

Capítulo 12

12.4 Ejercicios resueltos (Continuación...)

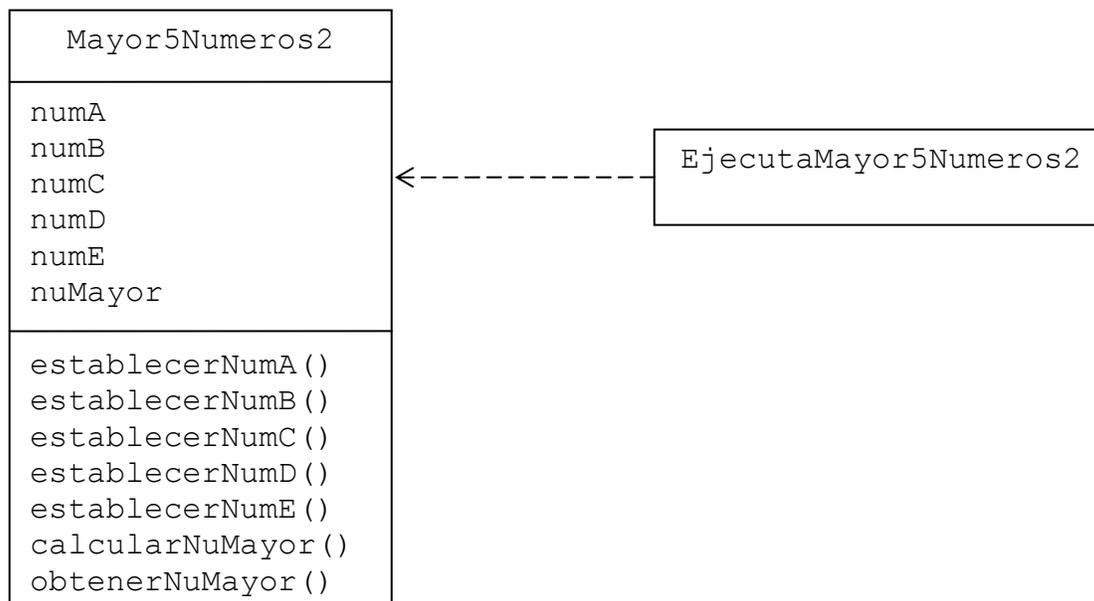
Ejercicio 12.4.4

Elaborar un algoritmo que lea cinco números y calcule e imprima el mayor. Se supone que son números diferentes. Restricciones: No usar else ni AND. Usando if-then.

A continuación se presenta el algoritmo de la solución:

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

Diagrama de clases



Algoritmo MAYOR 5 NUMEROS

Clase Mayor5Numeros2

1. Declarar

Datos

numA, numB, numC, numD, numE, nuMayor: Entero

2. Método establecerNumA(a: Entero)

a. numA = a

b. Fin Método establecerNumA

3. Método establecerNumB(b: Entero)

a. numB = b

b. Fin Método establecerNumB

4. Método establecerNumC(c: Entero)
 - a. numC = c
 - b. Fin Método establecerNumC

 5. Método establecerNumD(d: Entero)
 - a. numD = d
 - b. Fin Método establecerNumD

 6. Método establecerNumE(e: Entero)
 - a. numE = e
 - b. Fin Método establecerNumE

 7. Método calcularNuMayor()
 - a. nuMayor = numA
 - b. if numB > nuMayor then
 - a. nuMayor = numB
 - c. endif
 - d. if numC > nuMayor then
 - a. nuMayor = numC
 - e. endif
 - f. if numD > nuMayor then
 - a. nuMayor = numD
 - g. endif
 - h. if numE > nuMayor then
 - a. nuMayor = numE
 - i. endif
 - j. Fin Método calcularNuMayor

