

Capítulo 11

11.4 Ejercicios resueltos (Continuación...)

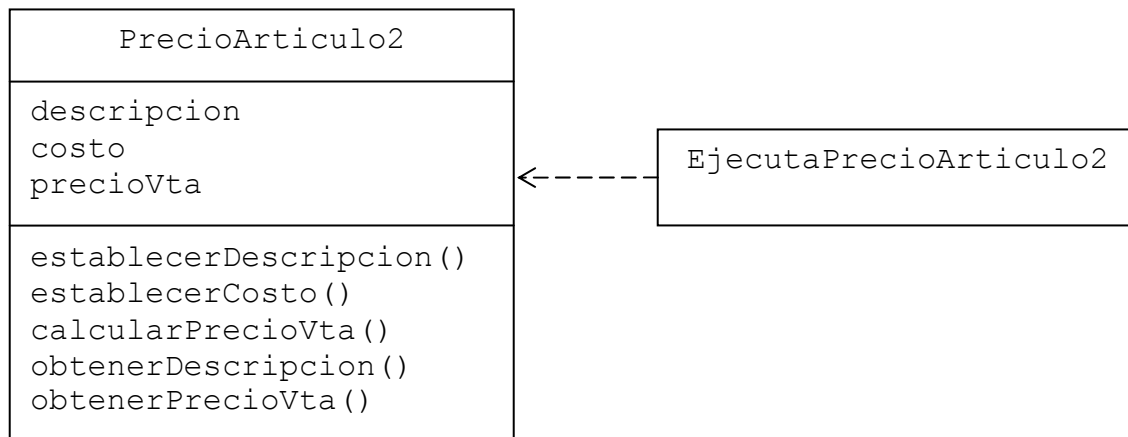
Ejercicio 11.4.4

Elaborar un algoritmo que calcule e imprima el precio de venta de un artículo. Se tienen los datos Descripción del artículo y Costo de producción. El precio de venta se calcula añadiéndole al costo el 120 % como utilidad y el 15 % de impuesto.

A continuación se presenta el algoritmo de la solución:

(Primero hágalo usted....después compare la solución)

Diagrama de clases



Algoritmo PRECIO DE VENTA

Clase PrecioArticulo2

1. Declarar

Datos

descripcion: Cadena

costo: Real

precioVta: Real

2. Método establecerDescripcion(des: Cadena)

a. descripcion = des

b. Fin Método establecerDescripcion

3. Método establecerCosto(co: Real)

a. costo = co

b. Fin Método establecerCosto

```

4. Método calcularPrecioVta()
  a. precioVta = costo + (costo*1.2) +
                    ((costo+(costo*1.2))*0.15)
  b. Fin Método calcularPrecioVta

5. Método obtenerDescripcion() Cadena
  a. return descripcion
  b. Fin Método obtenerDescripcion

6. Método obtenerPrecioVta() Real
  a. return precioVta
  b. Fin Método obtenerPrecioVta
Fin Clase PrecioArticulo2

Clase EjecutaPrecioArticulo2
1. Método principal()
  a. Declarar
      Variables
      desc: Cadena
      cos: Real
  b. Declarar, crear e iniciar objeto
      PrecioArticulo2 objArticulo = new PrecioArticulo2()
  c. Solicitar Descripción del artículo, Costo
  d. Leer desc, cos
  e. Establecer objArticulo.establecerDescripcion(desc)
      objArticulo.establecerCosto(cos)
  f. Calcular objArticulo.calcularPrecioVta()
  g. Imprimir objArticulo.obtenerDescripcion()
      objArticulo.obtenerPrecioVta()
  h. Fin Método principal
Fin Clase EjecutaPrecioArticulo2
Fin

```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: PrecioArticulo2.java y EjecutaPrecioArticulo2.java

Explicación:

El algoritmo tiene dos clases; la Clase PrecioArticulo2 y la clase EjecutaPrecioArticulo2.

