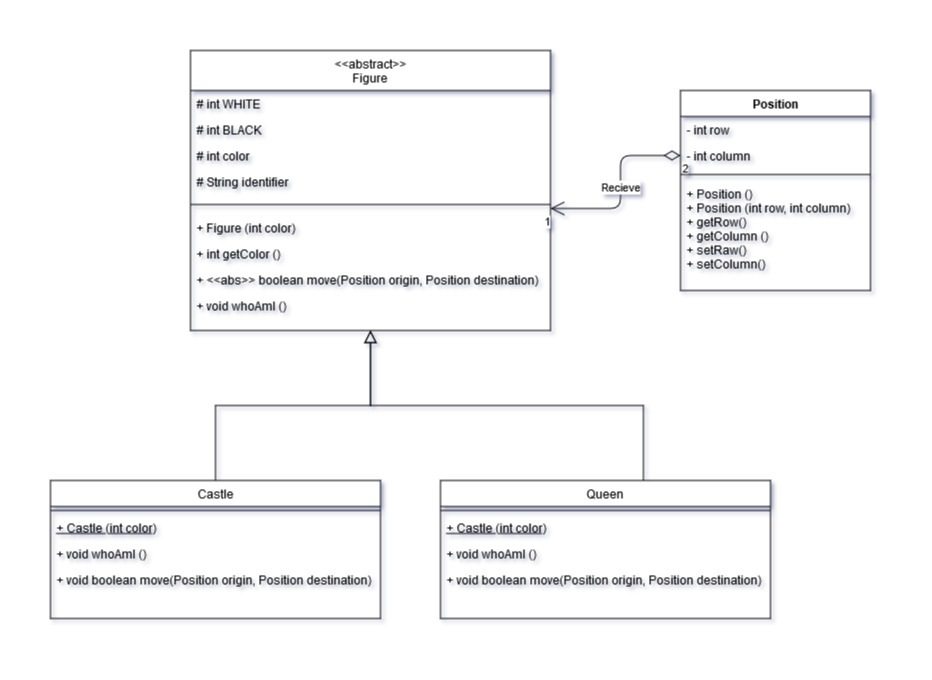
**Third Practice**

**Name:** Mirko Fernando Romay Ramos

1. **Inheritance and references to the base class**

**Class Diagram.**

**Further Questions.**

* Which methods can be really invoked on the collection elements?

Only not-abstract classes (Queen, Position and Castle), because of Figure can’t be instanced.

* If Castle class has implemented void castle() method, could it be possible to invoke that method from a reference to the base class? Why?

It can’t, because a son class obtains parent’s methods by inheritance, but father don’t receive any method of his sons.

* What should we have to do in order to be able to use the previous void castle() method from an object of Castle class that is pointed by a reference to Figure class?

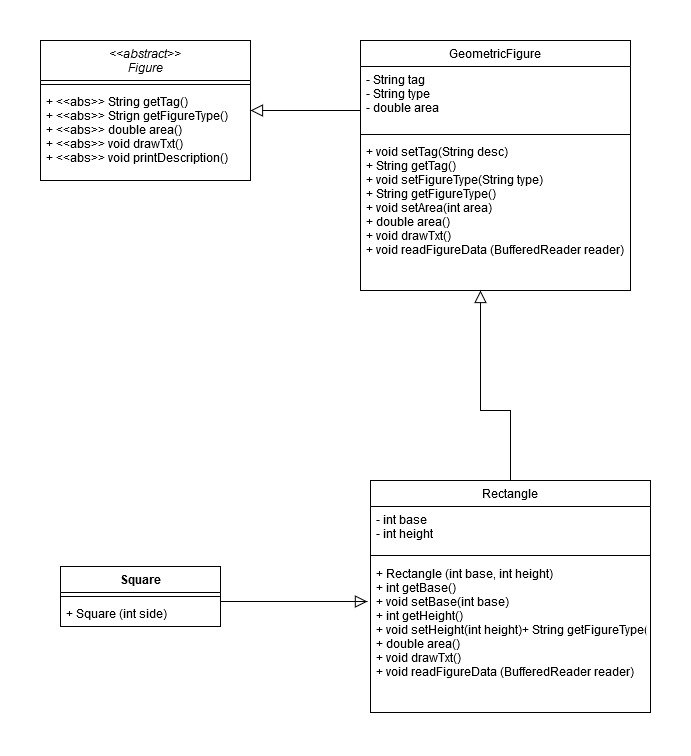
If an object is an instance of Castle Class, it can use castle() method only calling its constructor.

* What should we do to know exactly to which class belongs every object pointed by a reference to the base class?

We have to use “instanceof” reserved word in order to know if an object is an instance of a determined class.

1. **Exercise to practice with Figures.**

**Class Diagram.**

****

**Rectangle – Answer the following questions.**

* + Show the difference between a class and an object.

A “class” is a model or template that define some attributes and methods in common to some objects; and an “object” is an instance of a class, and we can say that an object belongs to a determined class.

* + Which steps are involved in the instantiation's process of an object?

First, we have to call the class that the object will belong.

Then, we have to declare a variable that contains the object.

After that, we will create a new object using the “new” reserved word.

Finally, we have to assign the object to the variable to be used.

* + How is an object instantiated in Java?

Follow the next notation:

**Rectangle rect = new Rectangle (“R-1”, 4, 2);**

That was an example to case of *Rectangle* class, it has a constructor defined but some classes use default void constructor.

**Square – Answer the following questions.**

* + What is inheritance?

It’s the capability of a class extended from another of receive all data and methods of its parent.

* + How do you express in Java that one class *inherits* from another?

When a class is defined, it has to be declared with the “extends” reserved word, to define which is the parent class of it.

* + Which methods of the superclass are visible from the subclasses?

All methods that are no created as “private”.

* + What is the meaning of method overriding?

It occurs when one or more new methods are created with the same name of another method created previously, so they must support different types and/or quantity of arguments.

* + Remember that, opposite to the rest of the methods of a class, subclasses don't automatically inherit constructors from the superclass, but they can be invoked by the use of super() keyword.

For example, in Square class that extends from Rectangle class, we have the follow constructor:

**public Square(int side) {**

**super(side, side);**

**}**

Where *super(int, int)* refers to Rectangle Class’ constructor, so its base and height are defined by Square’s side vaue.

**Reference to Interfaces – Answer the following questions.**

* Which type are both instantiated objects (in options 1 and 2)?

They are a Rectangle and a Square respectively, but both extends form GeometricFigure so they have its methods too.

* Which type is the variable that references them?

The variable is a GeometricFigure firstly, but then it changes when is instanced as a Rectangle or a Square.

* Which methods from superclass are visible from the subclass?

There are methods visible because of they are public methods.

* Can you use the same variable as a reference for different types of figures? Why?

Yes, I can because at the first this variable only has been declared, but it can be instanced later.