Taller 04

1. Con ayuda de los IDEs[1] presentados en clase y teniendo en cuenta la definición de las clases `ComplexNumber` y `ComplexCalculator` (las cuales no tienen errores) **encuentre, mencione y explique** cuáles son los errores de compilación (aquellos que no permitan que el programa sea ejecutado correctamente) de los siguientes fragmentos de código Java. Algunos fragmentos de código pueden tener más de un error, o no tener ☺.

   Nota: los códigos pueden tener uno o más errores.

```java
package oop.assignments.a02.p1;

public class ComplexNumber {

    private double realPart;
    private double imaginaryPart;

    public double getImaginaryPart() {
        return imaginaryPart;
    }

    public void setImaginaryPart(double imaginaryPart) {
        this.imaginaryPart = imaginaryPart;
    }

    public double getRealPart() {
        return realPart;
    }

    public void setRealPart(double realPart) {
        this.realPart = realPart;
    }
}
```

Figura 1 – Definición de la clase ComplexNumber

[1] IDE Integrated development environment (Entorno de desarrollo integrado)

```java
package oop.assignments.a02.p1;

public class ComplexCalculator {

    public static ComplexNumber sum(ComplexNumber numA, ComplexNumber numB) {

        double sumRealPart = numA.getRealPart() + numB.getRealPart();
        double sumImaginaryPart = numA.getImaginaryPart() + numB.getImaginaryPart();

        ComplexNumber result = new ComplexNumber();
        result.setRealPart(sumRealPart);
        result.setImaginaryPart(sumImaginaryPart);

        return result;
    }

    public static ComplexNumber substraction(ComplexNumber numA, ComplexNumber numB) {

        double subRealPart = numA.getRealPart() - numB.getRealPart();
        double subImaginaryPart = numA.getImaginaryPart() - numB.getImaginaryPart();

        ComplexNumber result = new ComplexNumber();
        result.setRealPart(subRealPart);
        result.setImaginaryPart(subImaginaryPart);

        return result;
    }

    public static ComplexNumber opposite(ComplexNumber number) {

        ComplexNumber opposite = new ComplexNumber();
        opposite.setRealPart(-number.getRealPart());
        opposite.setImaginaryPart(-number.getImaginaryPart());

        return opposite;
    }
}
```

Figura 2 - Definición de la clase ComplexCalculator

1.1

```java
package oop.assignments.a02.fail;

public class A02P11 {

    public static void main(String[] args) {

        ComplexNumber number = new ComplexNumber();

        number.realPart = 3.3;
        System.out.println("Real Part is: " + number.getRealPart());
    }
}
```

1.2

```java
package oop.assignments.a02.p1;

public class A02P12 {

    public static void main(String[] args) {

        ComplexNumber number = new ComplexNumber();

        number.setImaginaryPart(456, 897);
        number.setRealPart(521);

        String realPart = number.getRealPart();
        System.out.println(realPart + number.getImaginaryPart(123));
    }
}
```

1.3

```java
package oop.assignments.a02.p1;

public class A02P13 {

    public static void main(String[] args) {

        ComplexNumber number = new ComplexNumber();

        number.setRealPart('c');

        ComplexCalculator.opposite(number, number);
    }
}
```

1.4

```java
package oop.assignments.a02.p1;

import java.util.ArrayList;

public class A02P14 {

    private static ArrayList<ComplexNumber> numbers;

    public static void main(String[] args) {

        ComplexCalculator operator = new ComplexCalculator();

        numbers = new ArrayList<ComplexNumber>();
        numbers.add(operator);
        numbers.size(5);
    }
}
```

1.5

```
1  package oop.assignments.a02.p1;
2
3  public class A02P15 {
4
5      ComplexNumber number;
6
7      number.setRealPart(4587);
8
9      public static void main(String[] args) {
10
11         ComplexNumber  number = new ComplexNumber()
12
13         System.out.printf("%f + %f",number.getImaginaryPart(),number.getRealPart());
14     }
15 }
```

2. De los siguientes enunciados identifique las clases involucradas y realice el respectivo diagrama UML con atributos, métodos, nombres, relaciones con multiplicidad y rótulos (ver [Barker] capítulo 10). No se deben colocar los métodos get y set en los diagramas

2.1 **Portal de ventas por catálogo seletiene.com**

Su equipo de desarrollo ha sido contratado para implementar un portal de ventas denominado seletiene.com. A través de la interfaz grafica de usuario este portal proporciona a sus clientes un catalogo de productos, un carrito de compras y multiples esquemas de envío. El portal permite vender productos y bolsas de productos. Cada producto o bolsa de producto viene en tres presentaciones que son gama alta, gama media, gama baja. Las bolsas de productos son ofertas de productos especiales de la misma gama que se empaquetan por demanda. Cada bolsa debe tener mínimo 3 productos de los ofrecidos para la bolsa. Al hacer la compra el usuario puede seleccionar tres formas de pago que van de acuerdo al esquema de entrega.

2.2 **Sistema de gestión de citas médicas**

Recientemente la administración de salud regional ASR de su ciudad ha decidido abrir una convocatoria para la realización del nuevo sistema para la gestión de citas médicas para el seguro social. Este sistema cuenta con las siguientes funcionalidades

- Los médicos afiliados obligatoriamente a una sucursal hospitalaria puedan consultar las citas que tienen en su agenda.

- Los usuarios del servicio podrán verificar de manera online la disponibilidad de los médicos así como la información de los mismos (especialidad, horario, nombre, etc)
- Los usuarios podrán consultar el estado de sus citas (fecha, hora consultorio, etc) así como también dispondrán de las opciones para cancelar citas o solicitar nuevas únicamente en la sucursal en la que estén afiliados

2.3 **Pac-Man**

Ver detalles http://en.wikipedia.org/wiki/Pac-Man

3. Realice las respectivas definiciones de las clases (en Java) de cada uno de los enunciados del punto 2. Las clases deben estar encapsuladas y cada una debe estar definida en un archivo .java. Tenga en cuenta que no debe desarrollar código funcional, únicamente la definición de las clases.