

PROGRAMACIÓN I

FECHA: 30-enero-2002

1. (0.5 puntos) Escribir lo que significan las siguientes siglas:
 - AWT
 - GUI
 - HTML
 - <HR>
 - <H1>
 - IDE
 - JDK
 - JVM
 - RAM
 - ROM
2. (1.5 puntos) Obtener la representación interna en el ordenador (en binario) de los siguientes números:
 - a. Del entero en base decimal -142316 . ¿Cuántos bytes ocupa?
 - b. Del int en base hexadecimal $85A22234$. ¿Cómo se puede saber si es positivo o negativo echando un vistazo rápido (1 segundo)?

3. (1.5 puntos) Responder a las siguientes preguntas:
- En un proceso en DOS que trata con INPUTs/OUTPUTs, ¿cuáles son los nombres estándar de los manejadores (STREAMERS) de ficheros secuenciales? Para ello dibujar un gráfico, que incluya esos nombres y también las flechas que indiquen las direcciones (INPUT, OUTPUT, I/O) de dichos streamers.
 - Relacionándolo con el apartado a), ¿qué hacen los siguientes comandos?
 - C:>COPY FICH1.DAT CON:
 - C:>TYPE FICH1.DAT
 - C:>MORE < FICH1.DAT
 - C:>TYPE FICH1.DAT | MORE
 - ¿Qué diferencias hay entre cada par de comandos de MS-DOS?
 - C:>TYPE autoexec.bat > NUL y
 - C:>TYPE autoexec.bat > NULL
 - C:>TYPE autoexec.bat > PRN y
 - C:>type autoexec.bat > LPT1
 - C:>TYPE autoexec.bat > COM1: y
 - C:>TYPE autoexec.bat > COM2

4. (0.5 puntos) Escribir en lenguaje Java la sentencia de asignación equivalente a la siguiente ecuación, incluyendo sólo los paréntesis estrictamente necesarios:

$$y = \frac{-b + (b^2 - 4ac)^{1/n}}{2a}$$

$$y = (-b + \text{Math.pow}(b * b - 4 * a * c, 1 / n)) / (2 * a)$$

5. (1 punto) Enunciar las características formales que tienen los constructores con respecto a otros métodos miembro de una clase instanciable, así como escribir si un constructor se trata de un método de instancia o de clase y también si es un método void o non-void.

a. ¿En qué caso es necesario incluir la palabra reservada 'this' dentro de un .metodoMiembro () para referirse a un .datoMiembro de la clase?

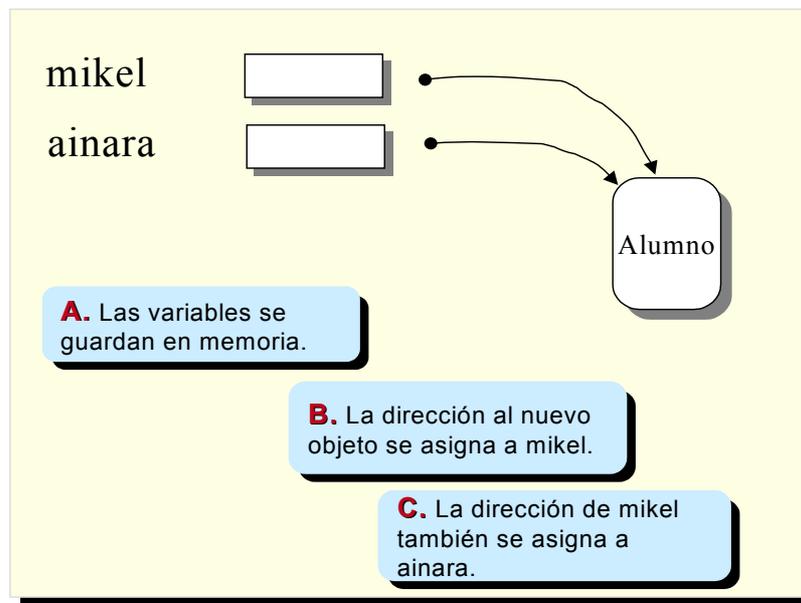
b. ¿Cuál(es) de los siguientes constructores no es(son) válido(s)? Razonarlo.

```
public int ClaseA (int uno) {  
    ...  
}  
public ClaseB (int uno) {  
    ...  
}  
public ClaseC (int uno, int dos) {  
    ...  
}
```

El constructor no válido es el correspondiente al constructor de objetos de la Clase A, pues es un método que no devuelve nada (es void y no puede ser int).

