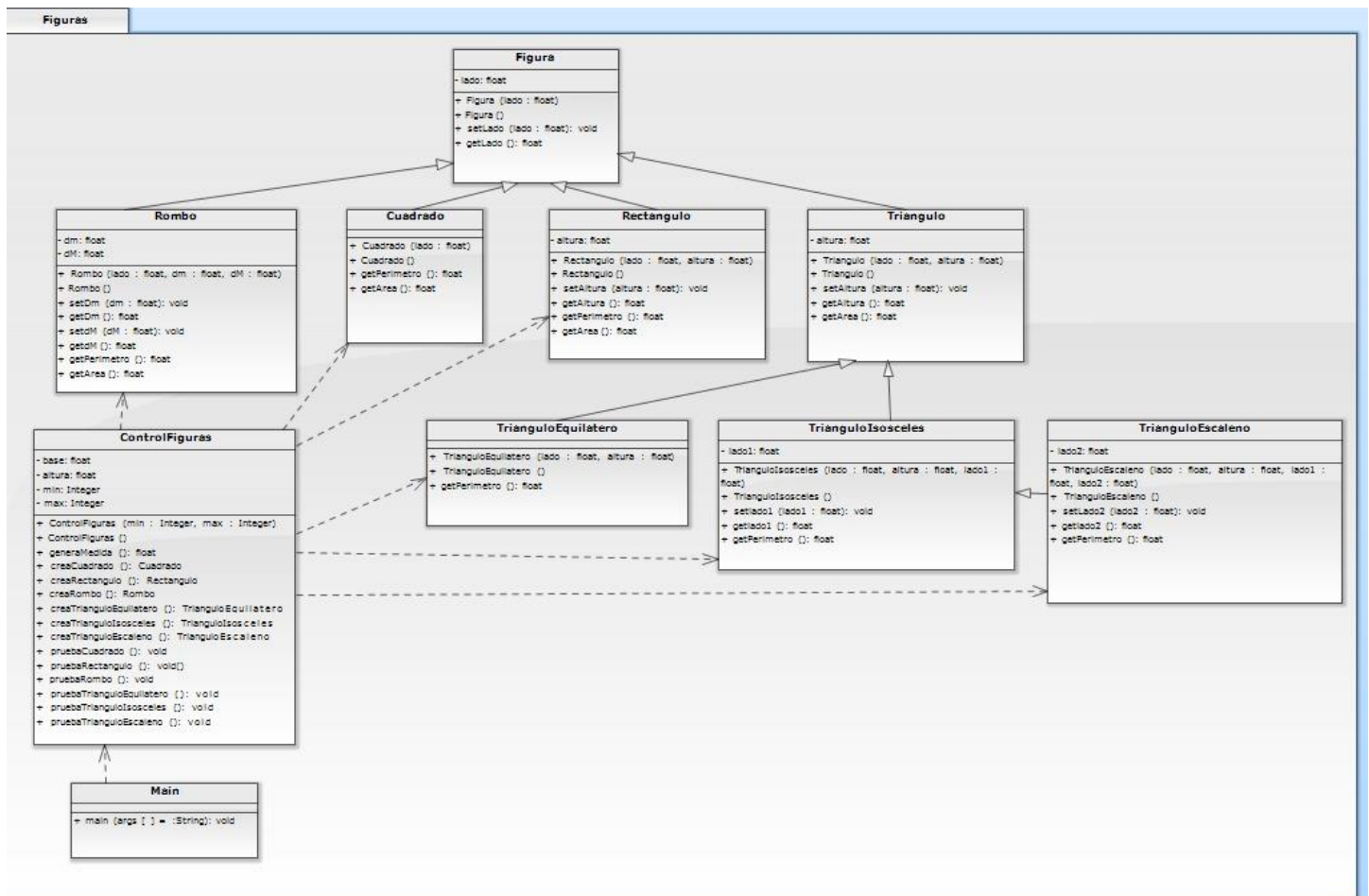


PROBLEMA 1. FIGURAS HERENCIA VERSIÓN 1

Definición del problema

Mostrar perímetro y área de un Cuadrado, Rectángulo, Triangulo Equilátero, Triangulo Isósceles, Triangulo escaleno y Rombo, aplicando HERENCIA.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

```

Output - figuraherencia_1_2923 (run)

run:
Pruebas
Cuadrado:
LADO: 39.934265
Perimetro cuadrado: 159.73706
Area cuadrado: 1594.7455
  