 8. Método obtenerNuMayor() Entero
 - a. return nuMayor
 - b. Fin Método obtenerNuMayor
- Fin Clase Mayor5Numeros2

Clase EjecutaMayor5Numeros2

1. Método principal()
 - a. Declarar
Variables
n1, n2, n3, n4, n5: Real
 - b. Declarar, crear e iniciar objeto
Mayor5Numeros2 objMayor5Numeros= new Mayor5Numeros2()
 - c. Solicitar Numero 1, Numero 2, Numero 3,
Numero 4 y Numero 5
 - d. Leer n1, n2, n3, n4, n5
 - e. Establecer objMayor5Numeros.establecerNumA(n1)
objMayor5Numeros.establecerNumB(n2)
objMayor5Numeros.establecerNumC(n3)
objMayor5Numeros.establecerNumD(n4)
objMayor5Numeros.establecerNumE(n5)
 - f. Calcular objMayor5Numeros.calcularNuMayor()
 - g. Imprimir objMayor5Numeros.obtenerNuMayor()
 - h. Fin Método principal

```
Fin Clase EjecutaMayor5Numeros2
Fin
```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: Mayor5Numeros2.java y EjecutaMayor5Numeros2.java

Explicación:

El algoritmo tiene dos clases; la Clase Mayor5Numeros2 y la clase EjecutaMayor5Numeros2.

En la Clase Mayor5Numeros2:

1. Se declaran los datos que representan la estructura de la clase:
numA, numB, numC, numD, numE para cada uno de los 5 números
nuMayor para número mayor
2. Método establecerNumA(a: Entero)
Recibe en el parámetro a el valor que luego coloca en el dato numA.
3. Método establecerNumB(b: Entero)
Recibe en el parámetro b el valor que luego coloca en el dato numB.
4. Método establecerNumC(c: Entero)
Recibe en el parámetro c el valor que luego coloca en el dato numC.
5. Método establecerNumD(d: Entero)
Recibe en el parámetro d el valor que luego coloca en el dato numD.
6. Método establecerNumE(e: Entero)
Recibe en el parámetro e el valor que luego coloca en el dato numE.
7. Método calcularNuMayor()
Calcula el mayor en nuMayor
 - a. Coloca numA en nuMayor
 - b. Si numB > nuMayor Entonces
 - a. Coloca numB en nuMayor
 - c. Fin del if
 - d. Si numC > nuMayor Entonces
 - a. Coloca numC en nuMayor
 - e. Fin del if
 - f. Si numD > nuMayor Entonces
 - a. Coloca numD en nuMayor
 - g. Fin del if
 - h. Si numE > nuMayor Entonces
 - a. Coloca numE en nuMayor
 - i. Fin del if
 - j. Fin Método calcularNuMayor
8. Método obtenerNuMayor() Entero
 - a. Retorna nuMayor

Fin de la Clase Mayor5Numeros2

En la Clase EjecutaMayor5Numeros2; en el Método principal():

- a. Se declara:
Las variables n1, n2, n3, n4, n5 para leer los 5 números.

- b. Se declara el objeto objMayor5Numeros, usando como base a la clase Mayor5Numeros2; dicho objeto se crea e inicializa mediante el constructor por defecto Mayor5Numeros2().
- c. Se solicitan Numero 1, Numero 2, Numero 3, Numero 4 y Numero 5
- d. Se leen en n1, n2, n3, n4, n5
- e. Se llama al método establecerNumA(n1) del objeto objMayor5Numeros; para colocar el valor de n1 en el dato numA.
Se llama al método establecerNumB(n2) del objeto objMayor5Numeros; para colocar el valor de n2 en el dato numB.
Se llama al método establecerNumC(n3) del objeto objMayor5Numeros; para colocar el valor de n3 en el dato numC.
Se llama al método establecerNumD(n4) del objeto objMayor5Numeros; para colocar el valor de n4 en el dato numD.
Se llama al método establecerNumE(n5) del objeto objMayor5Numeros; para colocar el valor de n5 en el dato numE.
- f. Se llama al método calcularNuMayor() del objeto objMayor5Numeros; para calcular el mayor.
- g. Se llama al método obtenerNuMayor() del objeto objMayor5Numeros; para acceder e imprimir el valor del dato nuMayor.
- h. Fin del Método principal
Fin de la Clase EjecutaMayor5Numeros2
Fin del algoritmo

Ejercicio 12.4.5

Elaborar un algoritmo que lea el tamaño de un ángulo en grados e imprimir si es:

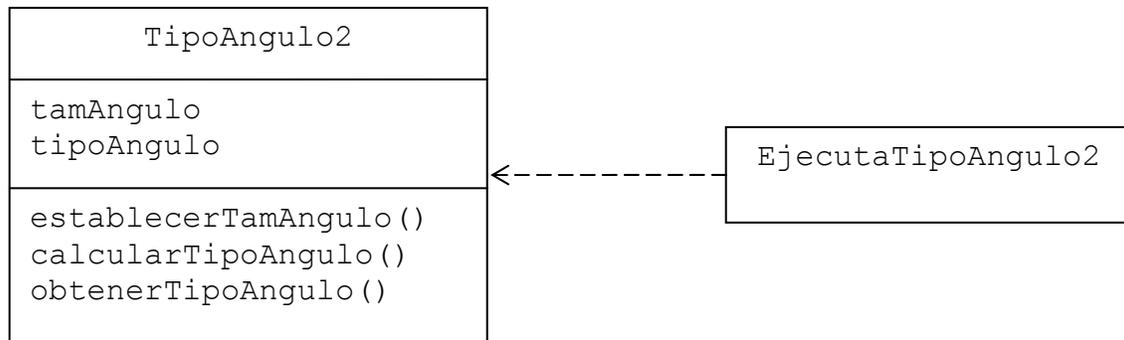
Agudo	si es menor a 90°
Recto	si es igual a 90°
Obtuso	si es mayor que 90° pero menor que 180°
Llano	si es igual a 180°
Concavo	si es mayor que 180° pero menor que 360°

Utilizar if-then.

A continuación se presenta el algoritmo de la solución:

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

Diagrama de clases



Algoritmo TIPO ANGULO

Clase TipoAngulo2

1. Declarar

Datos

tamAngulo: Real

tipoAngulo: Cadena

2. Método establecerTamAngulo(ang: Real)

a. tamAngulo = ang

b. Fin Método establecerTamAngulo

3. Método calcularTipoAngulo()

a. if tamAngulo < 90 then

1. tipoAngulo = "Agudo"

b. endif

c. if tamAngulo == 90 then

1. tipoAngulo = "Recto"

d. endif

e. if (tamAngulo > 90) AND (tamAngulo < 180) then

1. tipoAngulo = "Obtuso"

f. endif

g. if tamAngulo == 180 then

1. tipoAngulo = "Llano"

h. endif

i. if (tamAngulo > 180) AND (tamAngulo < 360) then

1. tipoAngulo = "Concavo"

j. endif

k. Fin Método calcularTipoAngulo

4. Método obtenerTipoAngulo() Cadena

a. return tipoAngulo

b. Fin Método obtenerTipoAngulo

Fin Clase TipoAngulo2

Clase EjecutaTipoAngulo2

1. Método principal()

a. Declarar

```

        Variables
        angulo: Real
    b. Declarar, crear e iniciar objeto
        TipoAngulo2 objAngulo = new TipoAngulo2()
    c. Solicitar Tamaño del angulo
    d. Leer angulo
    e. Establecer objAngulo.establecerTamAngulo(angulo)
    f. Calcular objAngulo.calcularTipoAngulo()
    g. Imprimir objAngulo.obtenerTipoAngulo()
    h. Fin Método principal
Fin Clase EjecutaTipoAngulo2
Fin

```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: TipoAngulo2.java y EjecutaTipoAngulo2.java

Explicación:

El algoritmo tiene dos clases; la Clase TipoAngulo2 y la clase EjecutaTipoAngulo2.
En la Clase TipoAngulo2:

1. Se declaran los datos que representan la estructura de la clase:
 - tamAngulo para el tamaño del ángulo
 - tipoAngulo para el tipo de angulo
 2. Método establecerTamAngulo(ang: Real)
 - Recibe en el parámetro ang el valor que luego coloca en el dato tamAngulo.
 3. Método calcularTipoAngulo()
 - a. Si tamAngulo < 90 Entonces
 1. Coloca “Agudo” en tipoAngulo
 - b. Fin del if
 - c. Si tamAngulo == 90 Entonces
 1. Coloca “Recto” en tipoAngulo
 - d. Fin del if
 - e. Si (tamAngulo>90) y (tamAngulo<180) Entonces
 1. Coloca “Obtuso” en tipoAngulo
 - f. Fin del if
 - g. Si tamAngulo == 180 Entonces
 1. Coloca “Llano” en tipoAngulo
 - h. Fin del if
 - i. Si (tamAngulo>180) y (tamAngulo<360) Entonces
 1. Coloca “Concavo” en tipoAngulo
 - j. Fin del if
 - k. Fin Método calcularTipoAngulo
 4. Método obtenerTipoAngulo() Cadena
 - a. Retorna tipoAngulo
- Fin de la Clase TipoAngulo2

