

PRUEBA 2 PROGRAMACIÓN Diciembre 2008 INGENIERÍA INFORMÁTICA Colmenarejo		 Universidad Carlos III de Madrid	
Apellidos		Nombre	
Firma		NIA	Grupo

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.

PARTE 1: CUESTIONES

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

“Para definir un método de clase hay que poner la palabra `static` en su declaración”

Verdadero, para definir un método o estático o “de clase”, se utiliza la palabra `static` dentro de la declaración del método. Esto hace que el método pueda ser accedido de manera estática, es decir, sin necesidad de ser ejecutado sobre una instancia de la clase.

Pregunta 2 (1 Punto).- Indicar y **explicar brevemente** cuál es el resultado por pantalla del siguiente programa:

```
public class ExamenPregunta2 {  
    public static void main(String[] args) {  
        String uno = "hola ";  
        System.out.println(metodo1(uno));  
    }  
  
    public static String metodo1(String dos) {  
        System.out.println(dos + "amparo");  
        return metodo3(dos);  
    }  
  
    public static String metodo3(String tres) {  
        System.out.println(tres + "antonia");  
        return (tres + "fermina");  
    }  
}
```

Al ejecutar el programa, lo primero que se hace es invocar a `metodo1` pasándole como argumento la cadena de texto “hola”. En el `metodo1`, se imprime “hola amparo” y posteriormente se llama al `metodo3` también con la cadena “hola”. En el `metodo3` se imprime por pantalla “hola antonia” y se devuelve la cadena de texto “hola fermina”, que a su vez el `metodo1` devuelve a `main`. Desde el `main` por último se imprime el valor devuelto, “hola fermina”. Por tanto, la salida por pantalla al ejecutar el programa es:

```
hola amparo  
hola antonia  
hola fermina
```

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

"Para utilizar clases definidas en el mismo paquete es necesario importarlas a través de la palabra clave import"

Falso, en java se importan automáticamente las clases definidas dentro del mismo paquete que el paquete actual, por tanto no es necesario importarlas explícitamente utilizando la palabra clave "import".

Pregunta 4 (1 Punto).- Encontrar y **explicar** los 3 errores de compilación que aparecen en el siguiente código Java. ¿Cómo los resolvería?

```
public class ExamenPregunta4 {  
    public static void main(String[] args) {  
        long a = 5;  
        int b;  
        b = devuelveElDoble(a);  
        System.out.println("El doble de " + a + " es:" b);  
    }  
  
    public static devuelveElDoble(int valor) {  
        return 2 * valor;  
    }  
}
```

1) El primer error es que se ha omitido el tipo del **retorno** del método devuelveElDoble. Para corregirlo, habría que cambiar la declaración del método para meter el tipo int:

```
public static int devuelveElDoble(int valor) {
```

2) El segundo error es que la variable "a" es de tipo **long** y se está intentando pasar como argumento al método devuelveElDoble que recibe un entero de tipo **int**. Para arreglar esto, es necesario realizar un casting de la variable en la llamada al método:

```
b = devuelveElDoble((int)a);
```

3) El tercer error es que se ha omitido el operador de concatenación de cadenas de texto "+" al intentar imprimir por pantalla la variable "b". Para corregirlo habría que meter el "+":

```
System.out.println("El doble de " + a + " es:" + b);
```

Pregunta 5 (1 Punto).- Crear un método estático llamado *invertir* que reciba un array de caracteres de **cualquier** longitud y devuelva otro array de caracteres en el que los elementos estén en orden inverso. Ejemplo: recibe {'a', 'b', 'c', 'd'} y devuelve {'d','c','b','a'}

```
public static char[] invertir(char[] original) {  
    int len = original.length;  
    char[] ret = new char[len];  
    for(int i=0; i<len; i++) {  
        ret[i] = original[len-i-1];  
    }  
    return ret;  
}
```

Pregunta 6 (1 Punto).- Indicar y **explicar brevemente** cuál es el resultado por pantalla del siguiente programa:

```
public class ExamenPregunta6 {  
    public static void main(String[] args) {  
        for(int i=1; i<=3; i++) {  
            for(int j=1; j<4; j++) {  
                if(i % 2 == 0) {  
                    continue;  
                } else {  
                    System.out.println(i*j);  
                    break;  
                }  
            }  
        }  
    }  
}
```

El primer bucle avanza desde 1 hasta 3 con la variable i. El segundo bucle, para cada uno de los tres valores de i, avanza con la variable j desde 1 hasta 3.

Dentro del segundo bucle bucle, si se trata de valores pares de i, no hace nada (utiliza continue). Si se trata de valores impares de i, entonces se imprime el producto de i por j y se sale del bucle (por tanto el segundo bucle no es necesario porque en cualquier caso para valores pares no hace nada y para valores impares sólo se ejecuta una vez).