6. (0.5 puntos) Escribir las sentencias necesarias en Java que muestren un ejemplo donde dos direcciones de memoria se refieran al mismo objeto de la clase Alumno. Asimismo dibujar el diagrama de estado de la memoria después de cada una de las instrucciones anteriores.

Alumno mikel, ainara; ----- A
 mikel = new Alumno (); ----- B
 ainara = mikel; ----- C



7. (1 punto) Suponiendo que se tienen las siguientes declaraciones:

```
int m, n, i = 3, j = 4, k = 5;
float v, w, x = 34.5f, y = 12.25f;
```

Determinar el valor asignado a la variable en cada una de las siguientes sentencias de asignación o explicar por qué no son sentencias válidas.

- `m = int (x * i / 2);`
 - `v = x / i / j;`
 - `n = Math.pow (i, Math.pow (j, k));`
 - `w = Math.sqrt(i * i - 4 * j * k);`
 - `i = Math.ceil(y) % k;`
- La sentencia no es válida, pues la palabra reservada `int` debería ir entre paréntesis. Se trata de una conversión de tipo explícita.
 - La regla de precedencia es dividir primero $34.5 / 3$ y luego $11.5 / 4$, siendo el resultado final de 2.875.
 - La expresión equivale a $i^{(j^k)}$, cuyo resultado i^{1024} origina un error de overflow.
 - La sentencia no es válida, pues el radicando de la raíz es negativo.
 - El resultado es 3, que es el resto de la división entera: $13 / 5$.

8. (0.5 puntos) ¿Las dos sentencias `if` (denominadas /* A */ y /* B */) obtienen el mismo resultado? Justificar la respuesta tanto si es positiva como negativa.

```
int num = 45;
-----
/* A */
if (num == 45)
{
    System.out.println ("¡Kaixo!");
}
else
    System.out.println ("¿Qué tal el examen de hoy?");
    System.out.println ("¡Gero arte!");
-----
/* B */
if (num != 45)
{
    System.out.println ("¿Qué tal el examen de hoy?");
    System.out.println ("¡Gero arte!");
}
else
{
    System.out.println ("¡Kaixo!");
}
-----
```

No se obtiene el mismo resultado.

Teniendo en cuenta que el valor de `num` es 45,

Opción A: La salida en pantalla es ¡Kaixo!
 ¡Gero arte!

Opción B: La salida en pantalla es ¡Kaixo!

9. (0.5 puntos) ¿El resultado de la sentencia `switch` es siempre `Ondo` ? En caso afirmativo, explicar cómo funciona dibujando el diagrama de flujo correspondiente a dicha instrucción. En caso negativo, explicar cómo se resolvería dibujando el diagrama de flujo correspondiente a dicha solución. En ambos casos, la(s) instrucción(es) dentro de cada cuadro debe(n) estar claramente especificada(s).

```
int num = 45;

switch ( num )
{
    case 35: System.out.println ("Gaizki");
    case 45: System.out.println ("Ondo");
    case 55: System.out.println ("Bikain");
}

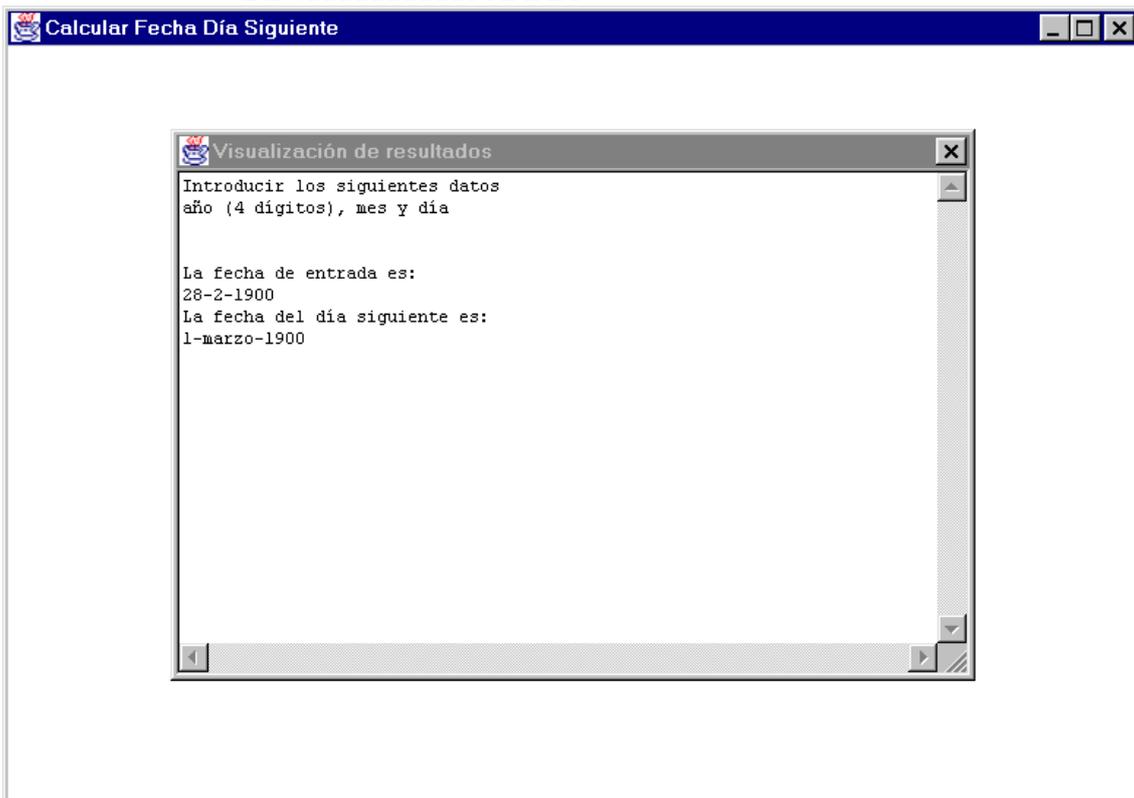
```


