

# programación

EJERCICIOS COMPLEMENTARIOS AL LIBRO

**CFGS de Desarrollo  
de aplicaciones  
multiplataforma**

**CFGS de desarrollo  
de aplicaciones web**

## ■ UNIDAD 3

### Identificación de los elementos de un programa informático

#### 3.1. ESTRUCTURAS Y BLOQUES FUNDAMENTALES

Ejemplo para ver manejar el recorrido de un proyecto (con varias clases):

##### Ejemplo

###### CLASE PRINCIPAL

```
package ejercicio1;

public class Principal
{
    public static void main(String args[])
    {
        System.out.println("Ha entrado usted en el main de la clase principal");
        Clase1 objclase1=new Clase1();
        objclase1.mensajeClase1();
    }
}
```

###### CLASE 1

```
package ejercicio1;

public class Clase1
{
    public void mensajeClase1( )
    {
        System.out.println("Este mensaje lo lanza el método mensajeClase1 de la
                           clase 1");

        Clase2 objclase2=new Clase2();
        objclase2.mensajeClase2();
    }
}
```

###### CLASE 2:

```
package ejercicio1;
```

### Ejemplo (cont.)

```
public class Clase2
{
    public void mensajeClase2()
    {
        System.out.println("Este mensaje lo lanza el método mensajeClase2 de la
                           clase 2");
    }
}
```

## 3.4. VARIABLES. DECLARACIÓN, INICIALIZACIÓN Y UTILIZACIÓN. ALMACENAMIENTO EN MEMORIA

Ejemplo para ver el uso de las variables:

El objetivo es ver la diferencia entre variables de la clase y variables locales.

### Ejemplo

#### CLASE 1

```
package ejercicio1;

public class Clase1 {

    int varmiembroClase1;

    public Clase1() { }

    public void mensaje1Clase1()
    {
        double varlocalClase1=12.3;

        System.out.println("Este mensaje lo lanza el método mensaje1 la clase 1");

        System.out.println("El valor de la variable miembro de la clase Clase1 es
                           " + varmiembroClase1);
        System.out.println("El valor de la variable local (del método mensaje1) de
                           la clase Clase1 es " + varlocalClase1);

        varlocalClase1=varlocalClase1+2;
        System.out.println("El valor de la variable local de la clase Clase1
                           (después del cambio) es" + varlocalClase1);

        varmiembroClase1=8;
    }
}
```

### Ejemplo (cont.)

```

        System.out.println("El valor de la variable miembro de la clase Clase1
                           (después del cambio) es " + varmiembroClase1);

        Clase2 objclase2=new Clase2();
        objclase2.mensajeClase2();

        mensaje2Clase1();

        System.out.println("El valor de la variable miembro de la clase Clase1
                           (después de ejecutar el método mensaje2Clase1( )) es "
                           + varmiembroClase1);

    }

    public void mensaje2Clase1()
    {
        double varlocalClase1=17.8;

        System.out.println("Este mensaje lo lanza el método mensaje2 la clase
                           1");
        System.out.println("El valor de la variable miembro de la clase Clase1 es
                           " + varmiembroClase1);
        System.out.println("El valor de la variable local (del método mensaje2) de
                           la clase Clase1 es " + varlocalClase1);

        varlocalClase1=varlocalClase1+5;
        System.out.println("El valor de la variable local de la clase Clase1
                           (después del cambio) es" + varlocalClase1);

        varmiembroClase1=60;
        System.out.println("El valor de la variable miembro de la clase Clase1
                           (después del cambio) es " + varmiembroClase1);
    }
}

```

### CLASE 2:

```

package ejercicio1;

public class Clase2 {

    public Clase2() { }
    public void mensajeClase2()
    {
        char varlocalClase2='p';
        System.out.println("Este mensaje lo lanza un método de la clase 2");
    }
}

```

### Ejemplo (cont.)

```

        System.out.println("El valor de varlocalClase2 es " + varlocalClase2);

        varlocalClase2='m';
        System.out.println("El valor de varlocalClase2 (después del cambio) es "
                           + varlocalClase2);
    }
}

```

### CLASE PRINCIPAL

```

package ejercicio1;

public class Principal {
    public Principal() { }

    public static void main(String args[])
    {
        int varmain=8;

        System.out.println("Ha entrado usted en la clase principal");
        System.out.println("El valor de la variable del main es " + varmain);

        Clase1 objclase1=new Clase1();
        objclase1.mensajelClase1();

        varmain=15;
        System.out.println("El valor de la variable del main (después del cambio)
                           es " + varmain);
    }
}

```

### 3.9. OPERADORES Y EXPRESIONES. PRECEDENCIA DE OPERADORES

Ejemplo para ver el uso de los operadores ++ y -

#### Ejemplo

```
package borrar;

public class Principal {

    public static void main(String args[])
    {
        Principal obj=new Principal();
        obj.modoprefijo();
        obj.modosufijo();
    }

    void modoprefijo ( )
    {
        /*modo prefijo*/

        /*primero incrementa la variable y después realiza la asignación*/

        int a=1,b;

        b=++a;

        System.out.println("\n\nDespués de realizar la siguiente asignación: b=++a,
                               el valor de las dos variables son: ");
        System.out.println("\n\n\n\t a = " + a + " \t b = " + b + "\n\n\n"); /*a=2
                                                                                   b=2*/
    }

    void modosufijo( )
    {
        int a=1,b;

        b=a++;
        /*modo sufijo*/

        /*primero realiza la asignación y después incrementa la variable*/
        System.out.println("\nDespués de realizar la siguiente asignación: b=a++,
                               el valor de las dos variables son: ");
        System.out.println("\n\n\n\t a = " + a + " \t b = " + b + "\n\n\n"); /*a=2
                                                                                   b=2*/
    }
}
```