```

```
Rectangulo:
BASE: 49.079403
ALTURA: 17.844357
Perimetro rectangulo: 133.84752
Area rectangulo: 875.79034

Rombo:
BASE: 21.658844
Diagonal menor: 18.029995
Diagonal mayor: 20.36546
Perimetro rombo: 86.635376
Area rombo: 183.59456

Triangulo Equilatero:
BASE: 32.158924
ALTURA: 16.830807
Perimetro Triangulo Equilatero: 96.476776
Area Triangulo Equilatero: 270.6303

Triangulo Isosceles:
BASE: 43.566833
LADO: 19.34377
ALTURA: 35.893032
Perimetro Triangulo Escaleno: 82.25437
Area Triangulo Escaleno: 781.87286

Triangulo Escaleno:
BASE: 32.12801
LADO 1: 20.205267
LADO 2: 19.821707
ALTURA: 37.18245
Perimetro Triangulo Escaleno: 72.15498
Area Triangulo Escaleno: 597.2991

BUILD SUCCESSFUL (total time: 5 seconds)
```



Output



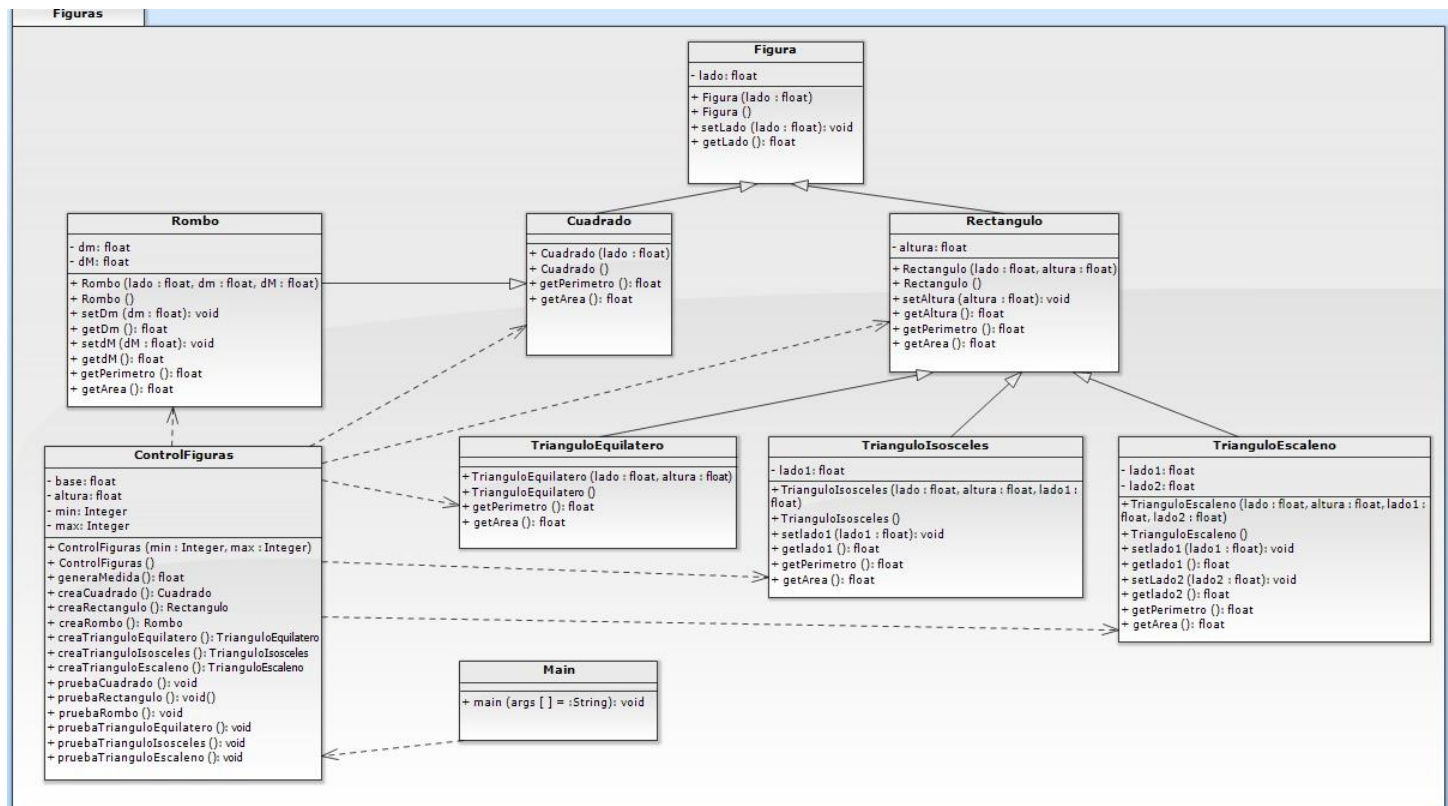
Notifications

PROBLEMA 2. FIGURAS HERENCIA VERSIÓN 2

Definición del problema

Mostrar perímetro y área de un Cuadrado, Rectángulo, Triangulo Equilátero, Triangulo Isósceles, Triangulo escaleno y Rombo, aplicando HERENCIA. Ahora Rombo heredará de Cuadrado y Triangulo Equilátero, Triangulo Isósceles y Triangulo escaleno heredarán de Rectángulo.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