- Método *empezar*

```
public void empezar ()  
{  
    ventanaPrincipal.show ();  
    cuadroSalida.show ();  
    describirPrograma ();  
    obtenerEntrada ();  
    mostrarFechaEntrada ();  
    calcularFechaDiaSiguiete ();  
    mostrarFechaSiguiete ();  
}
```

- Privados

- Procedimiento *describirPrograma*
- Procedimiento *obtenerEntrada*
- Procedimiento *mostrarFechaEntrada*
 - Visualizar la fecha de entrada según el siguiente formato:
DD-MM-AAAA
- Procedimiento *mostrarFechaSiguiete*
 - Visualizar la fecha de salida según el siguiente formato:
DD-xxxxxxxxxx-AAAA



- Procedimiento *calcularFechaDiaSiguiete*
 - En general, en el 90% de los casos hay que aumentar en 1 el día.
 - Pero se deben considerar todos los casos especiales dependiendo del número del mes que se haya introducido. A continuación, se exponen la mayoría de los casos que hay que tener en cuenta:
 - Si el día resultante es 32 y el mes es marzo, entonces hay que modificar el día a 1 y sumar 1 al mes. Si se trata del mes de diciembre, entonces hay que modificar el día a 1, el mes a 1 y sumar 1 al año.

- Si el día resultante es 31 y el mes es abril, entonces hay que cambiar el día a 1 y sumar 1 al mes.
- Si el día resultante es 29, el mes es febrero y el año es no bisiesto, entonces hay que modificar el día a 1 y el mes a 3. Esta misma modificación se realiza cuando el día resultante es 30, el mes es febrero y el año es bisiesto. Para saber si un año es o no bisiesto, se debe llamar al método *añoBisiesto*. Esta función devuelve un valor lógico: true o false, una vez que el argumento, en este caso el año del objeto, se pasa por valor al parámetro formal de la función añoBisiesto.
- El resto de los casos debes pensarlos tú.
- Función *añoBisiesto*, que devuelve un valor lógico: true o false.
 - Su parámetro formal es un año cualquiera.
 - Un año es bisiesto si es divisible:
 - por 400 o
 - por 4 y no divisible por 100.