En la Clase PrecioArticulo2:

1. Se declaran los datos que representan la estructura de la clase:
 - descripcion para el nombre del artículo
 - costo para el costo del artículo
 - precioVta para el precio de venta del artículo
2. Método establecerDescripcion(des: Cadena)
 - Recibe en el parámetro des el valor que luego coloca en el dato descripcion.
3. Método establecerCosto(co: Real)
 - Recibe en el parámetro co el valor que luego coloca en el dato costo.
4. Método calcularPrecioVta()

- Calcula el precio de venta del artículo
5. Método obtenerDescripcion() Cadena
Retorna descripcion del artículo
 6. Método obtenerPrecioVta() Real
Retorna precioVta del artículo
- Fin de la Clase PrecioArticulo2

En la Clase EjecutaPrecioArticulo2; en el Método principal():

- a. Se declara:
 - La variable desc para leer el nombre del artículo.
 - La variable cos para leer el costo del artículo.
 - b. Se declara el objeto objArticulo, usando como base a la clase PrecioArticulo2; dicho objeto se crea e inicializa mediante el constructor por defecto PrecioArticulo2().
 - c. Se solicitan Descripción del artículo y Costo
 - d. Se leen en desc y cos
 - e. Se llama al método establecerDescripcion(desc) del objeto objArticulo; para colocar el valor de desc en el dato descripcion .
Se llama al método establecerCosto(cos) del objeto objArticulo; para colocar el valor de cos en el dato costo.
 - f. Se llama al método calcularPrecioVta() del objeto objArticulo; para calcular el precio de venta.
 - g. Se llama al método obtenerDescripcion() del objeto objArticulo; para acceder e imprimir el valor del dato descripcion .
Se llama al método obtenerPrecioVta() del objeto objArticulo; para acceder e imprimir el valor del dato precioVta.
 - h. Fin del Método principal
- Fin de la Clase EjecutaPrecioArticulo2()
Fin del algoritmo

Ejercicio 11.4.5

Dados los lados A y B de un triángulo rectángulo, según el teorema de Pitágoras, el cuadrado de la hipotenusa (C), es igual a la suma del cuadrado de los catetos (lados).

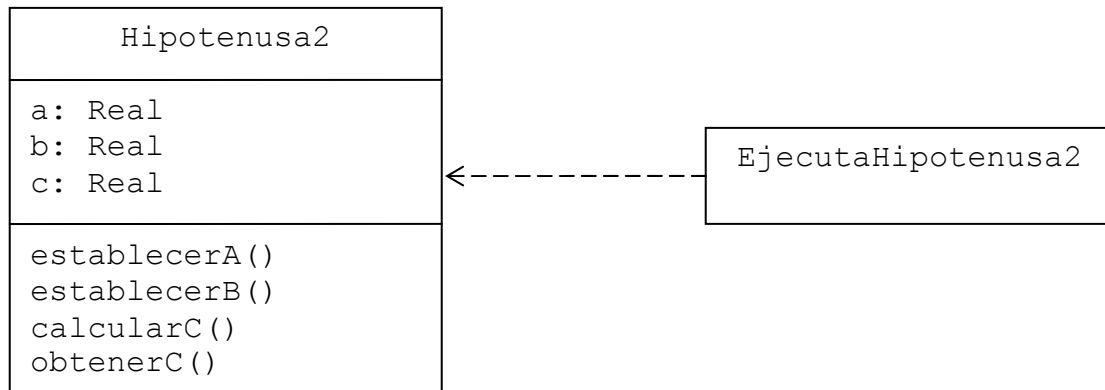
$$c^2 = a^2 + b^2$$

Elaborar un algoritmo que lea el tamaño de los lados A y B, y calcule e imprima C (hipotenusa).

A continuación se presenta el algoritmo de la solución:

(Primero hágalo usted....después compare la solución)

Diagrama de clases



Algoritmo CALCULAR HIPOTENUSA

Clase Hipotenusa2

1. Declarar

Datos

a: Real
b: Real
c: Real

2. Método establecerA(catetoA: Real)

a. a = catetoA
b. Fin Método establecerA

3. Método establecerB(catetoB: Real)

a. b = catetoB
b. Fin Método establecerB

4. Método calcularC()

a. $c = \text{RaizCuad}(\text{Potencia}(a, 2) + \text{Potencia}(b, 2))$
b. Fin Método calcularC

5. Método obtenerC() Real

a. return c
b. Fin Método obtenerC

Fin Clase Hipotenusa2

Clase EjecutaHipotenusa2

1. Método principal()

a. Declarar

VARIABLES

cateA, cateB: Real

b. Declarar, crear e iniciar objeto

Hipotenusa2 objHipotenusa = new Hipotenusa2()

c. Solicitar Cateto A y Cateto B

d. Leer cateA, cateB

e. Establecer objHipotenusa.establecerA(cateA)
objHipotenusa.establecerB(cateB)

```
f. Calcular objHipotenusa.calcularC()  
g. Imprimir objHipotenusa.obtenerC()  
h. Fin Método principal  
Fin Clase EjecutaHipotenusa2  
Fin
```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: Hipotenusa2.java y EjecutaHipotenusa2.java

Explicación:

El algoritmo tiene dos clases; la Clase Hipotenusa2 y la clase EjecutaHipotenusa2.