```

Output - figuraherencia_2_2923 (run)

run:
Pruebas
Cuadrado:
LADO: 37.33582
Perimetro cuadrado: 149.34328
Area cuadrado: 1393.9634

Rectangulo:
BASE: 21.805576
  
```

```
ALTURA: 26.744709
Perimetro rectangulo: 97.10057
Area rectangulo: 583.1838

Rombo:
BASE: 38.797443
Diagonal menor: 21.500343
Diagonal mayor: 33.442986
Perimetro rombo: 155.18977
Area rombo: 359.51782

Triangulo Equilatero:
BASE: 38.188793
ALTURA: 16.730118
Perimetro Triangulo Equilatero: 114.566376
Area Triangulo Equilatero: 319.4515

Triangulo Isosceles:
BASE: 26.423601
LADO: 34.511757
ALTURA: 44.484413
Perimetro Triangulo Escaleno: 95.44711
Area Triangulo Escaleno: 587.7192

Triangulo Escaleno:
BASE: 37.050797
LADO 1: 34.49493
LADO 2: 41.229744
ALTURA: 24.802488
Perimetro Triangulo Escaleno: 112.775475
Area Triangulo Escaleno: 459.47598

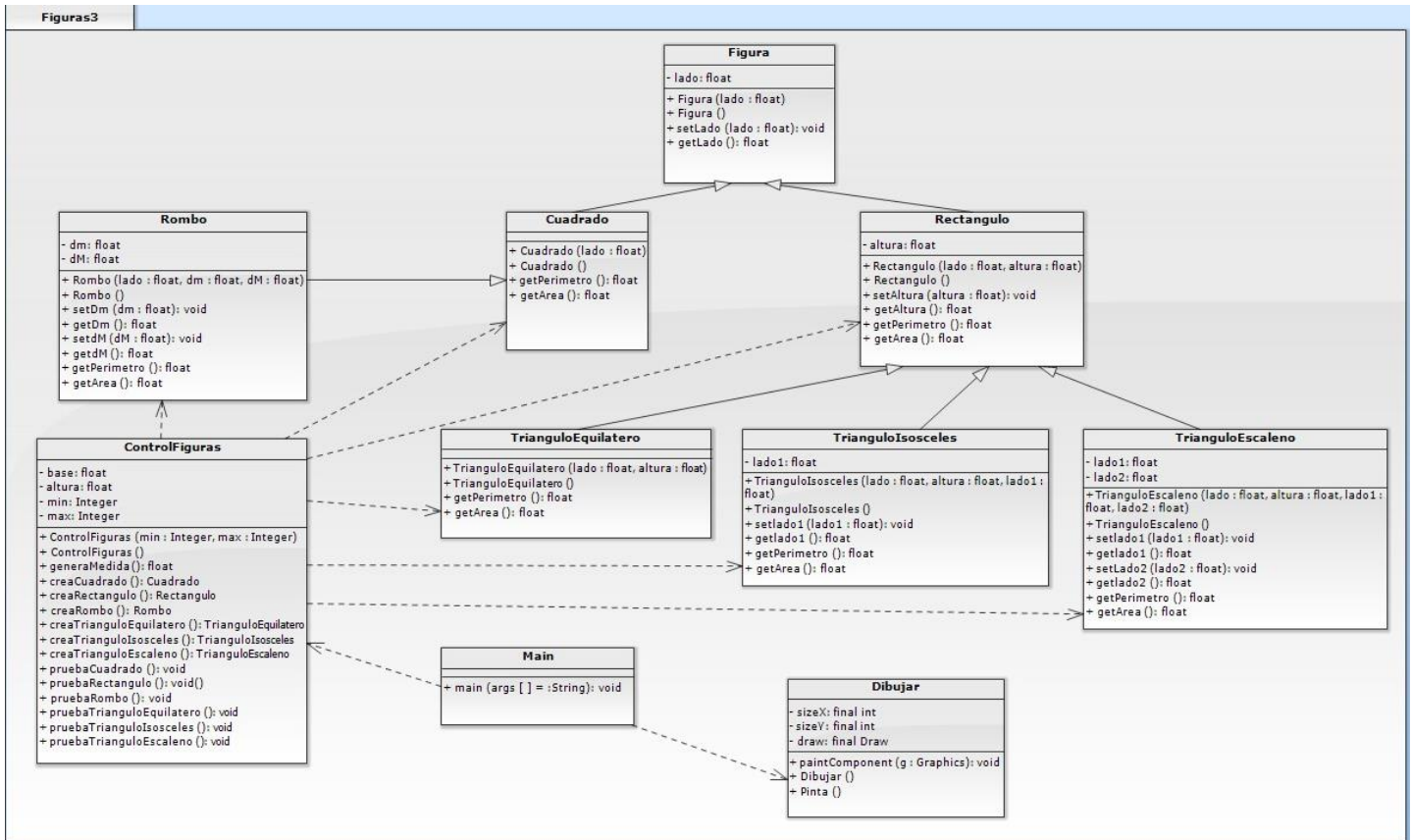
BUILD SUCCESSFUL (total time: 1 second)
```