Otro ejemplo para ver el uso de los operadores ++ y -

### Ejemplo

```
public class Principal
{
    public static void main(String args[])
    {
        int edad=8, edadnueva, otraedad=5;

        System.out.println("PRIMERA PRUEBA: ");
        System.out.println("El valor de la variable edad es "+ edad);
        System.out.println("\n\nEl valor de la variable otraedad es "+ otraedad);

        edad++;
        System.out.println("El valor de la variable edad es (9) "+ edad);
        System.out.println("El valor de la variable otraedad es (5) "+ otraedad);

        ++edad;
        System.out.println("El valor de la variable edad es (10)" + edad);
        System.out.println("El valor de la variable otraedad es (5)" + otraedad);

        edadnueva=edad++;
        System.out.println("El valor de la variable edad es (11)" + edad);
        System.out.println("El valor de la variable otraedad es (5)" + otraedad);
        System.out.println("El valor de la variable nuevaedad es (10) "+
            edadnueva);

        edadnueva=++edad;
        System.out.println("El valor de la variable edad es (12)" + edad);
        System.out.println("El valor de la variable otraedad es (5)" + otraedad);
        System.out.println("El valor de la variable nuevaedad es (12)" +
            edadnueva);

        edadnueva=otraedad++ + ++edad;
        System.out.println("El valor de la variable edad es (13)" + edad);
        System.out.println("El valor de la variable otraedad es (6)" + otraedad);
        System.out.println("El valor de la variable nuevaedad es (18) "+
            edadnueva);

    }
}
```

### 3.10. CONVERSIONES DE TIPO. IMPLÍCITAS Y EXPLÍCITAS (CASTING)

#### Conversión implícita

#### EJEMPLOS DE CASOS DONDE NO SE PRODUCE LA CONVERSIÓN IMPLÍCITA

El siguiente ejemplo, de conversión implícita, provocaría errores.

Da error en las tres asignaciones, ya que pasamos de un tipo mayor a un tipo menor, con lo cual, se pierde información.

#### Ejemplo

```
package ejercicio1;

public class Principal {

    public static void main(String args[])
    {
        int int1,int2;
        char chark ;
        byte bytes=13;
        float floath=9.2f; /*Se pone la f ya que sino me da error ya que me
                           obliga a que la variable sea de tipo double*/
        long longc=9;

        // Conversiones implícitas.

        int1 = longc; // Error. El tipo de destino es menor que el tipo de origen.
        chark=bytes; /* Cuando la variable destino es "char" y el origen es
                     numérico,independientemente del tipo específico,la conversión no
                     podrá ser automática.*/
        int2 = floath; /* Cuando la variable destino es entera y el origen es
                     decimal (float o double), la conversión no podrá ser
                     automática.*/

        System.out.println("El valor de longc (tipo long) es " + longc + " y ese
                           valor en un int ha quedado en " + int1 );
        System.out.println("El valor de bytes (tipo byte) es " + bytes + " y ese
                           valor en un char ha quedado en " + chark );
        System.out.println("El valor de floath (tipo float) es " + floath + " y ese
                           valor en un int ha quedado en " + int2 );

    }
}
```



## CONVERSIÓN EXPLÍCITA

### Ejemplo

```
package ejercicio2;

public class Principal {
    public Principal() { }

    public static void main(String args[])
    {
        int intp=400;
        char charc ;
        byte bytek=13;
        double doubled=34.6;

        // Conversiones explícitas.

        charc = (char)doubled; /*Se elimina la parte decimal (trunca). No se
                               redondea*/

        bytek=(byte)intp; /* Se provoca una pérdida de datos, pero la conversión
                           es posible*/

        System.out.println("El valor de doubled (tipo double) es " + doubled + " y
                           ese valor en un char ha quedado en " + charc);

        System.out.println("El valor de intp (tipo int) es " + intp + " y ese valor
                           en un byte ha quedado en " + bytek );

    }
}
```

### Ejemplo

En este ejemplo vamos a ver una utilidad, muy importante, de la conversión de tipos y es el hecho de obtener decimales (en una división).

En este ejercicio hay que tener muy presente que Java no admite que metas un dato superior en una variable inferior como no sea, a través, de un casting.

```
package ejercicio3;

public class Principal {
    public Principal() { }

    public static void main(String args[])
    {
        int e1=6, e2=4, re1, re2, re3;
        float f1=6, f2=4, rf1, rf2, rf3, rf4, rf5;
```

### Ejemplo (cont.)

```

rel=(int)f1/e2; /*En C y C++ el int no hacia falta, pero en Java hay
                que hacer la conversión, sino no me deja guardar algo
                superior en algo inferior.*/

re2=e1/ e2;
rf1=f1 / f2;
rf2=(float)e1/e2;

rf3=f1 /e2;
rf4=e1/e2;

System.out.println(f1 + "\t:\t" + e2 + "= " +rel);
System.out.println(e1 + "\t:\t" + e2 + "= " +re2);

System.out.println(f1 + "\t:\t" + f2 + "= " +rf1);
System.out.println("(float)" + e1 + "\t:\t" + e2 + "= " +rf2);
System.out.println(f1 + "\t:\t" + e2 + "= " +rf3);
System.out.println(e1 + "\t:\t" + e2 + "= " +rf4);
    }
}

```

### Ejercicio

Estudia cual sería el resultado del siguiente programa:

```

public class Principal
{
    public static void main(String args[ ])
    {
        int e1=4, e2=3,rel;
        float rf1,rf2,rf3;

        rel=e1 +(float)e2/2;
        rf1=e1 + (float)e2/2;
        rf2=(float)e1 + e2/2;
        rf3=((float)e1 + e2)/2;

        System.out.println(" El resultado de la suma de \"rel=e1 +(float)
                           e2/2;\" es " + rel);
        System.out.println(" El resultado de la suma de \"rf1=e1 + (float)
                           e2/2;\" es " + rf1);
        System.out.println(" El resultado de la suma de \"rf2=(float)e1 +
                           e2/2;\" es " + rf2);
        System.out.println(" El resultado de la suma de \"rf3=((float)e1 +
                           e2)/2;\" es " + rf3);
    }
}

```

### Solución

La instrucción `re1=e1 +(float)e2/2;` daría un error de codificación ya que pretendo insertar un float en un int.

Las demás instrucciones visualizaría los siguientes mensajes:

El resultado de la suma de `"rf1=e1 + (float)e2/2;"` es 5.5

El resultado de la suma de `"rf2=(float)e1 + e2/2;"` es 5.0

El resultado de la suma de `"rf3=((float)e1 + e2)/2;"` es 3.5

## Ejercicios de repaso de tema 3

1 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1,num2,suma;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
    cadena=teclado.readLine();
    num2=Integer.parseInt(cadena);

    suma=num1 + num2;

    if( expresión a determinar)
        System.out.println("La suma de los dos números es " + suma);
    else
        System.out.println("Los dos números insertados son " + num1 + ", " +
                            num2);
}
```

Lo que hace es pedir dos números. Si la suma, de dichos números, da un resultado positivo, se visualizará dicha suma. Si no es así, se visualizará el valor de dichos números.

### Solución

```
SUMA>0
```

2 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    if( expresión a determinar)
        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}
```

Lo que hace es pedir un número. Si dicho número está entre 1 y 100 (ambos inclusive) se visualizará el mensaje: "correcto".