En la Clase Hipotenusa2:

1. Se declaran los datos que representan la estructura de la clase:
 - a para el cateto a
 - b para el cateto b
 - c para la hipotenusa
 2. Método establecerA(catetoA: Real)
Recibe en el parámetro catetoA el valor que luego coloca en el dato a.
 3. Método establecerB(catetoB: Real)
Recibe en el parámetro catetoB el valor que luego coloca en el dato b.
 4. Método calcularC()
Calcula la hipotenusa
 5. Método obtenerC() Real
Retorna c
- Fin de la Clase Hipotenusa2

En la Clase EjecutaHipotenusa2; en el Método principal():

- a. Se declara:
 - La variable cateA para leer el cateto a.
 - La variable cateB para leer el cateto b.
 - b. Se declara el objeto objHipotenusa, usando como base a la clase Hipotenusa2; dicho objeto se crea e inicializa mediante el constructor por defecto Hipotenusa2().
 - c. Se solicitan Cateto A y Cateto B
 - d. Se leen en cateA, cateB
 - e. Se llama al método establecerA(cateA) del objeto objHipotenusa; para colocar el valor de cateA en el dato a .
Se llama al método establecerB(cateB) del objeto objHipotenusa; para colocar el valor de cateB en el dato b.
 - f. Se llama al método calcularC() del objeto objHipotenusa; para calcular la hipotenusa.
 - g. Se llama al método obtenerC() del objeto objHipotenusa; para acceder e imprimir el valor del dato c .
 - h. Fin del Método principal
- Fin de la Clase EjecutaHipotenusa2
Fin del algoritmo

Ejercicio 11.4.6

En los lenguajes de programación, las funciones seno, coseno, arco tangente, etcétera, requieren que el argumento o parámetro que se les envíe, esté dado en radianes. Si el dato disponible está dado en grados, se puede hacer la conversión a radianes con la siguiente equivalencia:

$$1^\circ = \frac{\pi}{180} = 0.0174775 \text{ radianes}$$

Para convertir de radianes a grados:

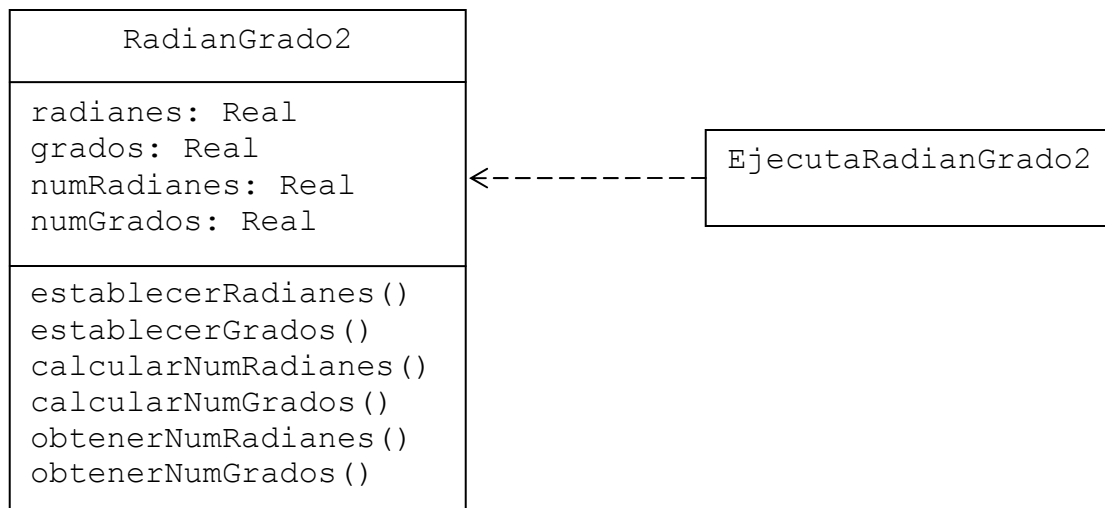
$$1 \text{ radián} = \frac{180}{\pi} = 57.21 \text{ grados}$$

Elaborar un algoritmo que permita leer un número en radianes e imprima su equivalencia en grados; asimismo, leer un número en grados e imprima su equivalencia en radianes.

