

**PRUEBA 1 PROGRAMACIÓN  
Marzo 2007  
INGENIERÍA INFORMÁTICA**



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 olvide rellenar el NIA y el grupo real al que pertenece.
- El tiempo máximo de realización es de 50 minutos
- El único material permitido sobre la mesa es la hoja de test y un bolígrafo

**NO PASE DE ESTA HOJA, hasta que se le indique**

<b>Apellidos</b>	<b>Nombre</b>	
<b>Firma</b>	<b>NIA</b>	<b>Grupo</b>

**PARTE 1: CUESTIONES**

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

*Si en la definición de una clase no escribimos un método constructor, no es posible compilar la clase.*

Falso. Si no escribimos ningún método constructor, Java creará uno por defecto sin parámetros que pondrá los atributos a sus valores por defecto: 0 para números, cadena vacía para char, false para booleanos y null para objetos.

---

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

*En Java no es necesario hacer conversiones entre tipos porque se hacen automáticamente*

Falso. Las conversiones entre tipos son automáticas sólo si los tipos son compatibles y el rango del tipo de destino es mayor que el del tipo de origen, por ejemplo la conversión es automática para pasar de *int* a *double*, o de *int* a *float*. Pero si el tipo de destino es de menor rango que el de origen, no es posible hacer la conversión automática y hay que hacer un *casting* manual, como por ejemplo se hace a continuación:

```
double d=3.0;
```

```
int a = (int) d;
```

**Pregunta 3 (1 Punto).**- Indicar si la siguiente afirmación es cierta, y explicar brevemente por qué.

*"String no es un tipo básico, aunque muchas veces se trata como tal, y un ejemplo de esto es la existencia de un envoltorio (clase para tratar los tipos básicos como si fuesen objetos) que se llama String."*

Falso. La primera parte de la afirmación es verdadera, String es una clase a la que se puede tratar muchas veces como a un tipo básico. Pero no hay un envoltorio asociado a String, puesto que ya es una clase de Java.

**Pregunta 4 (1 Punto).**- Corregir y explicar el error de ejecución que hay en el siguiente código y mostrar la salida.

```
public class Preguntal {
    int linea[];
    static int n;

    Preguntal(int t){
        n=n+1;
        for (int i=0; i<t; i++) linea[i]=i+1;
    }
    void imprimir(){
        for (int i=0; i<linea.length; i++)System.out.print(linea[i]+" ");
        System.out.println();
        System.out.println("n: "+n);
    }
}

public class Ejecutar1 {
    public static void main (String[] args){
        Preguntal p2 = new Preguntal(4);
        p2.imprimir();
    }
}
```

Falta definir el tamaño del array en el constructor:

linea = new int [t];

El error de ejecución ocurre en el constructor al asignarles valores a las posiciones del array:

"Exception in thread "main" java.lang.NullPointerException"

El constructor da valor 1 a la posición 0 del array, 2 a la posición 1, y así sucesivamente.

El método imprimir, imprime el array y su longitud.

Salida:

1 2 3 4

n: 1

---

**Pregunta 5 (1 Punto).**- Escribir un método que devuelva el factorial de un entero mayor o igual que cero

```
public int factorial (int x){

    int salida = 1;
    for (int i=1; i<=x; i++) salida = salida * i;
    return salida;
}
```

**PARTE 2: PROBLEMAS**

**Problema 1 (3 Puntos).-** Dado el siguiente código java:

```
public class Alumno{

    private final static int CURSOS = 5; // nº de cursos de la carrera
    private String nombre;                // nombre del alumno
    private int id;                       // identificador del alumno
    private int anio_nacimiento;          // año de nacimiento del alumno
    private double notas[]; // notas medias de cada uno de los 5 años
    private static int total_alumnos = 0; // número total de alumnos
}
```

Escribir los siguientes métodos de la clase Alumno:

1. Un constructor que tenga como parámetros el nombre y la fecha de nacimiento del alumno.
2. Un método mostrar que muestre por pantalla el nombre del alumno, el identificador y las notas de los 5 años.
3. Un método setNota que asigne una nota media del alumno en el año correspondiente. Controlar que el año indicado sea correcto (del primer al quinto).
4. Un método getNota: que devuelve la nota media del alumno en el año indicado. Controlar que el año indicado sea correcto.

```
public Alumno(String n, int an){

    nombre = new String(n);
    anio_nacimiento = an;
    notas = new double[CURSOS];
    id = total_alumnos +1;
    total_alumnos++;
}

public void mostrar(){

    System.out.println("Nombre: "+nombre);
    System.out.println("Id:      "+id);
    System.out.println("Notas:  ");
    for (int i=0; i<CURSOS; i++) System.out.println(notas[i]+" ");
    System.out.println();
}

public void setNota(int anio, double not){

    if (anio>0 && anio<=CURSOS){
        notas[anio-1] = not;
    }
    else System.out.println("No existe ese año académico");
}

public double getNota(int anio){

    double salida = -1.0;
    if (anio>0 && anio<=CURSOS){
        salida = notas[anio];
    }
    else System.out.println("No existe ese año académico");
    return salida;
}
```



**Problema 2 (2 Puntos).-** Dado el siguiente código java:

```
public class UsoSuma3 {  
    public static void main(String[] args) {  
        String linea;  
        int numero=23;  
        Suma3 s= new Suma3(numero);  
        System.out.println(s.devuelveId());  
    }  
}
```

Implementar la clase Suma3, sabiendo que está compuesta por un atributo, un método que devuelve el valor de ese atributo y un constructor, de manera que la salida a consola sea:

26

Hay varias soluciones para este problema, una posible solución es:

```
public class Suma3 {  
    private int id;  
  
    public Suma3 (int id){  
        this.id = id+3;  
    }  
  
    public int devuelveId(){  
        return id;  
    }  
}
```