PROGRAMACIÓN ORIENTADA A OBJETOS

Profesor: RICHAR GUAYA

Alumno: Bryan Vicente

1º Bimestre

Abril-Agosto 2020

Tarea 5

Seguir el proceso de programación orientada a objetos de la sección 4.4 los ejercicios 1,4,7,8 y 16 y realizar la implementación de las clases en java.

Ejercicio 1:

1- Definición del problema:

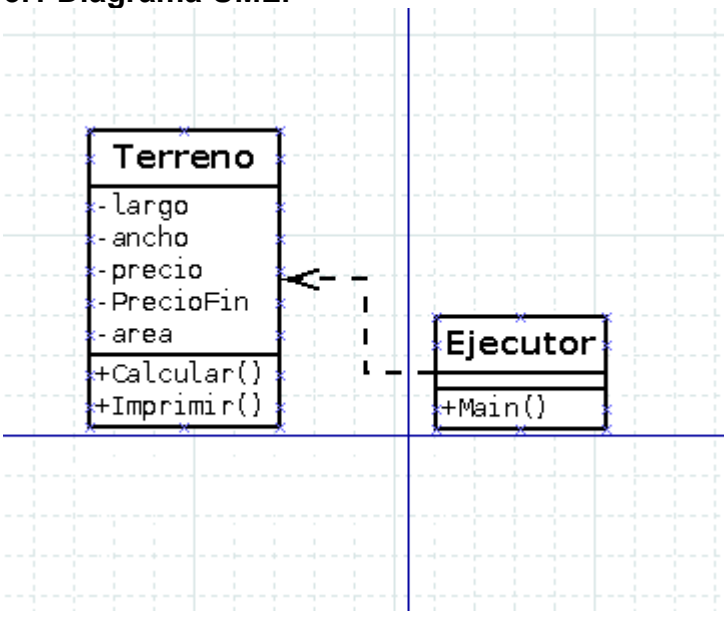
Elaborar un algoritmo para calcular e imprimir el precio de un terreno del cual se tienen los siguientes datos: largo, ancho y precio por metro cuadrado. Si el terreno tiene más de 400 metros cuadrados se hace un descuento de 10 %.

2- Análisis del problema:

- Entradas: largo, ancho, precio.
- Procesos: calcular(), imprimir().
- Salidas: precioFinal, area.

3-Diseño del programa

3.1 Diagrama UML:



3.2 Pseudocódigo:

```
//Clase Terreno

Incio  Clase Terreno:
//Creacíon de variables:

1       largo , ancho, precio, area, precioFin: Real;

//constructor
2       Inicio Constructor Terreno(decimal largo, decimal ancho, decimal precio){
                largo = largo;
                ancho = ancho;
3       Fin Constructor Terreno

//Método calcular
4       Inicio Método Calcular(){
                area = largo * ancho;
                Si (area > 400 ) entonces
                        precioFin = precio*0.9;
                Sino entonces
                        precioFin = precio
                Fin_Si
5       Fin Método Calcular

//Método Imprimir

6       Inicio Método Imprimir()
                Imprimir “El área del Terreno es: ” + area;
                Imprimir “El precio final del terreno es: “ + precioFin;
        Fin Método Imprimir

Fin ClaseTerreno

////////////////////////////////////////////////////////////////
//Clase Ejecutor

Inicio Clase Ejecutor:

//Método Main

1       Inicio Método Main()
        a       //Declarar Variables
                largo, ancho, precio: Real;
        b       //Pide los datos;
                Imprimir  “Introduzca El largo del terreno”;
                Leer largo;
```

Imprimir “Introduzca el ancho del terreno”;
Leer ancho;
Escribir “Introduzca el precio del terreno”;
Leer precio;

c //Crea el objeto terreno
Terreno terreno = new Terreno(largo , ancho ,precio);

d //Hace los calculos
terreno.Calcular();

e //Imprime los datos
terremo.Imprimir();

