**Name:** Laura Helen Montaño Toro

**Trainer:** Cristelh Miranda Suarez

**Matter:** PROG101

**Date:** 08/05/2020

**2.2 Exercise to practice with Figures**

**UML**

|  |
| --- |
| Abstract Figure |
| # WHITE: int = 0  #BLACK: int = 1  #color: int  #identifier: String |
| +getArea(): int  +whoAmI(): void  +abstract move(Position origin, Position destination): boolean |

|  |
| --- |
| Position |
| -row: int  -column: int |
| +getRow(): int  +getColumn(): int  +setRow(int row): void |

|  |
| --- |
| GeometricFigure |
| +tag: String = ””  +label: String = ””  -area1: double |
| +getTag(): String  +setLabel(GeometricFigure figure): void  +setTag(String tag): void  +getLabel(): void  +getArea(GeometricFigure figure): double  +printDescription(GeometricFigure gFigure): void  +abstract move(Position origin, Position destination): boolean |

|  |
| --- |
| Rectangle |
| +height: int  +width:int  +LABEL: “Rectangle” |
| +getHeight(): int  +getWidth(): int  +setHeight(int height): void  +setWidth(int width): void  +getFigureType(): String  +getLabel(): String  +area(): double  +drawTxt(): void |

|  |
| --- |
| Square |
| +size: int  +LABEL: “Square” |
| +getFigureType(): String  +getLabel(): String  +area(): double |

**The rectangle**

* Show the difference between a class and an object.

R: una clase es una generalización, abstracción de lo que queremos representar del mundo real. El objeto es una instancia de la clase, es la representación

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

R: Se crea un nuevo objeto de la clase y llama al constructor de la clase y se envían los argumentos que este requiere.

* How is an object instantiated in Java?

R: Haciendo una llamada al método Constructor de la clase.

**The square**

* What is inheritance?

R: La herencia es una forma de reutilización de software en la que se crea una nueva clase absorbiendo los miembros de una clase existente, y se mejoran con nuevas capacidades, o con modificaciones en las capacidades existentes.

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

R: Con la palabra reservada: extends.

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

R: Los que tienen modificadores de accceso públicos y protected.

* What is the meaning of method overriding?

R: Es sobreeescribir el método heredado de una superclase.

* 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.

**Reference to interfaces**

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

R: Como la clase Rectangle implementa a la Interfaz GeometricFigureTest, puede decirse que los objetos instanciados en la opción 1 y 2 son objetos GeometricFigureTest.

* Which type is the variable that references them?

R: Una variable de tipo interfaz.

* Which methods from superclass are visible from the subclass?

R: Son visibles todos los métodos con modificadores de acceso públicos y protected.

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

R: Si, porque generalmente una interfaz se usa cuando clases no relacionadas necesitan compartir métodos y constantes comunes. Esto permite que los objetos de clases no relacionadas se procesen en forma polimórfica.