Por tanto **sólo se imprimirán por pantalla los valores impares de i, multiplicados por el primer valor que toma j (por tanto multiplicados por uno).**

La salida por pantalla es:

1
3

Pregunta 7 (1 Punto).- Dada la siguiente clase:

```
public class ClaseTemporal {  
    public int var;  
    public String var2;  
    public ClaseTemporal(int var, String var2) {  
        this.var = var;  
        this.var2 = var2;  
    }  
    public ClaseTemporal() {  
        this(2, "eugenio");  
    }  
}
```

Indicar y **explicar brevemente** cuál es el resultado por pantalla del siguiente programa:

```
public class ExamenPregunta7 {  
    public static void main(String[] args) {  
        ClaseTemporal uno = new ClaseTemporal();  
        ClaseTemporal dos = new ClaseTemporal(3, "chiquito");  
        System.out.println(uno.var);  
        System.out.println(uno.var2);  
        System.out.println(dos.var);  
        System.out.println(dos.var2);  
        dos = uno;  
        dos.var = 5;  
        dos.var2 = "ozores";  
        System.out.println(uno.var);  
        System.out.println(uno.var2);  
    }  
}
```

Al crearse la primera instancia de ClaseTemporal (que se almacena en la variable “uno”), se invoca al constructor sin parámetros. A su vez, este constructor invoca al constructor de dos parámetros con los valores “2” y “eugenio”.

Al crearse la segunda instancia de ClaseTemporal (que se almacena en la variable “dos”), se invoca directamente al constructor de dos parámetros con los valores “3” y “chiquito”.

Posteriormente se imprimen por pantalla los valores de las variables de “uno” y “dos”, por tanto se imprime por pantalla:

2

eugenio

3

Chiquito

Después se hace que el objeto dos apunte al objeto uno, y se modifican los valores de dicho objeto (por tanto se verán modificados los valores del objeto uno). Así, finalmente al imprimir los valores del objeto uno se volcará por pantalla:

5

ozores

PARTE 2: PROBLEMAS**Problema 1 (3 Puntos).-**

Crear una clase llamada `Casa` que tendrá las siguientes características:

- (0,2 puntos) Debe tener 3 atributos públicos denominados `direccion`, `altura` y `nombreDueño`.
- (0,2 puntos) Hacer un método denominado `getDireccion` que devuelva la dirección de la casa.
- (0,4 puntos) Hacer un método `setAltura` que reciba como parámetro la altura de la casa en metros. Deberá comprobar que el valor recibido es válido, sabiendo que la altura no puede ser inferior a 3 metros ni superior a 100. En tal caso imprimirá un error por pantalla.
- Hacer los siguientes constructores (todos deberán comprobar que la altura recibida es correcta)
 - o (0,5 puntos) Uno que reciba valores para todos los atributos de la clase `Casa`.
 - o (0,4 puntos) Uno por defecto, sin parámetros que usando el anterior cree una `Casa` de 10 metros, en la *calle Mayor número 7*, y cuyo dueño sea Mariano.
 - o (0,4 puntos) Uno que sólo reciba valores para la dirección y el dueño y ponga la altura a 5 metros.
 - o (0,4 puntos) Uno de copia que reciba como parámetro un objeto de tipo `Casa` y cree otra casa con los mismos atributos.

(0,5 puntos) Crear una clase Denominada `UsoCasa` que cree cuatro objetos de tipo `Casa`, uno con cada uno de los constructores anteriores.

```
public class Casa {
    public String direccion;
    public float altura;
    public String nombreDueño;

    public String getDireccion() {
        return direccion;
    }

    public void setAltura(float altura) {
        if(!alturaCorrecta(altura)) {
            System.out.println("Error, la altura tiene que estar entre 3 y 100
metros");
            return;
        }
        this.altura = altura;
    }

    public boolean alturaCorrecta(float altura) {
        return (altura >=3 && altura <=100);
    }

    public Casa(String direccion, float altura, String nombreDueño) {
        if(!alturaCorrecta(altura)) {
```

```
        System.out.println("Error, la altura tiene que estar entre 3 y 100
metros");
        return;
    }
    this.direccion = direccion;
    this.altura = altura;
    this.nombreDueño = nombreDueño;
}

public Casa() {
    this("calle Mayor número 7", 10, "Mariano");
}

public Casa(String direccion, String nombreDueño) {
    this.direccion = direccion;
    this.altura = 5;
    this.nombreDueño = nombreDueño;
}

public Casa(Casa original) {
    if(!alturaCorrecta(original.altura)) {
        System.out.println("Error, la altura tiene que estar entre 3 y 100
metros");
        return;
    }
    this.direccion = original.direccion;
    this.altura = original.altura;
    this.nombreDueño = original.nombreDueño;
}

}

public class UsoCasa {
    public static void main(String[] args) {
        Casa c1 = new Casa("direccion 1", 10, "Jose");
        Casa c2 = new Casa();
        Casa c3 = new Casa("direccion 3", "Luis");
        Casa c4 = new Casa(c3);
    }
}
```