PROBLEMA 3. FIGURAS HERENCIA VERSIÓN 3

Definición del problema

Mostrar perímetro y área de un Cuadrado, Rectángulo, Triangulo Equilátero, Triangulo Isósceles, Triangulo escaleno y Rombo, aplicando HERENCIA. Ahora Rombo heredará de Cuadrado y Triangulo Equilátero, Triangulo Isósceles y Triangulo escaleno heredarán de Rectángulo. Además, agregar una clase la cual pinte las figuras.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

Output - figuraherencia_3_2923 (run)

```

run:
Pruebas
Cuadrado:
LADO: 32.031815
Perimetro cuadrado: 128.12726
Area cuadrado: 1026.0371

Rectangulo:
BASE: 29.701057
ALTURA: 44.700363
Perimetro rectangulo: 148.80284
Area rectangulo: 1327.6481

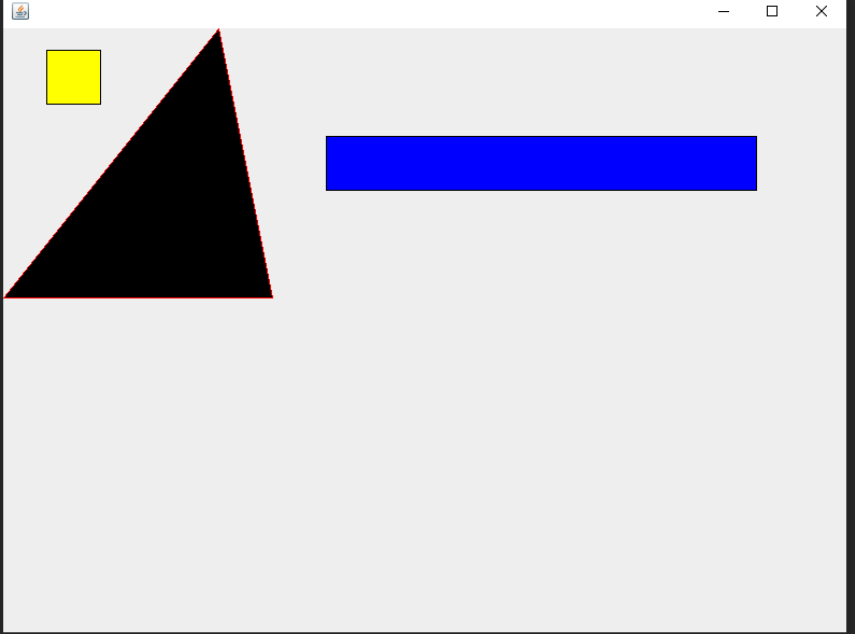
Rombo:
BASE: 15.970419
Diagonal menor: 34.01695
Diagonal mayor: 24.682241
Perimetro rombo: 63.881676
Area rombo: 419.80728

Triangulo Equilatero:
BASE: 46.38097
ALTURA: 38.236835
Perimetro Triangulo Equilatero: 139.14291
Area Triangulo Equilatero: 886.7308

Triangulo Isosceles:
BASE: 41.803417
LADO: 40.882133
ALTURA: 47.43051
Perimetro Triangulo Escaleno: 123.56769
Area Triangulo Escaleno: 991.3787

Triangulo Escaleno:
BASE: 20.730858
LADO 1: 15.818126
LADO 2: 42.23901
ALTURA: 24.905226
Perimetro Triangulo Escaleno: 78.787994
Area Triangulo Escaleno: 258.15335