### Solución

```
num1>=1 && num1<=100
```

3 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    if( expresión a determinar)
```

```

        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}

```

Lo que hace es pedir un número. Si dicho número no está entre 1 y 100 (ambos inclusive) se visualizará el mensaje: "correcto".

### Solución

```
num1<1 || num1>100
```

**4** Dado el siguiente programa:

```

public static void main(String args[]) throws IOException
{
    int num1,num2;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
    cadena=teclado.readLine();
    num2=Integer.parseInt(cadena);

    if( expresión a determinar)
        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}

```

Lo que hace es pedir dos números. Si los dos números están entre 1 y 100 (ambos inclusive), se visualizará el mensaje: "correcto".

### Solución

```
(num1>=1 && num1 <=100) && (num2>=1 && num2<=100)
```

5 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1,num2;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
    cadena=teclado.readLine();
    num2=Integer.parseInt(cadena);

    if( expresión a determinar)
        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}
```

Lo que hace es pedir dos números. Si alguno de los dos números están entre 1 y 100 (ambos inclusive), se visualizará el mensaje: "correcto".

### Solución

```
NUM1>=1 && NUM1<=100 || NUM2>=1 && NUM2<=100
```

6 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1,num2;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
```

```
cadena=teclado.readLine();
num2=Integer.parseInt(cadena);

if( expresión a determinar)
    System.out.println("\n\tCORRECTO");

System.out.println("\n\nFIN DEL PROGRAMA");
}
```

Lo que hace es pedir dos números. Si el primer número está entre 1 y 100 (ambos inclusive) y el segundo no, se visualizará el mensaje: "correcto".

### Solución

```
(NUM1>=1 && NUM1<=100) && (NUM2<1 || NUM2>100)
```

7 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1,num2;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
    cadena=teclado.readLine();
    num2=Integer.parseInt(cadena);

    if( expresión a determinar)
        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}
```

Lo que hace es pedir dos números. Si ninguno de ellos, están entre 1 y 100 (ambos inclusive), se visualizará el mensaje: "correcto".

### Solución

```
(num1<1 || num1>100) && (num2<1 || num2>100)
```

8 Dado el siguiente programa:

```
public static void main(String args[]) throws IOException
{
    int num1,num2;
    String cadena;

    InputStreamReader flujo=new InputStreamReader(System.in);
    BufferedReader teclado=new BufferedReader(flujo);

    System.out.print("\nInserte el primer número: ");
    cadena=teclado.readLine();
    num1=Integer.parseInt(cadena);

    System.out.print("\nInserte el segundo número: ");
    cadena=teclado.readLine();
    num2=Integer.parseInt(cadena);

    if( expresión a determinar)
        System.out.println("\n\tCORRECTO");

    System.out.println("\n\nFIN DEL PROGRAMA");
}
```

Lo que hace es pedir dos números. Si alguno de los dos números no están entre 1 y 100 (ambos inclusive), se visualizará el mensaje: "correcto".

### Solución

```
(num1<1 || num1>100) || (num2<1 || num2>100)
```



## ■ UNIDAD 4

### Utilización de los objetos

#### 4.3. UTILIZACIÓN DE MÉTODOS. PARÁMETROS Y VALORES DE RETORNO

##### PARÁMETROS

##### Ejemplo

###### Clase Empleado

```
package metodos;
import java.io.*;
public class Empleado
{
    int antiguedad;
    int edad;
    String nombre;

    public void pedirTodosDatos () throws IOException
    {
        String cadedad;

        InputStreamReader flujo=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(flujo);

        System.out.print("\n\tIndique el nombre del empleado: ");
        nombre=teclado.readLine()
        System.out.print("\n\tIndique la edad de " + nombre + ":\t");
        cadedad=teclado.readLine();
        edad=Integer.parseInt(cadedad);

        antiguedad=0;//Si no lo pongo, le daría el mismo valor.
    }

    /*El método siguiente en un método con parámetros, con lo cual, cuando
    llame a este método tiene que recibir,como argumento, un número entero.
    Dicho número entero lo recoge la variable ed para guardarlo en la variable del
    objeto*/

    public void cambiarEdad(int ed)
    {
        edad=ed;
    }

    /*El método siguiente en un método con parámetros, con lo cual, cuando
    llame a este método tiene que recibir, como argumentos, una cadena y un
```

### Ejemplo (cont.)

número entero, que será el nombre del alumno y la edad. Dichos datos lo recogerán las variables `nomenv` y `edadenv` para guardarlos en las variables del objeto.\*/

```
public void InicDatosEnviados(String nomenv, int edadenv )
{
    nombre=nomenv;
    edad=edadenv;
}

public void visualizarDatos( )
{
    System.out.println("\tNombre:\t\t" + nombre);
    System.out.println("\tEdad:\t\t" + edad);
    System.out.println("\t\t" + antiguedad);
}
}
```

Por ello, la clase Principal, que hace uso de la clase Empleado, podría ser esta:

### Clase Principal

```
package metodos;
import java.io.*;

public class Principal
{

    public static void main(String args[]) throws IOException
    {
        Empleado empl,emp2;

        empl=new Empleado( );
        emp2=new Empleado( );

        System.out.println("\n\nVamos a pedir todos los datos del empleado 1: ");
        empl.pedirTodosDatos( );

        System.out.println("\n\n\nDel empleado 2 no pedimos datos, ya que se le
                           asigna unos datos concretos.");
        /*En la llamada al método siguiente, como es un método con parámetros,
        hay que enviarle los valores correspondientes que espera recibir.*/
        emp2.InicDatosEnviados("David Abad", 28);

        System.out.println("\n\nLos datos del empleado 1 son: ");
        empl.visualizarDatos();

        System.out.println("\n\nLos datos del empleado 2 son: ");
```

### Ejemplo (cont.)

```
emp2.visualizarDatos();

System.out.println("\n\n\nFIN DE PROGRAMA");

} // cierra main

} // cierra clase
```

### Ejercicio

Modifica el ejemplo anterior, asignando a cada empleado una antigüedad que será indicada por el teclado. Dicha antigüedad es la misma para los dos empleados por ello, se pedirá una sola vez. Posteriormente, se visualizará los datos de dichos empleados.

