

PRUEBA 1 PROGRAMACIÓN
Noviembre 2008
INGENIERÍA INFORMÁTICA
Colmenarejo



Universidad
Carlos III de Madrid

LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:

- Rellene todas las hojas a bolígrafo, tanto los datos personales como las respuestas
- No utilice lápiz ni bolígrafo rojo
- No olvide rellenar el NIA y el grupo real al que pertenece
- El tiempo máximo de realización es de 1 hora
- El único material permitido sobre la mesa es la hoja de test y un bolígrafo
- Utilice exclusivamente esta hoja de test para las respuestas, use las caras posteriores para contestar si lo necesita. No se recogerá ninguna otra hoja adicional.

NO PASE DE ESTA HOJA, hasta que se le indique

Apellidos	Nombre	
Firma	NIA	Grupo

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Indicar si la siguiente afirmación es cierta, y **explicar** brevemente por qué.

*“Para asignar a una variable de tipo **short** el valor de una variable de tipo **byte** es necesario hacer un casting”*

Falso, ya que java reserva 16 bits para una variable de tipo short y sólo 8 bits para una variable de tipo byte, por lo que una variable de tipo byte “entra” en una de tipo short, por lo que el casting es automático, no es necesario hacerlo explícito.

Pregunta 2 (1 Punto).- Indicar si la siguiente afirmación es cierta, y **explicar** brevemente por qué.

*“En java tenemos dos maneras de hacer un bucle, utilizando **for** o utilizando **while**”*

Falso, existe otro tipo de bucle que es el **do-while**.

Pregunta 3 (1 Punto).- Explicar brevemente para qué sirve el Recolector de Basura (Garbage Collector).

El garbage collector o recolector de basura es un proceso que corre dentro de la Java Virtual Machine, y que se encarga de liberar la memoria de los objetos que ya no estamos utilizando.

En lenguajes como el C no existía recolector de basura, por lo que era necesario liberar la memoria explícitamente, y esto era una fuente muy común de errores de programación.

Pregunta 4 (1 Punto).- Explicar cuál sería el resultado de compilar y ejecutar la siguiente clase java:

```
public class ProblemaCuatro {  
    public static void main(String[] args) {  
        for(int i=0; i<12; i++) {  
        }  
        if(i==12) {  
            System.out.println("Doce");  
        }  
        System.out.println("Fin");  
    }  
}
```

La clase no compila, ya que la variable **i** está definida dentro del ámbito del bucle **for**, por lo que no podemos utilizarla fuera de este ámbito (la sentencia **if** es incorrecta porque hace uso de la variable **i** fuera de su ámbito).

Pregunta 5 (1 Punto).- Explicar cuál sería el resultado de compilar y ejecutar la siguiente clase java:

```
public class ProblemaCinco {  
    public static void main(String[] args) {  
        int i,j;  
        for(i=0; i<=3; i++) {  
            for(j=0; j<1; j++) {  
                System.out.println(j+i);  
            }  
            System.out.println(i++);  
        }  
    }  
}
```

El programa compila, y la salida por pantalla es la siguiente:

```
0  
0  
2  
2
```

Pregunta 6 (1 Punto).- Dada la siguiente clase java, reescribirla para que realice la misma función sin utilizar el bucle **for** (suponer que en java no existiese el bucle **for** y quisiésemos implementar la misma clase)

```
public class ProblemaSeis {  
    public static void main(String[] args) {  
        int[] arr = new int[1500];  
        int i;  
        for(i=arr.length; i>0; i--) {  
            arr[i-1] = i+1;  
        }  
    }  
}
```

```
public class ProblemaSeis {  
    public static void main(String[] args) {  
        int[] arr = new int[1500];  
        int i;  
        i = arr.length;  
        while(i>0) {  
            arr[i-1] = i+1;  
            i--;  
        }  
    }  
}
```

PARTE 2: PROBLEMAS

Problema 1 (2 Puntos).- Escribir el código de una clase java que cree un array de 1000 posiciones de tipo `long` y lo rellene de la siguiente manera:

- Las casillas de posiciones pares deben contener un número que represente el número de casillas que quedan hasta el final (la casilla en la posición cero debe contener 999, la casilla en la posición dos debe contener 997, y así sucesivamente).
- Las casillas de posiciones impares deberán tener como valor el triple de su índice (la casilla en la posición 1 contendrá 3, la casilla en la posición 3 contendrá 9, y así sucesivamente).

```
public class ProblemaUno {  
    public static void main(String[] args) {  
        long[] array = new long[1000];  
        for(int i=0; i<array.length; i++) {  
            if(i%2 == 0) {  
                array[i] = array.length - i - 1;  
            }  
            else {  
                array[i] = 3 * i;  
            }  
        }  
    }  
}
```

Problema 2 (2 Puntos).- Escribir el código de una clase java que realice lo siguiente:

- Crear un array de dos dimensiones de números enteros de tipo `int` de tamaño 50x50.
- Inicializar el array de manera que
 - o Todas las casillas de la fila cero contengan el valor 1.
 - o Todas las casillas de la fila uno contengan el valor 2.
 - o Todas las casillas de la fila dos contengan el valor 3.
 - o ...
 - o Todas las casillas de la fila cuarenta y nueve contengan el valor 50.
- Imprimir por pantalla el array en forma de matriz, de manera que los valores entre columnas estén separados por coma, y exista un salto de línea entre cada fila.

```
public class ProblemaDos {  
    public static void main(String[] args) {  
        int[][] array = new int[50][50];  
        for(int i=0; i<array.length; i++) {  
            for(int j=0; j<array.length; j++) {  
                array[i][j] = i+1;  
            }  
        }  
  
        for(int i=0; i<array.length; i++) {  
            for(int j=0; j<array.length; j++) {  
                System.out.print(array[i][j]);  
                System.out.print(",");  
            }  
            System.out.println();  
        }  
    }  
}
```