En la Clase EjecutaTipoAngulo2; en el Método principal():

- a. Se declara:

- La variable angulo para leer el tamaño del angulo.
- Se declara el objeto objAngulo, usando como base a la clase TipoAngulo2; dicho objeto se crea e inicializa mediante el constructor por defecto TipoAngulo2().
 - Solicita el Tamaño del angulo
 - Se lee en angulo
 - Se llama al método establecerTamAngulo(angulo) del objeto objAngulo; para colocar el valor de angulo en el dato tamAngulo.
 - Se llama al método calcularTipoAngulo() del objeto objAngulo; para calcular el tipo de angulo.
 - Se llama al método obtenerTipoAngulo() del objeto objAngulo; para acceder e imprimir el valor del dato tipoAngulo.
 - Fin del Método principal
- Fin de la Clase EjecutaTipoAngulo2
Fin del algoritmo

Ejercicio 12.4.6

La ecuación de la segunda ley de Newton:

$$F = ma$$

Donde:

F es la fuerza que actúa sobre el objeto (cuerpo), en gramos por centímetro cuadrado.

m es la masa del cuerpo (en gramos).

a es la aceleración en centímetros por segundo cuadrado.

y dice; que la fuerza (F) que actúa sobre un cuerpo se obtiene multiplicando la masa por la aceleración.

Se puede calcular la aceleración (a) si se tienen los datos F y m; con la fórmula:

$$a = \frac{F}{m}$$

Se puede calcular la masa (m) si se tienen los datos fuerza (F) y aceleración (a),

con la fórmula:

$$m = \frac{F}{a}$$

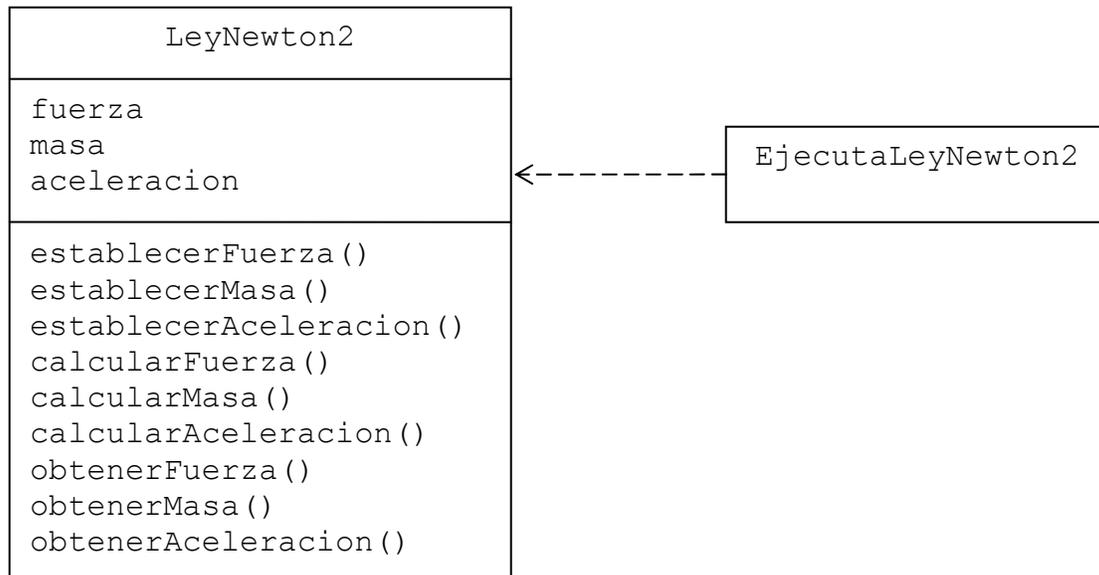
Elaborar un algoritmo que pregunte lo que desea calcular; fuerza (F), masa (m) o aceleración (a). Si es F, se leen los datos m y a. Si es a, se leen los datos F y m.

Si es m, se leen los datos F y a. Usando if-then.