Para ello lo que haremos es pedir la antigüedad en el main y enviársela a cada uno de los objetos.

### Solución

```
package paquete;
import java.io.*;

public class Principal
{
    public static void main(String args[]) throws IOException
    {
        Empleado empl,emp2;
        int antig;
        InputStreamReader flujo=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(flujo);
        String antigCad;

        empl=new Empleado( );
        emp2=new Empleado( );

        System.out.println("\n\nVamos a pedir todos los datos del empleado 1: ");
        empl.pedirTodosDatos( );

        System.out.println("\n\n\nDel empleado 2 no pedimos datos, ya que se le
                           asigna unos datos ya dados.");

        /*En la llamada al método siguiente como es un método con parámetros
        hay que enviarle los valores correspondientes que dicho método espera
        recibir.*/
        emp2.InicDatosEnviados("David Abad", 28);

        System.out.print("Inserte la antigüedad de los dos: ");
        antigCad=teclado.readLine();
```

## Solución (cont.)

```

        antig=Integer.parseInt(antigCad);

        empl1.asignarantig(antig);
        emp2.asignarantig(antig);

        System.out.println("\n\nLos datos del empleado 1 son: ");
        empl1.visualizarDatos();

        System.out.println("\n\nLos datos del empleado 2 son: ");
        emp2.visualizarDatos();

        System.out.println("\n\n\nFIN DE PROGRAMA");
    } // cierra main
} // cierra clase

package paquete;
import java.io.*;

public class Empleado
{
    int antiguedad;
    int edad;
    String nombre;

    public void pedirTodosDatos () throws IOException
    {
        String cadedad;

        InputStreamReader flujo=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(flujo);

        System.out.print("\n\tIndique el nombre del empleado: ");
        nombre=teclado.readLine();

        System.out.print("\n\tIndique la edad de " + nombre + ":\t");
        cadedad=teclado.readLine();
        edad=Integer.parseInt(cadedad);

        antiguedad=0;//Si no lo pongo, le daría el mismo valor.
    }

    public void cambiarEdad(int ed)
    {
        edad=ed;
    }

    public void asignarantig(int ant)
    {
        antiguedad=ant;
    }
}

```

### Solución (cont.)

```

    }

    /*El método siguiente es un método con parámetros, con lo cual, cuando
    llame a este método le tiene que enviar, como argumentos, una cadena y un
    número entero, que será el nombre del alumno y la edad. Dichos datos los
    recogerán las variables nomenv y edadenv para guardarlos en las variables
    del objeto.*/

    public void InicDatosEnviados(String nomenv, int edadenv )
    {
        nombre=nomenv;
        edad=edadenv;
    }

    public void visualizarDatos( )
    {
        System.out.println("\tNombre:\t\t" + nombre);
        System.out.println("\tEdad:\t\t" + edad);
        System.out.println("\n\t" + antigüedad);
    }
}

```

### Ejercicio

Escribe un programa que pida los datos de cinco alumnos. Los datos que interesan de dichos alumnos son: nombre del ciclo, nombre del alumno y la cantidad de horas en las que se ha matriculado dicho alumno.

El programa pedirá el nombre de cada alumno pero, el nombre del ciclo y las horas en la que está matriculado, son los mismos datos para los cinco, ya que están matriculado en el mismo ciclo y a la misma cantidad de horas, por ello, se pedirá estos datos en el main y se enviarán los datos, a los cinco alumnos, para que cada uno de ellos tengan esta información.

Posteriormente, se visualizará los datos de los cinco alumnos.

Seguidamente, tres de los alumnos cambian de ciclo. Se pasan los tres al mismo ciclo. El nombre del nuevo ciclo y el número de horas (en los que se matriculan los alumnos) son los mismos datos para los tres, por ello, se pedirán en el main (con el fin de pedir dichos datos una sola vez) para enviárselo a cada uno de los alumnos.

Posteriormente, se visualizará los datos de los cinco alumnos.

## Solución

```
package proyalumciclo;
import java.io.*;

public class ProyAlumCiclo
{
    public static void main(String[] args) throws IOException
    {
        String nomCiclo, horasCad;
        int horasCiclo;
        InputStreamReader via=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(via);

        Alumno alum1,alum2,alum3,alum4,alum5;
        alum1=new Alumno();
        alum2=new Alumno();
        alum3=new Alumno();
        alum4=new Alumno();
        alum5=new Alumno();

        alum1.pedirNombre();
        alum2.pedirNombre();
        alum3.pedirNombre();
        alum4.pedirNombre();
        alum5.pedirNombre();
        System.out.print("Indique el nombre del ciclo donde están matriculado los
                        cinco alumnos: ");
        nomCiclo=teclado.readLine();

        System.out.print("¿En cuantas horas se van a matricular los alumnos?
                        ");
        horasCad=teclado.readLine();
        horasCiclo=Integer.parseInt(horasCad);

        alum1.inicNomCicHorasCic(nomCiclo, horasCiclo);
        alum2.inicNomCicHorasCic(nomCiclo, horasCiclo);
        alum3.inicNomCicHorasCic(nomCiclo, horasCiclo);
        alum4.inicNomCicHorasCic(nomCiclo, horasCiclo);
        alum5.inicNomCicHorasCic(nomCiclo, horasCiclo);

        visualTodosAlumnos(alum1, alum2, alum3, alum4, alum5);

        System.out.print("Indique el nombre del ciclo donde se van tres alumnos:
                        ");
        nomCiclo=teclado.readLine();

        System.out.print("¿En cuantas horas se van a matricular los alumnos?
                        ");
        horasCad=teclado.readLine();
        horasCiclo=Integer.parseInt(horasCad);

        alum1.inicNomCicHorasCic(nomCiclo, horasCiclo);
        alum3.inicNomCicHorasCic(nomCiclo, horasCiclo);
    }
}
```