2 Fin Método Main

Fin Clase Ejecutor

//

Ejercicio 4:

1-Definición del problema:

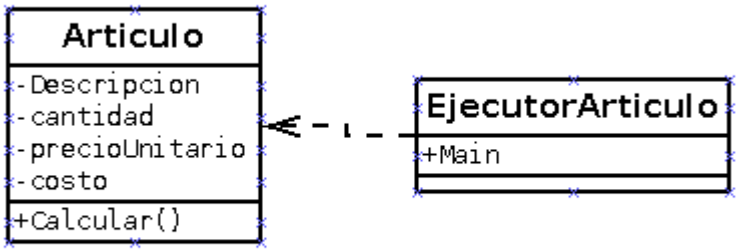
Elaborar un algoritmo que imprima el costo de un pedido de un artículo del cual se tiene la descripción, la cantidad pedida y el precio unitario. Si la cantidad pedida excede de 50 unidades, se hace un descuento de 15%.

2- Análisis del problema:

- Entrada: descripción, cantidad, precio unitario.
- Procesos: calcular()
- Salida: costo.

3-Diseño del programa

3.1 Diagrama de clases:



3.2 Pseudocódigo:

```
//Clase Articulo
Inicio Clase Articulo
1.      Declarar datos
        descripcion: Cadena
        cantidad: Entero
        precioU, costo: Real
2       Método constructor Articulo(descripcion: cadena, cantidad: entero, precioU: Real)
                descripcion = descripcion
                cantidad = cantidad
                precioU = precioU
        Fin constructor

3       Método getCosto()
                devolver costo
        Fin Método getCosto

4       Método getDescripcion()
                devolver descripcion
        Fin Método getDescripcion

5       Método Calcular()
                costo = precioU * cantidad
                if (cantidad > 50) then
                        costo = costo * 0.85
                endif
        Fin Método Calcular
Fin Clase Articulo

//////////////////////////////////////////////////////////////////

Clase EjecutorArticulo
1       Método main()
        a       Declarar variables
                descripcion: Cadena
                precioU: Real
                cantidad: Entero
        b       Solicitar descripcion, precio unitario y cantidad
        c       Leer descripcion, precioU, cantidad
        d       Inicializar el objeto articulo
                Articulo articulo = new Articulo(descripcion, precioU, cantidad)
        e       Calcular el precio
                articulo.Calcular()
```

f Imprimir “Costo total de” + articulo.getDescripcion() + “es de: ” + articulo.getCosto()
Fin Método main
Fin Clase Ejecutor

//
Ejercicio 7:

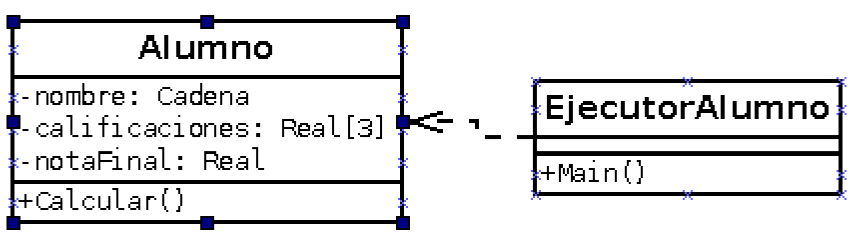
1-Definición del problema:

Elaborar un algoritmo que lea los datos de un estudiante (nombre y tres calificaciones parciales) e imprima el nombre y la califi cación fi nal de acuerdo a lo siguiente: para aprobar el curso debe tener 70 o más en cada una de las tres calificaciones, la califi cación final será el promedio. En caso de haber reprobado uno o más exámenes ordinarios, la calificación final será NA (No Acreditado).