A continuación se presenta el algoritmo de la solución:

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

Diagrama de clases



Algoritmo SEGUNDA LEY NEWTON

Clase LeyNewton2

1. Declarar

Datos

fuerza, masa, aceleracion: Real

2. Método establecerFuerza(fu: Real)

a. fuerza = fu

b. Fin Método establecerFuerza

3. Método establecerMasa(ma: Real)

a. masa = ma

b. Fin Método establecerMasa

4. Método establecerAceleracion(accel: Real)

a. aceleracion = accel

b. Fin Método establecerAceleracion

5. Método calcularFuerza()

a. fuerza = masa * aceleracion

b. Fin Método calcularFuerza

6. Método calcularMasa()

a. masa = fuerza / aceleracion

```

    b. Fin Método calcularMasa

7. Método calcularAceleracion()
  a. aceleracion = fuerza / masa
  b. Fin Método calcularAceleracion

8. Método obtenerFuerza() Real
  a. return fuerza
  b. Fin Método obtenerFuerza

9. Método obtenerMasa() Real
  a. return masa
  b. Fin Método obtenerMasa

10. Método obtenerAceleracion() Real
  a. return aceleracion
  b. Fin Método obtenerAceleracion
Fin Clase LeyNewton2

```

Clase EjecutaLeyNewton2

```

1. Método principal()
  a. Declarar
      Variables
          f, a, m: Real
          resp: Carácter
  b. Declarar, crear e iniciar objeto
      LeyNewton2 objNewton = new LeyNewton2()
  c. Imprimir "¿QUÉ DESEA CALCULAR? "
          "¿FUERZA(F), ACELERACION(A), MASA(M) ?: "
  d. Leer resp
  e. if resp == 'F' then
      1. Solicitar MASA
      2. Leer m
      3. Solicitar ACELERACION
      4. Leer a
      5. Establecer objNewton.establecerMasa(m)
          objNewton.establecerAceleracion(a)
      6. Calcular objNewton.calcularFuerza()
      7. Imprimir objNewton.obtenerFuerza()
  f. endif
  g. if resp == 'A' then
      1. Solicitar MASA
      2. Leer m
      3. Solicitar FUERZA
      4. Leer f
      5. Establecer objNewton.establecerMasa(m)
          objNewton.establecerFuerza(f)

```

```

        6. Calcular objNewton.calcularAceleracion()
        7. Imprimir objNewton.obtenerAceleracion()
    h. endif
    i. if resp == 'M' then
        1. Solicitar FUERZA
        2. Leer f
        3. Solicitar ACELERACION
        4. Leer a
        5. Establecer objNewton.establecerFuerza(f)
           objNewton.establecerAceleracion(a)
        6. Calcular objNewton.calcularMasa()
        7. Imprimir objNewton.obtenerMasa()
    j. endif
    k. Fin Método principal
Fin Clase EjecutaLeyNewton2
Fin

```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: LeyNewton2.java y EjecutaLeyNewton2.java

Explicación:

El algoritmo tiene dos clases; la Clase LeyNewton2 y la clase EjecutaLeyNewton2.
En la Clase LeyNewton2:

1. Se declaran los datos que representan la estructura de la clase:
fuerza para el dato fuerza
masa para el dato masa
aceleración para el dato aceleración
 2. Método establecerFuerza(fu: Real)
Recibe en el parámetro fu el valor que luego coloca en el dato fuerza.
 3. Método establecerMasa(ma: Real)
Recibe en el parámetro ma el valor que luego coloca en el dato masa.
 4. Método establecerAceleracion(accel: Real)
Recibe en el parámetro accel el valor que luego coloca en el dato aceleracion.
 5. Método calcularFuerza()
Calcula fuerza
 6. Método calcularMasa()
Calcula masa
 7. Método calcularAceleracion()
Calcula aceleracion
 8. Método obtenerFuerza() Real
Retorna fuerza
 9. Método obtenerMasa() Real
Retorna masa
 10. Método obtenerAceleracion(): Real
Retorna aceleracion
- Fin de la Clase LeyNewton2

En la Clase EjecutaLeyNewton2; en el Método principal():

a. Se declara:

La variable f para leer la fuerza

La variable a para leer la aceleracion

La variable m para leer la masa

La variable resp para leer la respuesta a la pregunta ¿Qué desea calcular?

b. Se declara el objeto objNewton, usando como base a la clase LeyNewton2; dicho objeto se crea e inicializa mediante el constructor por defecto LeyNewton2().

c. Pregunta “¿QUÉ DESEA CALCULAR?”