## Solución (cont.)

```

        alum5.inicNomCicHorasCic(nomCiclo, horasCiclo);

        visualTodosAlumnos(alum1, alum2, alum3, alum4, alum5);
    }

    void visualTodosAlumnos(Alumno al1, Alumno al2, Alumno al3, Alumno al4,
                            Alumno al5)
    {
        al1.visualDatosAlumno();
        al2.visualDatosAlumno();
        al3.visualDatosAlumno();
        al4.visualDatosAlumno();
        al5.visualDatosAlumno();
    }
}

package proyalumciclo;

import java.io.*;
public class Alumno
{
    String nombre;
    String nomCiclo;
    int horasCiclo;

    void pedirNombre() throws IOException
    {
        InputStreamReader via=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(via);
        System.out.print("Inserte el nombre del alumno: ");
        nombre=teclado.readLine();
    }

    void visualDatosAlumno()
    {
        System.out.println("\n\tEl nombre es: " + nombre);
        System.out.println("\n\tEstá matriculado en el ciclo: " + nomCiclo);
        System.out.println("\n\tDicho ciclo tiene " + horasCiclo + " horas");
    }

    int devolverHorasAlumno()
    {
        return horasCiclo;
    }

    String devolverNombreAlumno( )
    {
        return nombre;
    }

    void inicNomCicHorasCic(String nom, int horasCiclo )
    {

```

### Solución (cont.)

```
        nomCiclo=nom;
        this.horasCiclo=horasCiclo;
    }
}
```

## 4.5. CONSTRUCTORES

### Ejemplo

#### CÓDIGO DE LA CLASE "EMPLEADO"

```
package ejercicio1;
import java.io.*;

public class Empleado
{
    int antiguedad;
    int edad;
    String nombre;

    public Empleado() throws IOException{
        String cadedad;
        InputStreamReader flujo=new InputStreamReader(System.in);
        BufferedReader teclado=new BufferedReader(flujo);

        System.out.print("\n\tIndique el nombre del empleado: ");
        nombre=teclado.readLine();

        System.out.print("\n\tIndique la edad de " + nombre + ":\t");
        cadedad=teclado.readLine();
        edad=Integer.parseInt(cadedad);

        antiguedad=0;//Si no lo pongo, le daría el mismo valor.
    }

    public Empleado(String nomenv, int edadenv )
    {
        nombre=nomenv;
        edad=edadenv;
    }

    public void visualizardatos( )
    {
        System.out.println("\tNombre:\t\t" + nombre);
        System.out.println("\tEdad:\t\t" + edad);
        System.out.println("\tAntiguedad:\t" + antiguedad);
    }
}
```



## Ejemplo (cont.)

### CÓDIGO DE LA CLASE "PRINCIPAL"

```
package ejercicio1;
import java.io.*;

public class Principal {
    public Principal() { }

    public static void main(String args[]) throws IOException
    {
        System.out.println("\nVamos a declarar el empleado 1");
        Empleado empl1=new Empleado();

        System.out.println("\nVamos a declarar el empleado 2");
        Empleado empl2=new Empleado();

        System.out.println("\nVamos a declarar el empleado 3");
        Empleado empl3=new Empleado("David Abad", 30);

        System.out.println("\nVamos a declarar el empleado 4");
        Empleado empl4=new Empleado("Paula Abad", 21);

        System.out.println("\nVamos a visualizar los datos del empleado 1:\n");
        empl1.visualizardatos();

        System.out.println("\nVamos a visualizar los datos del empleado 2:\n");
        empl2.visualizardatos();

        System.out.println("\nVamos a visualizar los datos del empleado 3:\n");
        empl3.visualizardatos();

        System.out.println("\nVamos a visualizar los datos del empleado 4:\n");
        empl4.visualizardatos();
    } // cierra main
} // cierra clase
```

## Ejercicio

En el ejemplo anterior, ¿podría haber instanciado un objeto de tipo Empleado de esta forma?

```
Empleado empl5=new Empleado("Miguel Abad");
```

¿Porqué?

## Solución

No admite ese new debido a que no hay ninguna función constructora que reciba un sólo dato de tipo string.

## ■ UNIDAD 6

### Control y manejo de excepciones

Para poder poner en práctica la utilización de las clases y los objetos vamos a plantear el siguiente ejercicio:

Escribe un programa que pida los datos de dos empresas. De cada empresa nos interesa: número de empresa asignado, año de creación de la empresa, dinero invertido, información de los empleados (de dicha empresa) y la cuota asignada a los hijos de los empleados. Dicha cuota es una cantidad de dinero que da las empresas a uno de los hijos de los empleados; dicha cantidad es la misma para todas las empresas.

Suponemos que una empresa tiene, al menos, un empleado y, como mucho, tres empleados. Para saber cuantos empleados tiene una empresa, se le hará dicha pregunta al usuario del programa. Él nos tendrá que dar un número que estará entre 1 y 3.

La información que nos interesa de dicho empleado es: nombre, sueldo y la información de uno de sus hijos (si es que el empleado tiene algún hijo).

La información que nos interesa de un hijo es: nombre, si estudia o trabaja y año de nacimiento.

La petición de datos de los objetos se hará a través de funciones constructoras.

Controla la visibilidad de los diferentes elementos de una clase.

Haz todas las validaciones que creas conveniente con respecto a la entrada de datos por el teclado. Cuantas más hagas, mejor.

Una vez que el programa tenga toda esta información, se visualizará un menú con las siguientes opciones con el fin de que el programa gestione la opción elegida:

1. Visualizar los datos exclusivos de cada empresa.  
En este caso, se visualizará, para cada una de las dos empresas: el número de empresa asignado, año de creación de la empresa y el dinero invertido.
2. Visualizar el nombre y el sueldo de todos aquellos empleados que pertenezcan a una empresa que haya invertido una cantidad inferior a 300.000 €.
3. Visualizar los datos de cada una de las empresas y la información de todos sus empleados siempre que, dichos empleados, tengan algún hijo.
4. Visualizar los datos de cada una de las empresas y la media del sueldo que paga a sus empleados.
5. Cambiar la cuota asignada a los hijos de los empleados. Este cambio debe de afectar a todas las empresas.
6. Cambiar la cantidad invertida por una empresa.  
Para ello, se dará el número de la empresa (a la que queremos modificar dicha cantidad) y la cantidad nueva invertida.  
Puede suceder que el número de empresa que indique el usuario no corresponda con el que tienen asignados las dos empresas. En este caso, se visualizará el mensaje correspondiente y, se volverá a visualizar el menú.
7. Visualizar la cuota que se da a los hijos de los empleados.
8. Salir.