2- Análisis del problema:
- Entrada: nombre, 3 calificaciones parciales
- Procesos: Calcular()
- Salida: notaFinal

3-Diseño del programa

3.1 Diagrama de clases:



3.2 Pseudocódigo:

Clase Alumno
1 Declarar variables
nombre : Cadena
calificaciones : Real[]
notaFinal

2 Método Constructor Alumno(nombre: cadena, calificaciones: Real[])

```
            nombre = nombre
            calificaciones = calificaciones
        Fin Método constructor

3       Método getNombre()
            return nombre
        Fin Método getNombre

4       Método Calcular()
        a       declarar variables
                media: Real
                media = 0
        b       Comprobar las calificaciones
                For (i = 0; i <3 ; i++) then
                        if (calificaciones[i] < 70) then
                                return “NA”
                        endIf
                        media = media + calificaciones[i]
                endFor

                return media /3
        Fin Método Calcular
Fin Clase Alumno
```

//

Clase EjecutarAlumno

```
1       Método main()
        a       Declar variables
                nombre : Cadena
                calificaciones : Real[3]
        b       Solicitar nombre y las 3 calificaciones
        c       Leer nobre , calificaciones
        d       Crear e iniciar el objeto Alumno
                Alumno alumno = new Alumno(nombre , calificaciones)
        e       Imprimir “La nota final de: “+alumno.getNombre()+”Es: “ + alumno.Calcular();
        Fin Método main
```

Fin Clase EjecutarAlumno

//

Ejercicio 8:

1-Definición del problema:

De acuerdo con la clase de sus ángulos, los triángulos se clasifican en:

- Rectángulo tiene un ángulo recto (igual a 90°)
- Obtusángulo tiene un ángulo obtuso (mayor que 90° pero menor 180°)
- Acutángulo los tres ángulos son agudos (menor que 90°)

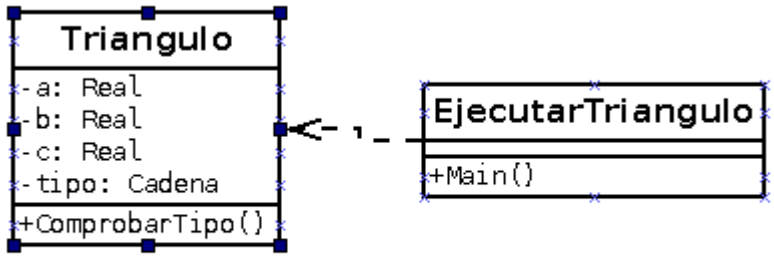
Elaborar un algoritmo que permita leer el tamaño de los tres ángulos (A,B,C) de un triángulo e imprima qué tipo es.

2- Análisis del problema:

- Entrada: 3 Angulos
- Procesos: ComprobarTipo()
- Salida: Tipo de angulo

3-Diseño del programa

3.1 Diagrama de clases:



3.2 Pseudocódigo:

```
Clase Triangulo
1       Declara Variables
        a, b, c : Real
        tipo: Cadena
2       Método constructor Triangulo(a : Real, b : Real, c: Real)
                a = a
                b = b
                c = c
        Fin Método constructor
```

```
3       Método ComprobarTipo()
        a       se declara variable
                angulos: Real
        b       Se calcula el tamaño de los angulos
                angulos = a + b + c
        c       Segun el tamaño se asigna el valor de tipo
                if(angulos = 90) then
                        tipo = “Rectángulo”;
                else if (angulos > 90 AND angulos < 180) then
                        tipo = “Obtusángulo”
                else
                        tipo = “Acutángulo”
                endIf
        d       Devolver el tipo de triangulo
                return tipo
        Fin Método ComprobarTipo
Fin Clase Triangulo
////////////////////////////////////////////////////////////////

Clase EjecutarTriangulo

1       Método Main()
        a       Declarar variables
                a, b, c : Real
        b       Solicitar angulos a , b y c
        c       Leer a, b, c
        d       Crear e inicializar el objeto Triangulo
                Triangulo triangulo = new Triangulo(a, b, c);

        e       Imprimir “El tipo de triangulo es: “ + triangulo.ComprobarTipo()
        Fin Método Main
Fin Clase EjecutarTriangulo
```