A continuación se presenta el algoritmo de la solución:

(Primero hágalo usted....después compare la solución)

Diagrama de clases



Algoritmo CONVIERTE RADIANES A GRADOS Y GRADOS A RADIANES

Clase RadianGrado2

1. Declarar

Datos

```
radianes: Real
grados: Real
numRadianes: Real
numGrados: Real
```

```

2. Método establecerRadianes(rad: Real)
  a. radianes = rad
  b. Fin Método establecerRadianes

3. Método establecerGrados(gra: Real)
  a. grados = gra
  b. Fin Método establecerGrados

4. Método calcularNumRadianes()
  a. Declarar
      Constantes
      PI = 3.14159265
  b. numRadianes = grados*(PI/180)
  c. Fin Método calcularNumRadianes

5. Método calcularNumGrados()
  a. Declarar
      Constantes
      PI = 3.14159265
  b. numGrados = radianes*(180/PI)
  c. Fin Método calcularNumGrados

6. Método obtenerNumRadianes() Real
  a. return numRadianes
  b. Fin Método obtenerNumRadianes

7. Método obtenerNumGrados() Real
  a. return numGrados
  b. Fin Método obtenerNumGrados
Fin Clase RadianGrado2

Clase EjecutaRadianGrado2
1. Método principal()
  a. Declarar
      Variables
      radian, grad: Real
  b. Declarar, crear e iniciar objeto
      RadianGrado2 objRadianGrado = new RadianGrado2()
  c. Solicitar Número de radianes y Número de grados
  d. Leer radian, grad
  e. Establecer
      objRadianGrado.establecerRadianes(radian)
      objRadianGrado.establecerGrados(grad)
  f. Calcular objRadianGrado.calcularNumRadianes()
      objRadianGrado.calcularNumGrados()
  g. Imprimir objRadianGrado.obtenerNumRadianes()
      objRadianGrado.obtenerNumGrados()

```

```
    h. Fin Método principal
Fin Clase EjecutaRadianGrado2
Fin
```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: RadianGrado2.java y EjecutaRadianGrado2.java

Explicación:

El algoritmo tiene dos clases; la Clase RadianGrado2 y la clase EjecutaRadianGrado2.

En la Clase RadianGrado2:

1. Se declaran los datos que representan la estructura de la clase:
 - radianes para los radianes
 - grados para los grados
 - numRadianes para el número de radianes
 - numGrados para el número de grados
 2. Método establecerRadianes(rad: Real)
Recibe en el parámetro rad el valor que luego coloca en el dato radianes.
 3. Método establecerGrados(gra: Real)
Recibe en el parámetro gra el valor que luego coloca en el dato grados.
 4. Método calcularNumRadianes()
Se declara la constante PI
Calcula la equivalencia en número de radianes
 5. Método calcularNumGrados()
Se declara la constante PI
Calcula la equivalencia en número de grados
 6. Método obtenerNumRadianes() Real
Retorna numRadianes
 7. Método obtenerNumGrados() Real
Retorna numGrados
- Fin de la Clase RadianGrado2

En la Clase EjecutaRadianGrado2; en el Método principal():

- a. Se declara:
 - La variable radian para leer el número de radianes.
 - La variable grad para leer el número de grados.
- b. Se declara el objeto objRadianGrado, usando como base a la clase RadianGrado2; dicho objeto se crea e inicializa mediante el constructor por defecto RadianGrado2().
- c. Se solicitan Número de radianes y Número de grados
- d. Se leen en radian, grad
- e. Se llama al método establecerRadianes(radian) del objeto objRadianGrado; para colocar el valor de radian en el dato radianes .
Se llama al método establecerGrados(grad) del objeto objRadianGrado; para colocar el valor de grad en el dato grados.
- f. Se llama al método calcularNumRadianes() del objeto objRadianGrado; para calcular el número de radianes.
Se llama al método calcularNumGrados() del objeto objRadianGrado; para calcular el número de grados.

g. Se llama al método obtenerNumRadianes() del objeto objRadianGrado; para acceder e imprimir el valor del dato numRadianes.

Se llama al método obtenerNumGrados() del objeto objRadianGrado; para acceder e imprimir el valor del dato numGrados.

h. Fin del Método principal

Fin de la Clase EjecutaRadianGrado2

Fin del algoritmo