## Solución

### CLASE PRINCIPAL

```
package utilizacionclases;
import java.io.*;

public class PoligonoIndustrial
{
    Empresa empr1, empr2;
    public static void main(String args[]) throws IOException
    {
        PoligonoIndustrial polig;
        polig=new PoligonoIndustrial();
        polig.darDatos();
    }

    darDatos()
    {
        int opc;
        /*Este método ya no puede ser estático ya que uso empr1 y empr2 que
        son datos de la clase.

        Son datos de la clase para que se puedan usar en cualquier método de la
        misma.*/

        Empresa.pedirCuotaHijos();
        System.out.println("Vamos a crear la primera empresa.");
        empr1=new Empresa();

        System.out.println("Vamos a crear la segunda empresa.");
        empr2=new Empresa();

        opc=visualizarMenu();/*Como a este método ya no lo llamas desde el
                               main, no hace falta que sea estático. Pero, lo dejo
                               así.*/

        while(opc!=8)
        {
            switch(opc)
            {
                case(1):
                    visualTodasEmp();
                    break;
                case(2):
                    visualTodosEmpl();
                    break;
                case(3):
                    visualEmpleHijos();
                    break;
                case(4):
                    visualDatosEmpresaMediaPagoEmpleados();
                    break;
            }
        }
    }
}
```

## Solución (cont.)

```

        case(5):
            cambiarCuotaHijos();
            break;
        case(6):
            cambiarCuotaInvertidaEmpresa();
            break;
        case(7):
            Empresa.visulCuotaHijos();
            break;
    }
    opc=visualizarMenu();
}
System.out.println("Final del programa");
} // cierra main

void cambiarCuotaInvertidaEmpresa()
{
    int numempre;
    char entrada='n';

    do
    {
        System.out.print("\n\nIndique el número de la empresa del que quiere
                           cambiar la cuota invertida: ");
        try
        {
            numempre = Leer.pedirEntero();
            entrada='s';
            if(empr1.getNumEmpre()==numempre)
            {
                System.out.print("Indique la nueva cuota invertida para la
                                   empresa con número " + numempre + ": ");
                empr1.pedirDineroInvert();
            }
            else
            {
                if(empr2.getNumEmpre()==numempre)
                {
                    System.out.print("Indique la nueva cuota invertida para la
                                       empresa con número " + numempre + ": ");
                    empr2.pedirDineroInvert();
                }
                else
                    System.out.println("No hay ninguna empresa con el numero "
                                         + numempre);
            }
        }
        catch (NumberFormatException e)
        {
            System.out.println("\n\n\t¡¡Por favor, inserte un número.!!");
        }
    }
}

```

## Solución (cont.)

```

    }
    } while(entrada!='n');
}

void cambiarCuotaHijos()
{
    System.out.println("Vamos a cambiar la cuota que las empresas asignan
                        a los hijos de los empleados: ");
    Empresa.pedirCuotaHijos();
    Utilidades.pausa();
    Utilidades.borrarPantalla();
}

void visualDatosEmpresaMediaPagoEmpleados()
{
    float media;

    empr1.visualEmpresa();

    media=empr1.mediaPagoEmpleados();
    System.out.println("La media que paga la empresa 1 a sus empleados
                        es " + media);
    empr2.visualEmpresa();

    media=empr2.mediaPagoEmpleados();
    System.out.println("La media que paga la empresa 2 a sus empleados es
                        " + media);
}

void visualEmpleHijos()
{
    System.out.println("Los datos de los empleados de la empresa 1: ");
    this.empr1.visualEmpresa();

    System.out.println("\n\t\t\t\tLos datos de los empleados que tienen hijos
                        son: ");
    empr1.visualDatosEmpleadosConHijos();

    System.out.println("Los datos de los empleados de la empresa 2: ");
    this.empr2.visualEmpresa();

    System.out.println("\n\t\t\t\tLos datos de los empleados que tienen hijos
                        son: ");
    empr2.visualDatosEmpleadosConHijos();
}

void visualTodosEmpl( )
{
    if(this.empr1.getDineroInvert()<300000)
        empr1.visualDatosEmpleados();
}

```

## Solución (cont.)

```

        if(this.empr2.getDineroInvert()<300000)
            empr2.visualDatosEmpleados();
    }

    void visualTodasEmp( )
    {
        empr1.visualEmpresa();
        empr2.visualEmpresa();
    }

    /*Declaro el método visualizarMenu static para poder llamarlo desde el main
    sin tener que declara un objeto de la clase principal. Este método puede ser
    estático ya que no usa datos de la clase.*/

    static int visualizarMenu( )
    {
        int opc=0;
        char correcto;
        correcto='n';
        do
        {
            System.out.println("Dado el siguiente menú: ");
            System.out.println("\t1.-\tVisualizar los datos (exclusivos) de cada
                                empresa.");
            System.out.println("\t2.-\tVisualizar los datos (exclusivos) de
                                los empleados según el dinero invertido en la
                                empresa.");
            System.out.println("\t3.-\tVisualizar los datos de los empleados e hijos,
                                siempre que tenga un hijo.");
            System.out.println("\t4.-\tVisualizar los datos (exclusivos) de la
                                empresa y la media que paga a sus empleados.");
            System.out.println("\t5.-\tCambiar la cuota de los hijos.");
            System.out.println("\t6.-\tCambiar la cuota invertida por una
                                empresa.");
            System.out.println("\t7.-\tVisualizar la cuota de los hijos");
            System.out.println("\t8.-\tSalir.");
            try
            {
                System.out.print("Elija una opción: ");
                opc = Leer.pedirEntero();
                if(opc<1 || opc>8)
                    System.out.println("\n\t\tPor favor, inserte un número entre el
                                        1 y el 8.");
                else
                    correcto='s';
            }
            catch(NumberFormatException e)
            {
                System.out.println("\n\t\tPor favor, inserte números y no
                                    letras.");
            }
        }
    }

```

## Solución (cont.)

```
    }
    }while(correcto=='n');
    return opc;
}
} // cierra clase
```

### CLASE UTILIDADES

```
package utilizacionclases;
import java.io.*;
public class Utilidades
{
    static void pausa( )
    {
        System.out.print("Pulse una tecla para volver a visualizar el menú: ");
        Leer.pedirCadena();
    }

    static void borrarPantalla( )
    {
        for(int pos=0;pos<25;pos++)
            System.out.println();
    }
}
```

### CLASE LEER

```
package utilizacionclases;
import java.io.*;
public class Leer
{
    static String pedirCadena( )
    {
        String cadLeida="";
        InputStreamReader flujo = new InputStreamReader(System.in);
        BufferedReader teclado = new BufferedReader(flujo);
        try
        {
            cadLeida=teclado.readLine();
        }
        catch(IOException e)
        {
            System.out.println("Nos salimos del programa ya que no se puede
                                realizar una entrada de datos por el teclado");
        }
        return cadLeida;
    }

    static int pedirEntero ( )throws NumberFormatException
    {
```