//

Ejercicio 16:

1-Definición del problema:

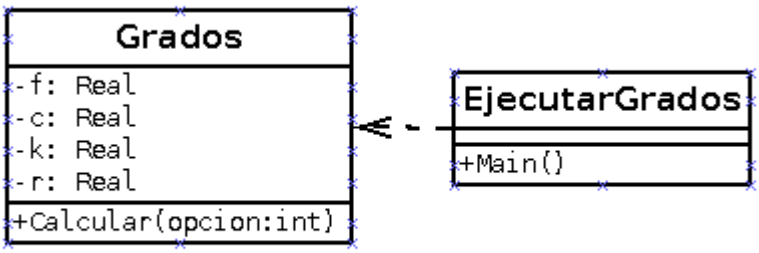
Elaborar un algoritmo que permita hacer conversiones de temperaturas entre grados Fahrenheit, Celsius, Kelvin y Rankine. Primero debe preguntar qué tipo de grados quiere convertir. Por ejemplo: si se le indica que se desea convertir una temperatura en grados Fahrenheit, debe leer la cantidad de grados y luego calcular e imprimir su equivalente en grados Celsius, Kelvin y Rankine, y así debe hacer lo mismo para cada uno de los otros tipos. Para convertir a Celsius a la temperatura Fahrenheit se le resta 32 y se multiplica por 5/9. Para convertir a Kelvin, se le suma 273 a los grados Celsius. Para convertir a Rankine a los grados Fahrenheit se le suma 460.

2- Análisis del problema:

- Entrada:Tipo de grados, cantidad de grados
- Procesos: Calcular()
- Salida: Celsius, Fahrenheit, Kelvin, Rankine

3-Diseño del programa

3.1 Diagrama de clases:



3.2 Pseudocódigo:

```
Clase Grados
1       Declarar variables
        c, f, k,r : Real
2       Métodos get y set de todas las variables

3       Método Calcular(opcion : Entero)
                switch (opcion) then
                        case 1:
                                //Con C
                                f = ((c * 5) / 9) + 32;
                                k = c + 273;
                                r = f + 460;
                               break;
                         case 2:
                               //con f
                                c = ((f - 32) * 5) / 9;
                                k = c + 273;
                               r = f + 460;
                               break;
                        case 3:
                                 //con k
                                c = k - 273;
```

```
                    f = ((c * 5) / 9) + 32;
                    r = f + 460;
                   break;
              case 4:
                   //con r
                    f = r - 460;
                   c = ((f - 32) * 5) / 9;
                   k = c + 273;
                   break;
         endSwitch
    Fin Método Calcular

Fin Clase Grados


//////////////////////////////////////////////////////////////////

Clase EjecutarGrador

1    Método Main()
    a    Declarar Variables
         opcion : Entero
         grad : Real
    b    Inicializar objeto Grados
         Grados grados = new Grados()
    c     Pedir la opcion de tipo de grado
         Leer opcion
    d    Pedir la cantidad de grados
         Leer grad

    e    switch (opcion) then
              case 1:
                   //Con C
                    grados.setC(grad);
                     break;
               case 2:
                              //con f
                    grados.setF(grad);
                    break;
          case 3:
                    //con k
                    grados.setK(grad);
                   break;
                    case 4:
                     //con r
                    grados.setR(grad);
                      break;
         endSwitch
```

```
        f       grados.calcular(opcion)

        g       imprimir("Cantidad en:"
                  + "\n Cº: " + grados.getC()
                  + "\n Fº: " + grados.getF()
                  + "\n Kº: " + grados.getK()
                  + "\n Rº: " + grados.getR());
        Fin Método Main
Fin Clase EjecutarGrados
```