```

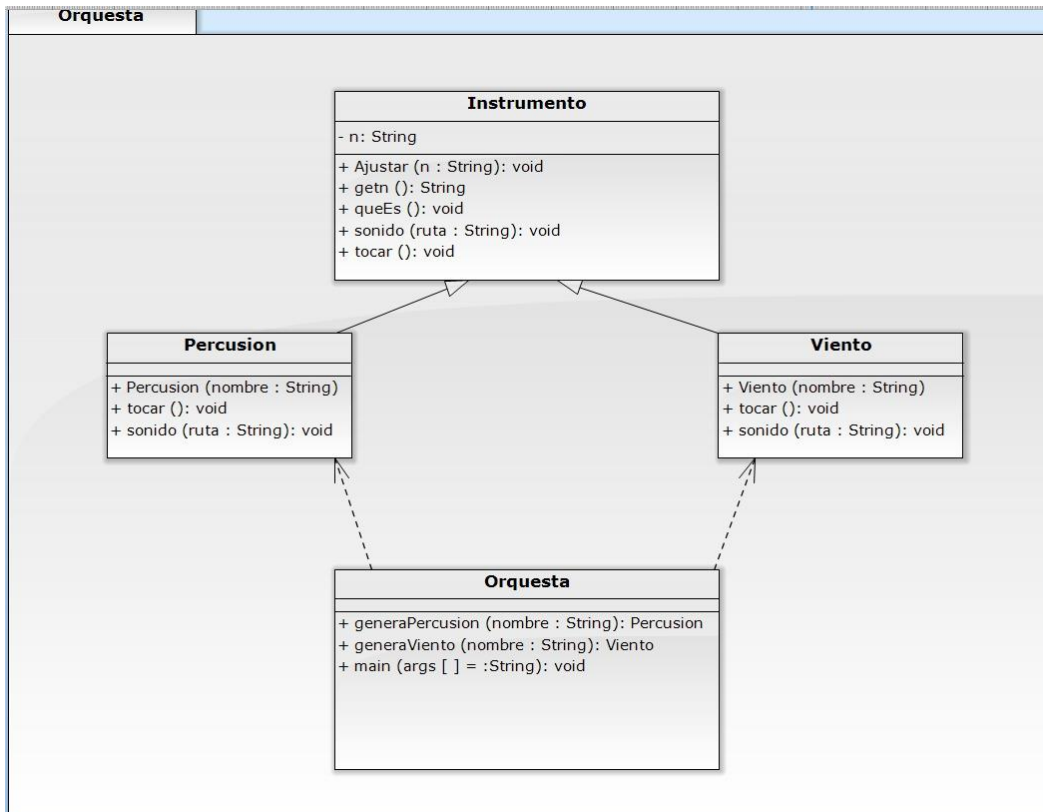


PROBLEMA 4. HERENCIA ORQUESTA

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, además de reproducir su respectivo sonido.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba

```

Output - HerenciaOrquesta_1 (run)

run:
Instrumento    TIMBAL
             Toca    DO

Instrumento    TROMPETA
             Toca    RE

Instrumento    PLATILLO
             Toca    DO

BUILD SUCCESSFUL (total time: 19 seconds)
  