## Solución (cont.)

```

        String cadLeida;
        int intLeido;
        cadLeida=pedirCadena();
        intLeido=Integer.parseInt(cadLeida);
        return intLeido;
    }

    static float pedirFloat ( )throws NumberFormatException
    {
        String cadLeida;
        float floatLeido;
        cadLeida=pedirCadena();
        floatLeido=Float.parseFloat(cadLeida);
        return floatLeido;
    }
}

```

## CLASE HIJO

```

package utilizacionclases;

public class Hijo
{
    String nombre;
    String estudiaTrabaja;
    int anioNac;

    public Hijo(
    {
        pedirNombre();
        pedirSiEstTrab();
        pedirAnioNac();
    }

    void visualDatosHijo()
    {
        System.out.println("\n\t\tLos datos del hijo del empleado es ");
        visualNombre();
        visualEstudiaTrabaja();
        visualAnioNac();
    }

    public void visualNombre( )
    {
        System.out.println("\n\t\tNOMBRE:  " + nombre);
    }

    void visualEstudiaTrabaja()
    {

```



## Solución (cont.)

```

        if(estudiaTrabaja.equals("e"))
            System.out.println("\t\t\tEN ESTOS MOMENTOS, ESTA PERSONA,
                                ESTUDIA");
        else
            System.out.println("\t\t\tEN ESTOS MOMENTOS, ESTA PERSONA,
                                TRABAJA");
    }

    void visualAnioNac( )
    {
        System.out.println("\t\t\tSU AÑO DE NACIMIENTO ES: " + anioNac);
    }

    void pedirNombre( )
    {
        System.out.println("\t\t\tIndique el nombre de uno de los hijos del
                                empleado: ");
        nombre=Leer.pedirCadena();
    }

    void pedirSiEstTrab()
    {
        System.out.println("\t\t\t¿Estudia o trabaja (e ó t)? : ");
        estudiaTrabaja=Leer.pedirCadena();
        while(!estudiaTrabaja.equals("e") && !estudiaTrabaja.equals("t"))
        {
            System.out.println("\n\t\t\tNo ha contestado a la pregunta.");
            System.out.println("\t\t\t¿Estudia o trabaja (e ó t)? : ");
            estudiaTrabaja=Leer.pedirCadena();
        }
    }

    void pedirAnioNac( )
    {
        char entrada='n';
        do
        {
            try
            {
                System.out.println("\tIndique el año de nacimiento del hijo: ");
                this.anioNac= Leer.pedirEntero();
                entrada='s';
            }
            catch (NumberFormatException e)
            {
                System.out.println("\n\t\tPor favor, vuelva a indicar el año de
                                    nacimiento, ya que no ha insertado un número");
            }
        }while(entrada!='n');
    }
}

```

## Solución (cont.)

### CLASE EMPRESA

```
package utilizacionclases;

public class Empresa
{
    int numempleados; /*Lo pongo como dato de la empresa para que a la hora
                        de visualizar los datos de los empleados sepa cuantos
                        hay.*/
    private int numEmpre;
    private int anioInicio;
    private float dineroInvert;
    private static float cuota;
    Empleado2 empl1, empl2, empl3;

    public float getDineroInvert()
    {
        return dineroInvert;
    }

    public int getNumEmpre()
    {
        return numEmpre;
    }

    public Empresa()
    {
        pedirNumEmpre();
        pedirAnioInicio();
        pedirDineroInvert();
        pedirDatosEmpleados();
    }

    float mediaPagoEmpleados( )
    {
        float total, media;
        if(this.numempleados==3)
        {
            System.out.println("\n\nEsta empresa tiene " + this.numempleados
                               + " empleados y el sueldo de ellos es " + empl1.
                               getSueldo() + "," + empl2.getSueldo() + " y " + empl2.
                               getSueldo());
            total=empl1.getSueldo()+empl2.getSueldo()+empl3.getSueldo();
        }
        else
        {
            if(this.numempleados==2)
            {
                System.out.println("\n\nEsta empresa tiene " + this.numempleados
                                   + " empleados y el sueldo de ellos es " + empl1.
                                   getSueldo() + " y " + empl2.getSueldo());
                total=empl1.getSueldo()+empl2.getSueldo();
            }
        }
    }
}
```

## Solución (cont.)

```

    }
    else
    {
        System.out.println("\n\nEsta empresa tiene " + this.numempleados
                            + " empleados y el sueldo de él es " + empl.
                            getSueldo());
        total = empl.getSueldo();
    }

    media=total/this.numempleados;
    return media;
}

void visualEmpresa( )
{
    System.out.println("\nLos datos de esta empresa son: ");
    visualNumeroEmpresa();
    visualAnioInicio();
    visualDineroInvert();
    // visualDatosEmpleados();
    // visualCuotaHijos();
}

/*Hago algunos métodos privados para que no se puedan ver desde fuera, al
poner el nombre del objeto.*/
private void visualNumeroEmpresa()
{
    System.out.println("\n\tNÚMERO DE LA EMPRESA: " + numEmpre);
}

private void visualAnioInicio()
{
    System.out.println("\n\tAÑO DE INICIO DE LA EMPRESA: " + anioInicio);
}

private void visualDineroInvert()
{
    System.out.println("\n\tDINERO INVERTIDO EN LA EMPRESA: " + this.
                        dineroInvert);
}

void visualDatosEmpleados()
{
    if (this.numempleados==1)
        empl.visualDatosEmpleado();
    else
        if(numempleados==2)
        {
            System.out.println("La empresa con numero " + this.numEmpre + "
                                tiene 2 empleados. Sus datos son:");
            empl.visualDatosEmpleado();
        }
    }

```

## Solución (cont.)

```

        emp2.visualDatosEmpleado();
    }
    else
    {
        emp1.visualDatosEmpleado();
        emp2.visualDatosEmpleado();
        emp3.visualDatosEmpleado();
    }
}

void visualDatosEmpleadosConHijos()
{
    if (this.numempleados==1)
    {
        if(emp1.getTieneHijos().equals("s"))
            emp1.visualDatosEmpleado();
    }
    else
        if(numempleados==2)
        {
            if(emp1.getTieneHijos().equals("s"))
                emp1.visualDatosEmpleado();

            if(emp2.getTieneHijos().equals("s"))
                emp2.visualDatosEmpleado();
        }
    else
    {
        System.out.println("Esta empresa tiene 3 empleados");
        if(emp1.getTieneHijos().equals("s"))
            emp1.visualDatosEmpleado();

        if(emp2.getTieneHijos().equals("s"))
            emp2.visualDatosEmpleado();

        if(emp3.getTieneHijos().equals("s"))
            emp3.visualDatosEmpleado();
    }
}

private void visualCuotaHijos()
{
    System.out.println("\n\tCUOTA ENTREGADA A LOS HIJOS: " + this.
        cuota);
}

public void pedirNumEmpre( )
{
    char entrada='n';
    do
    {