“¿FUERZA(F), ACELERACION(A), MASA(M) ?”

d. Lee en resp la respuesta

e. Si resp == 'F' Entonces

1. Solicita MASA

2. Leer en m

3. Solicita ACELERACION

4. Leer en a

5. Se llama al método establecerMasa(m) del objeto objNewton; para colocar el valor de m en el dato masa.

Se llama al método establecerAceleracion(a) del objeto objNewton; para colocar el valor de a en el dato aceleracion.

6. Se llama al método calcularFuerza() del objeto objNewton; para calcular la fuerza.

7. Se llama al método obtenerFuerza() del objeto objNewton; para acceder e imprimir el valor del dato fuerza.

f. Fin del if

g. Si resp == 'A' Entonces

1. Solicita MASA

2. Leer en m

3. Solicita FUERZA

4. Leer en f

5. Se llama al método establecerMasa(m) del objeto objNewton; para colocar el valor de m en el dato masa.

Se llama al método establecerFuerza(f) del objeto objNewton; para colocar el valor de f en el dato fuerza.

6. Se llama al método calcularAceleracion() del objeto objNewton; para calcular la aceleracion.

7. Se llama al método obtenerAceleracion() del objeto objNewton; para acceder e imprimir el valor del dato aceleracion.

h. Fin del if

i. Si resp == 'M' Entonces

1. Solicita FUERZA

2. Leer en f

3. Solicita ACELERACION

4. Leer en a

5. Se llama al método establecerFuerza(f) del objeto objNewton; para colocar el valor de f en el dato fuerza.

Se llama al método establecerAceleracion(a) del objeto objNewton; para colocar el valor de a en el dato aceleracion.

6. Se llama al método calcularMasa() del objeto objNewton; para calcular la masa.
7. Se llama al método obtenerMasa() del objeto objNewton; para acceder e imprimir el valor del dato masa.

j. Fin del if

k. Fin del Método principal

Fin de la Clase EjecutaLeyNewton2

Fin del algoritmo

Ejercicio 12.4.7

Una empresa vende hojas de hielo seco, con las condiciones siguientes:

Si el cliente es tipo 1 se le descuenta el 5 %

Si el cliente es tipo 2 se le descuenta el 8 %

Si el cliente es tipo 3 se le descuenta el 12 %

Si el cliente es tipo 4 se le descuenta el 15 %

Cuando el cliente realiza una compra se generan los datos siguientes:

Nombre del cliente

Tipo de cliente (1,2,3,4)

Cantidad de hojas

Precio por hoja

Elabore un algoritmo que lea estos datos, haga cálculos e imprima:

Nombre del cliente

Sub total a pagar (Cantidad de hojas X Precio por hoja)

Descuento (El porcentaje correspondiente del Sub total a pagar)