```

PROBLEMA 5. HERENCIA CENTRO CULTURAL

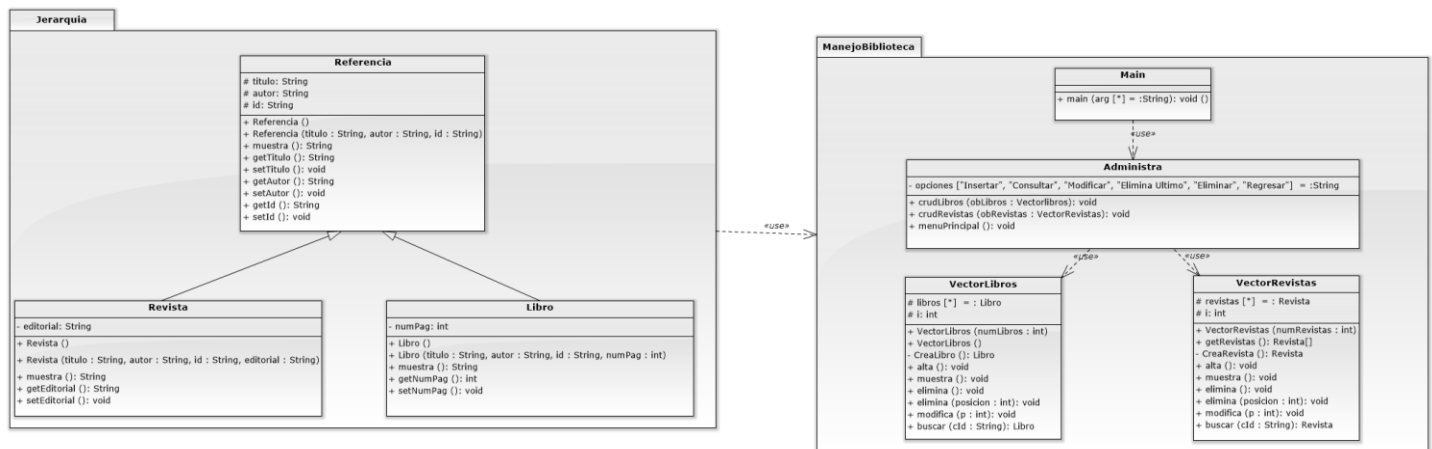
Definición del problema

Un centro cultural se dedica al préstamo de dos tipos de materiales: Revista y Libros. Para los dos se guarda información general, como su código identificativo, el título y el autor. En el caso de los libros, almacenamos también su número de páginas, y para los discos el nombre de la discografía.

Al centro cultural acuden una serie de clientes (de los que se pregunta su NC y Nombre), que realizan una serie de peticiones de revistas o libros (como mucho hasta 5 peticiones). Para cada petición se guarda la fecha de inicio y su fin de préstamo. Diseñar y construir las clases apropiadas para el supuesto planteado anteriormente.

Crear las clases Libro, Revista y Referencia en un paquete llamado jerarquía, estas primeras dos heredarán de la clase Referencia. Las demás clases se crearán en el paquete manejoBiblioteca.

Diseño de la solución mediante diagrama de clases (UML)



Casos de prueba



OPCIONES

Menu Libros

Insertar

Consultar

Modificar

Elimina Ultimo

Eliminar

Regresar

Input



CodID?

3020

OK

Cancel

Input



Titulo?

El Arte De Amarte

OK

Cancel

Input



autor?

Nappa

OK

Cancel

Input



Num Pag?

40

OK

Cancel

Message



Nombre: El Arte De Amarte

Autor: Nappa

ID: 3020

Numero de paginas: 40

OK

Input



Que posicion desea modificar?

0

OK

Cancel

Input



ESTO ES LO QUE DESEAS MODIFICAR si o no ?

Nombre: El Arte de Amarte

Autor: Nappa

ID: 3020

Numero de paginas: 40

si

OK

Cancel

Input



Nuevo ID?

2023

OK

Cancel

Input

?

Nuevo titulo?

POO

OK

Cancel

Input

?

Nuevo autor?

Jonathan

OK

Cancel

Input

?

Num pags?