```

## Solución (cont.)

```

        try
        {
            System.out.println("\tIndique el número de la empresa: ");
            numEmpre = Leer.pedirEntero();
            entrada='s';
        }
        catch (NumberFormatException e)
        {
            System.out.println("\n\t\tPor favor, vuelva a indicar el número de
                                empleado ya que no ha insertado un número");
        }
    }while(entrada!='n');
}

public void pedirAnioInicio( )
{
    char entrada='n';
    do
    {
        try
        {
            System.out.println("\tIndique el año de inicio de la empresa: ");
            this.anioInicio = Leer.pedirEntero();
            entrada='s';
        }
        catch (NumberFormatException e)
        {
            System.out.println("\n\t\tPor favor, vuelva a indicar el año de
                                inicio, ya que no ha insertado un número");
        }
    }while(entrada!='n');
}

public void pedirDineroInvert( )
{
    char entrada='n';
    do
    {
        try
        {
            System.out.println("\tIndique el dinero invertido en la empresa: ");
            this.dineroInvert = Leer.pedirFloat();
            entrada='s';
        }
        catch (NumberFormatException e)
        {
            System.out.println("\n\t\tPor favor, vuelva a indicar el dinero
                                invertido, ya que no ha insertado un número");
        }
    }
}

```

### Solución (cont.)

```

        }while(entrada=='n');
    }

    public void pedirDatosEmpleados( )
    {
        char entrada='n';
        /*Yo sé que va a tener seguro un valor, ya que lo tengo controlado,
        pero como él no estudia el código sino ve que se plantea que se pueda
        producir un error, y con lo cual
        que la variable no tenga datos, por eso le engaño asignándole un valor.*/
        do
        {
            try
            {
                System.out.println("\tIndique el numero de empleados que tiene la
                                   empresa: ");
                numempleados = Leer.pedirEntero();
                entrada='s';
            }
            catch (NumberFormatException e)
            {
                System.out.println("\n\t\tPor favor, vuelva a indicar el número de
                                   empleados, ya que no ha insertado un número");
            }
        }while(entrada=='n');

        if(numempleados==1)
        {
            System.out.println("\t\tVamos a pedir los datos del primer empleado:
                               ");
            this.empl = new Empleado2();
        }
        else
            if(numempleados==2)
            {
                System.out.println("\t\tVamos a pedir los datos del primer
                                   empleado: ");
                this.empl=new Empleado2();
                System.out.println("\t\tVamos a pedir los datos del segundo
                                   empleado: ");
                this.emp2=new Empleado2();
            }
            else
            {
                System.out.println("\t\tVamos a pedir los datos del primer
                                   empleado: ");
                this.empl=new Empleado2();

                System.out.println("\t\tVamos a pedir los datos del segundo
                                   empleado: ");
                this.emp2=new Empleado2();
            }
        }
    }

```

## Solución (cont.)

```

        System.out.println("\t\tVamos a pedir los datos del tercer empleado:
                               ");
        this.emp3=new Empleado2();
    }
}

static void visulCuotaHijos( )
{
    System.out.println("\n\tLa cuota que se asigna a los hijos de los
                        empleados es " + cuota);
    Utilidades.pausa();
}

static void pedirCuotaHijos( )
{
    char entrada='n';
    do
    {
        try
        {
            System.out.println("\tIndique la cuota que da las empresas, a los
                                hijos de los empleados: ");
            cuota= Leer.pedirFloat();
            entrada='s';
        }
        catch (NumberFormatException e)
        {
            System.out.println("\n\t\tPor favor, vuelva a indicar la cuota, ya
                                que no ha insertado un número");
        }
    }while(entrada=='n');
}
}

```

## CLASE EMPLEADO

```

package utilizacionclases;
public class Empleado2
{
    String nombre;
    float sueldo;
    String tieneHijos;
    Hijo hijoEmp;

    public String getTieneHijos(
    {
        return tieneHijos;
    }
}

```

## Solución (cont.)

```

public float getSueldo()
{
    return sueldo;
}

public Empleado2()
{
    pedirNombre();
    pedirSueldo();
    pedirInformaHijo();
}

void visualDatosEmpleado( )
{
    visualNombre();
    visualSueldo();
    if(tieneHijos.equals("s"))
        visualInforUnHijo();
    else
        System.out.println("\t\tEste empleado no tiene hijos");
}

void visualNombre( )
{
    System.out.println("\n\n\t\tEl nombre del empleado es " + nombre);
}

void visualSueldo( )
{
    System.out.println("\t\tEl sueldo del empleado es " + sueldo);
}

void visualInforUnHijo( )
{
    hijoEmp.visualDatosHijo();
}

void pedirNombre( )
{
    System.out.println("\t\tIndique el nombre del empleado: ");
    nombre=Leer.pedirCadena();
}

void pedirSueldo()
{
    char entrada='n';
    do
    {
        try
        {
            System.out.println("\t\tIndique el sueldo del empleado: ");

```



### Solución (cont.)

```

        sueldo = Leer.pedirFloat();
        entrada='s';
    }
    catch (NumberFormatException e
    {
        System.out.println("\n\t\tPor favor, vuelva a indicar el sueldo del
                                empleado, ya que no ha insertado un número");
    }
    }while(entrada!='n');
}

void pedirInformaHijo()
{
    System.out.println("\t\t¿Este empleado tiene algún hijo(s ó n)? : ");
    tieneHijos=Leer.pedirCadena();
    while(!tieneHijos.equals("s") && !tieneHijos.equals("n"))
    {
        System.out.println("\n\t\tNo ha contestado a la pregunta.");
        System.out.println("\t\t¿Este empleado tiene algún hijo(s ó n)? : ");
        tieneHijos=Leer.pedirCadena();
    }

    if(tieneHijos.equals("s"))
    {
        hijoEmp=new Hijo();
    }
    }
}

```