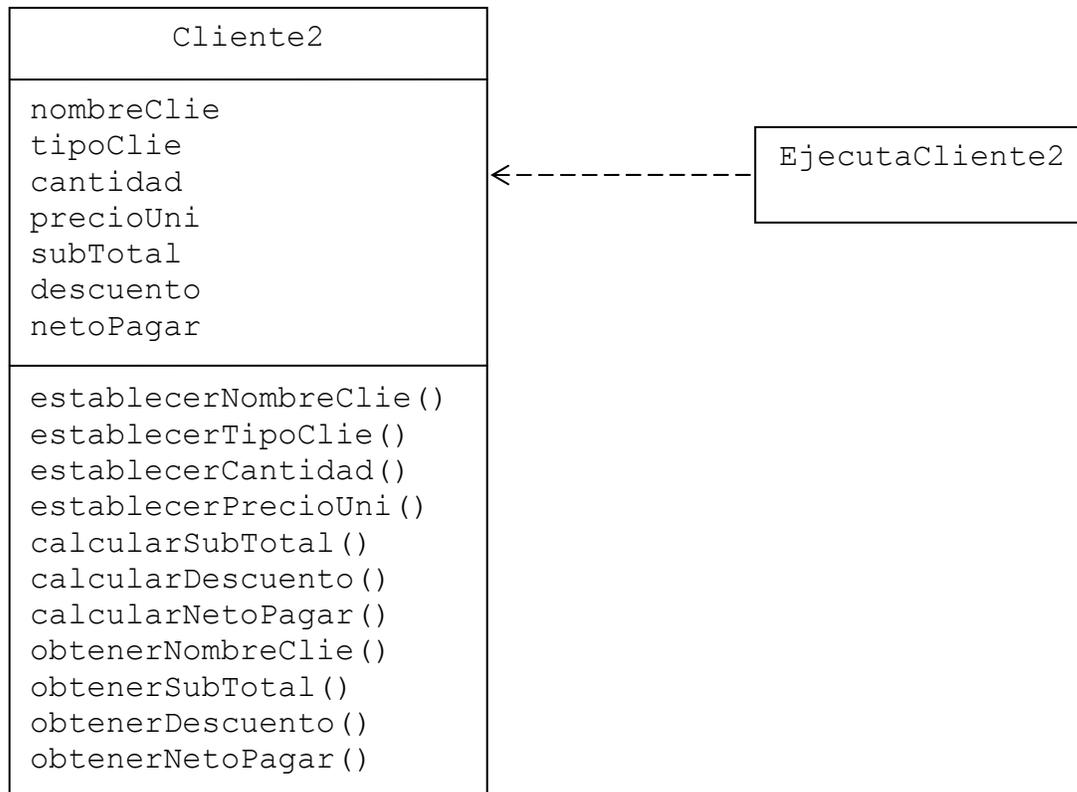
Neto a pagar (Sub total – Descuento)

Usando switch.

A continuación se presenta el algoritmo de la solución:

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

Diagrama de clases



Algoritmo CLIENTE HOJAS HIELO SECO

Clase Cliente2

1. Declarar

Datos

nombreClie: Cadena

tipoClie, cantidad: Entero

precioUni, subTotal, descuento, netoPagar: Real

2. Método establecerNombreClie(nom: Cadena)

a. nombreClie = nom

b. Fin Método establecerNombreClie

3. Método establecerTipoClie(tipo: Entero)

a. tipoClie = tipo

b. Fin Método establecerTipoClie

4. Método establecerCantidad(can: Entero)

a. cantidad = can

b. Fin Método establecerCantidad

5. Método establecerPrecioUni(pre: Real)

a. precioUni = pre

b. Fin Método establecerPrecioUni

```

6. Método calcularSubTotal()
  a. subTotal = cantidad * precioUni
  b. Fin Método calcularSubTotal

7. Método calcularDescuento()
  a. switch tipoClie
      1: descuento = subTotal * 0.05
      2: descuento = subTotal * 0.08
      3: descuento = subTotal * 0.12
      4: descuento = subTotal * 0.15
  b. endswitch
  c. Fin Método calcularDescuento

8. Método calcularNetoPagar()
  a. totalPagar = subTotal - descuento
  b. Fin Método calcularNetoPagar

9. Método obtenerNombreClie() Cadena
  a. return nombreClie
  b. Fin Método obtenerNombreClie

10. Método obtenerSubTotal() Real
  a. return subTotal
  b. Fin Método obtenerSubTotal

11. Método obtenerDescuento() Real
  a. return descuento
  b. Fin Método obtenerDescuento

12. Método obtenerNetoPagar() Real
  a. return totalPagar
  b. Fin Método obtenerNetoPagar
Fin Clase Cliente2

Clase EjecutaCliente2
1. Método principal()
  a. Declarar
      Variables
      nomCli: Cadena
      tiCli, cant: Entero
      preUni: Real
  b. Declarar, crear e iniciar objeto
      Cliente2 objCliente = new Cliente2()
  c. Solicitar Nombre, Tipo cliente, Cantidad,
      Precio unitario
  d. Leer nomCli, tiCli, cant, preUni
  e. Establecer objCliente.establecerNombreClie(nomCli)
      objCliente.establecerTipoClie(tiCli)