23

OK

Cancel

Message

i

Nombre: POO
Autor: Jonathan
ID: 2023
Numero de paginas: 23

OK

OPCIONES

Mnu Revistas

Insertar

Consultar

Modificar

Elimina Ultimo

Eliminar

Regresar

Input

?

CodID?

3020

OK

Cancel

Input

?

titulo?

Avengers Java Team

OK

Cancel

Input

?

autor?

Stan Lee

OK

Cancel

Input

?

Editorial?

Trillas

OK

Cancel

Message

i

Nombre: Avengers Java Team
Autor: Stan Lee
ID: 3020
Editorial: Trillas

OK

Conclusiones

Enríquez Montalvo Rigoberto: Durante esta unidad comprendí el significado y uso de la herencia, la cual hace referencia a qué una clase base (padre) herede sus atributos o métodos comunes a una subclase (hija), de modo que ésta no tenga que volver a reescribirlos y así el código quedé más simplificado.

Por su parte, en caso de agregar un método que contenga el mismo nombramiento de un método de la clase padre se tendrá que agregar el término "override" el cual permitirá agregar este método sin problema alguno.

Gil Rodriguez Jonathan: En este tema he aprendido el cómo funciona la herencia en la programación orientada a objetos, la cual nos permite modelar jerarquías de clases, haciendo que clases hijas (subclases) incorporen el comportamiento de clases con elementos más básicos (superclases). Además, también aprendí que los atributos de las superclases deben usar el modificador de acceso "protected", con la finalidad de que las demás clases no puedan acceder a estos directamente, sino que lo hagan por medio de sus respectivos métodos Setters y Getters. Al igual que las clases que ya no tendrán más clases hijas deben tener la palabra reservada "final", con el fin de que no se pueda heredar de estas clases.

Al hacer los distintos programas con herencia, me di cuenta que la herencia nos permite de alguna forma reutilizar código, al solo crear atributos y métodos en la clase padre los cuales tienen en común las demás clases hijas. Como el ejemplo del área del triángulo, el cual es un método en común en los 3 tipos de triángulos. Sin duda la herencia nos permite modelar con mayor exactitud cómo funcionan los objetos de la vida real en la programación, aun así, es importante usarla correctamente, con el fin de optimizar nuestro código.

Gutiérrez Cruz Ángel de Jesús: En esta unidad, se aprendió una nueva forma de programación orientada a compartir atributos y métodos a clases diferentes que compartan características similares, esto con el fin de escribir menos y optimizar las líneas de código. Igual se observó el comportamiento del @Override el cual permite sobreponer un método, cuando hay métodos similares entre la clase padre a la hija, pero que se comportan diferente esto con el fin de darle el comportamiento deseado a la clase hija.

Morales Vázquez Juan Diego: En este nuevo tema, vimos otro pilar fundamental de la programación orientada a objetos, un tema que se complementa con lo antes visto, es una nueva forma de crear relaciones entre clases con la que tenemos grandes ventajas como la centralización de código y el uso de una jerarquía de clases. Vimos muchos conceptos nuevos como lo una clase derivada, una superclase, la generalización y la especialización; que son formas de abstraer la realidad de los objetos con el fin de crear jerarquías, los tipos de herencia; simple y múltiple, un nuevo modificador; el `protected`; que hace solo visible los atributos a subclases y clases del paquete, la sobrescritura; con el `@override` como una forma de redefinir métodos de la clase padre, entre muchas cosas que se usaron al experimentar resolviendo los problemas en clase como el usar métodos de clases propias de java para poder reproducir un sonido o para dibujar diferentes figuras.

Esta nueva forma de programar donde hacemos uso de una jerarquía y factorización de clases, es muy útil y me parece muy interesante como se puede llegar a usar.

Sosa Figueroa Benjamín de Jesús: En este tema se vio una nueva forma de expandir código y separarlo más cómodamente, dándonos una herramienta fundamental para la programación con el paradigma Orientado a Objetos, la herencia nos permite compartir métodos entre clases y como si fuera poco otros paquetes, como adición no tenía ni idea sobre cómo colocar una imagen, un sonido en java y ahora ya tengo alternativas para hacerlo, con el `JFrame` o incluso con el `JOptionPane`, permitiendo dar una mejor presentación a los programas que se realicen, se vio un nuevo tipo de variable que sería la protegida (`protected`), la cual le otorga más sentido a la función de los `set` y los `get` gracias a la comunicación entre Paquetes.