Se pide:

- a. Completar la codificación de la clase *FechaI*, incluyendo los métodos ya hechos (en el constructor añadir los modificadores que falta(n): ??????????????).
- b. Escribir los pasos necesarios para ejecutar y obtener la fecha del día siguiente, suponiendo que:
 - las clases FechaMainI.java y FechaI.java residen en C:\Java\Aplicaciones,
 - las clases de javabook residen en C:\Java, y
 - el compilador y la máquina virtual de Java están en C:\JDK1.3.1\BIN.

a. La solución completa del programa es:

```

/* Clase FechaMainI */

import javabook.*;

public class FechaMainI
{
    public static void main (String args[])
    {
        //declarar y crear un objeto de la clase Fecha

        FechaI fechaTecleada;
        fechaTecleada = new FechaI ();
        fechaTecleada.empezar();
    }
}

/* Clase FechaI */

import javabook.*;

public class FechaI
{
    //datos miembro privados

```

```

private MainWindow ventanaPrincipal;
private InputBox cuadroEntrada;
private OutputBox cuadroSalida;

private int año, mes, dia;

//constructor

public FechaI ()
{
    ventanaPrincipal = new MainWindow(
        "Calcular Fecha Día Siguiente");
    cuadroEntrada = new InputBox(ventanaPrincipal,
        "Entrada de datos");
    cuadroSalida = new OutputBox(ventanaPrincipal,
        "Visualización de resultados");
}

//métodos públicos

public void empezar ()
{
    ventanaPrincipal.show ();
    cuadroSalida.show ();
    describirPrograma ();
    obtenerEntrada ();
    mostrarFechaEntrada ();
    calcularFechaDiaSiguiente ();
    mostrarFechaSiguiente ();
}

//métodos privados

private void calcularFechaDiaSiguiente ()
{
    dia = dia + 1;
    switch (mes)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10: if (dia == 32)
            {
                dia = 1;
                mes = mes + 1;
                break;
            }
        case 12: if (dia == 32)
            {

```

```

        dia = 1;
        mes = 1;
        año = año + 1;
        break;
    }
    case 4:
    case 6:
    case 9:
    case 11: if (dia == 31)
        {
            dia = 1;
            mes = mes + 1;
            break;
        }
    case 2: if (dia == 29 && !añoBisiesto(año) ||
        dia == 30 && añoBisiesto(año))
        {
            dia = 1;
            mes = 3;
            break;
        }
    }
}

private boolean añoBisiesto (int añoB)
{
    return (añoB % 400 == 0 || (añoB % 4 == 0 &&
        añoB % 100 != 0));
}

private void describirPrograma ()
{
    cuadroSalida.println("Introducir los siguientes datos");
    cuadroSalida.println("año (4 dígitos), mes y día");
    cuadroSalida.skipLine(2);
}

private void obtenerEntrada ()
{
    año = cuadroEntrada.getInteger("Año: ");
    mes = cuadroEntrada.getInteger("Mes: ");
    dia = cuadroEntrada.getInteger("Día: ");
}

private void mostrarFechaEntrada ()
{
    cuadroSalida.println("La fecha de entrada es:");
    cuadroSalida.println(dia + "-" + mes + "-" + año);
}

private void mostrarFechaSiguiente ()

```

```

    {
        cuadroSalida.println("La fecha del día siguiente es:");
        cuadroSalida.print(dia + "-");
        switch (mes)
        {
            case 1: cuadroSalida.print("enero");
                    break;
            case 2: cuadroSalida.print("febrero");
                    break;
            case 3: cuadroSalida.print("marzo");
                    break;
            case 4: cuadroSalida.print("abril");
                    break;
            case 5: cuadroSalida.print("mayo");
                    break;
            case 6: cuadroSalida.print("junio");
                    break;
            case 7: cuadroSalida.print("julio");
                    break;
            case 8: cuadroSalida.print("agosto");
                    break;
            case 9: cuadroSalida.print("septiembre");
                    break;
            case 10: cuadroSalida.print("octubre");
                    break;
            case 11: cuadroSalida.print("noviembre");
                    break;
            case 12: cuadroSalida.print("diciembre");
                    break;
        }
        cuadroSalida.println("-" + año);
    }
}

```

b. Las modificaciones que hay que hacer en el autoexec.bat son:

```

PATH = %PATH%; C:\JDK1.3.1\BIN
SET CLASSPATH = .; C:\JAVA

```

La primera instrucción es para indicar al SO donde debe buscar el compilador y la máquina virtual de Java.

El segundo comando es para proporcionar el directorio donde se encuentran las clases predefinidas de Javabook y utilizadas en los programas de Java: MainWindow, InputBox y OutputBox.

A continuación debemos teclear los siguientes comandos:

```

C:\> CD JAVA\APLICACIONES

```

```

C:\JAVA\APLICACIONES> javac FechaMainI.java FechaI.java

```

Si no hay errores de compilación, obtenemos dos ficheros denominados FechaMainI.class y FechaI.java.

```

C:\JAVA\APLICACIONES> java FechaMainI

```

Sólo se ejecuta el archivo que contiene el método main.