```

```

        objCliente.establecerCantidad(cant)
        objCliente.establecerPrecioUni(preUni)
    f. Calcular objCliente.calcularSubTotal()
        objCliente.calcularDescuento()
        objCliente.calcularNetoPagar()
    g. Imprimir objCliente.obtenerNombreClie()
        objCliente.obtenerSubTotal()
        objCliente.obtenerDescuento()
        objCliente.obtenerNetoPagar()
    h. Fin Método principal
Fin Clase EjecutaCliente2
Fin

```

En la zona de descarga de la Web del libro, está disponible:
Programa en Java: Cliente2.java y EjecutaCliente2.java

Explicación:

El algoritmo tiene dos clases; la Clase Cliente2 y la clase EjecutaCliente2.
En la Clase Cliente2:

1. Se declaran los datos que representan la estructura de la clase:
 - nombreClie para el nombre del cliente
 - tipoClie para el tipo de cliente
 - cantidad para la cantidad de hojas compradas
 - precioUni para el precio unitario
 - subTotal para el sub total
 - descuento para el descuento
 - netoPagar para el total a pagar
2. Método establecerNombreClie(nom: Cadena)
 - Recibe en el parámetro nom el valor que luego coloca en el dato nombreClie.
3. Método establecerTipoClie(tipo: Entero)
 - Recibe en el parámetro tipo el valor que luego coloca en el dato tipoClie.
4. Método establecerCantidad(can: Entero)
 - Recibe en el parámetro can el valor que luego coloca en el dato cantidad.
5. Método establecerPrecioUni(pre: Real)
 - Recibe en el parámetro pre el valor que luego coloca en el dato precioUni.
6. Método calcularSubTotal()
 - Calcula el subTotal a pagar
7. Método calcularDescuento()
 - Calcula el descuento
 - a. switch tipoClie
 - Si tipoClie==1: descuento = subTotal * 0.05
 - Si tipoClie==2: descuento = subTotal * 0.08
 - Si tipoClie==3: descuento = subTotal * 0.12
 - Si tipoClie==4: descuento = subTotal * 0.15
 - b. endswitch
 - c. Fin Método calcularDescuento
8. Método calcularNetoPagar()

- Calcula el netoPagar a pagar
9. Método obtenerNombreClie() Cadena
Retorna nombreClie
 10. Método obtenerSubTotal() Real
Retorna subTotal
 11. Método obtenerDescuento() Real
Retorna descuento
 12. Método obtenerNetoPagar() Real
Retorna netoPagar
- Fin de la Clase Cliente2

En la Clase EjecutaCliente2; en el Método principal():

- a. Se declara:
 - La variable nomCli para leer el nombre del cliente.
 - La variable tiCli para leer el tipo de cliente.
 - La variable cant para leer la cantidad de hojas compradas.
 - La variable preUni para leer el precio unitario por hoja.
 - b. Se declara el objeto objCliente, usando como base a la clase Cliente2; dicho objeto se crea e inicializa mediante el constructor por defecto Cliente2().
 - c. Solicita Nombre, Tipo cliente, Cantidad,
Precio unitario
 - d. Lee en nomCli, tiCli, cant, preUni
 - e. Se llama al método establecerNombreClie(nomCli) del objeto objCliente; para colocar el valor de nomCli en el dato nombreClie.
Se llama al método establecerTipoClie(tiCli) del objeto objCliente; para colocar el valor de tiCli en el dato tipoClie.
Se llama al método establecerCantidad(cant) del objeto objCliente; para colocar el valor de cant en el dato cantidad.
Se llama al método establecerPrecioUni(preUni) del objeto objCliente; para colocar el valor de preUni en el dato precioUni.
 - f. Se llama al método calcularSubTotal() del objeto objCliente; para calcular el sub total.
Se llama al método calcularDescuento() del objeto objCliente; para calcular el descuento.
Se llama al método calcularNetoPagar() del objeto objCliente; para calcular el neto a pagar.
 - g. Se llama al método obtenerNombreClie() del objeto objCliente; para acceder e imprimir el valor del dato nombreClie.
Se llama al método obtenerSubTotal() del objeto objCliente; para acceder e imprimir el valor del dato subTotal.
Se llama al método obtenerDescuento() del objeto objCliente; para acceder e imprimir el valor del dato descuento.
Se llama al método obtenerNetoPagar() del objeto objCliente; para acceder e imprimir el valor del dato netoPagar.
 - h. Fin del Método principal
- Fin de la Clase EjecutaCliente2
Fin del algoritmo