EXAMEN DE PROGRAMACIÓN 18 de Junio de 2010 GRADO EN INGENIERÍA INFORMÁTICA Leganés	Universidad Carlos III de Madrid	
Apellidos	***Nombre***	***Grupo***

LEA ATENTAMENTE ESTAS INSTRUCCIONES ANTES DE COMENZAR LA PRUEBA:

- **Rellene todas las hojas a bolígrafo, tanto datos personales como respuestas**
- **No utilice lápiz ni bolígrafo rojo**
- **No olvide rellenar el grupo real al que pertenece**
- **El tiempo máximo de realización es de 3 horas**
- **Se permiten apuntes y/o libros para la realización del examen**
- **Para responder las cuestiones 1 a 3, utilice exclusivamente estas hojas de test. No se recogerá ninguna otra hoja adicional.**
- **Para los problemas 1 a 7, utilice las hojas adicionales que se han entregado.**

PARTE 1: CUESTIONES

Pregunta 1 (1 Punto).- Explicar el resultado de ejecutar el siguiente método:

```
public static int [][] metodo1 (int [] val){
      int k=0, j=0;
      int result [][];
      while (k<val.length){
            if (val[k]==1) j=j+1;
            k++;
      }
      k=0;
      result = new int [j][2];
      for (j=1;j<val.length;j++){
            if (val[j]==1) {
                  result[k][0]=j-1;
                  result[k][1]=val[j-1];
                  k++;
            }
      }
      return result;
}
```

El método recibe un array de enteros y devuelve una matriz de enteros. Mediante el primer bucle (`while`) se va recorriendo el array recibido como parámetro y se cuentan cuántos de sus elementos son iguales a `1`. El resultado se guarda en la variable `j`. A continuación se crea una matriz de `j` filas y 2 columnas. Para rellenarla se vuelven a buscar los `1` en el array y cada vez que se encuentra uno, en la matriz se pone en la primera fila, la posición del elemento anterior a ese `1` y en la segunda, el valor de ese elemento. En resumen, el método devuelve una matriz con las posiciones y los valores de los elementos de un array que están justo antes de un `1`.

Pregunta 2 (1 Punto).- Encontrar y explicar los 4 errores de compilación que aparecen en el siguiente código Java. ¿Cómo los resolvería?

```
public class Pregunta2 {
	static int c;

	public static metodo1 (String a, String b){
		String c = a+b;
		if (c.length()>2)
			c = c.substring(2);
		else c=b+a;
		System.out.println(c);
	}

	public static int metodo2 (){
		return c;
	}

	public static String metodo3(int j){
		c=j;
		System.out.println(c);
	}

	public static void metodo4 (int j){
		int k;
		c=k+j;
		System.out.println(c);
	}

	public static void main(String[] args) {
		int k = 16;
		metodo2(k);
		k = metodo2();

	}}
```

a) En `metodo1` no se especifica el tipo del valor que devuelve. Puesto que no devuelve nada, habría que decir que es `void`.

b) En `metodo3` se dice que va a devolver un `String` y no se devuelve nada. Habría que o cambiar el método a `void` o devolver un `String` usando `return`.

c) En `metodo4` se usa la variable `k` antes de haberle dado un valor. Habría que haberle dado valor antes.

d) En el método `main` se llama a `metodo2(k)`, pero `metodo2` no admite parámetros.

Pregunta 3 (1 Punto).- Explicar cuál es el resultado del siguiente código:

```
public class Pregunta3 {

	public static void main(String [] args) {

		int [] N = new int [] {1, 2, 3, 4, 5};
		int k = 2;

		metodo1(N, k + 1);

		System.out.println(k);
		metodo2(N);
	}

	public static void metodo1(int [] N, int k) {
		for(int i=0; i < N.length; i++) {
			N[i] *= k;
		}
		k = 0;
	}

	public static void metodo2(int [] N) {
		System.out.print("[");
		for(int i=0; i<N.length; i++) {
			System.out.print(N[i] + " ");
		}
		System.out.print("]");
	}
}
```

El método `main` declara un array de `int` y una variable `int`, a continuación llama a `metodo1` pasando el array como primer parámetro y el valor de la variable más 1 como segundo parámetro. En `metodo1` el bucle va recorriendo el array y multiplicando cada valor por 3 (k+1). A continuación pone k a 0. Como los array se pasan por referencia, el array original cambia, en cambio las variables de tipos básicos se pasan por valor, por lo que el valor original de k no cambia. En `metodo2` se coge el array que se recibe por parámetro y se imprimen sus valores entre corchetes. Por lo tanto, el resultado por pantalla será:

```
2
[3 6 9 12 15 ]
```

PARTE 2: PROBLEMAS

Problema 1 (1 Punto).- Crear un método denominado `contar` que reciba como parámetros un array de `String` y un `String` y devuelva un `int`, que será el resultado de contar cuántos de los elementos del array de `String` empiezan por el `String`.

Ejemplo, si ejecutamos:

```
String [] lista = new String [] {"hola", "adios", "arroz", "pescado", "foca",
"arrogante"};
int b = contar(lista,"a");
```

b valdrá 3

Nota: Se recomienda usar el método apropiado de la clase String.

SOLUCIÓN:

```
public static int contar (String [] a, String b){
      int result = 0;
      for (int j=0;j<a.length; j++)
            if (a[j].startsWith(b)) result++;
      return result;
}
```

Problema 2 (1 Punto).- Crear un método denominado contarVarios, que reciba como parámetros dos arrays de `String` e imprima por pantalla cuántos elementos del primer array de `String` empiezan por las letras del segundo array.

Ejemplo, si ejecutamos:

```
String [] lista = new String [] {"hola", "adios", "arroz", "pescado", "foca",
"arrogante"};
String [] c = new String []{"arr","p"};
contarVarios(lista,c);
```

Imprimirá:

```
Hay 2 palabras que empiezan por arr
Hay 1 palabras que empiezan por p
```

Nota: Se recomienda usar el método creado en el problema anterior

SOLUCIÓN:

```
public static void contarVarios (String [] a, String [] b){
      int result = 0;
      for (int j=0; j<b.length; j++){
            result = contar (a, b[j]);
            System.out.println("Hay "+result+" palabras que empiezan por
             "+b[j]);
      }
}
```

Problema 3 (1 Punto).- Crear un método que ordene números de tipo `double` de mayor a menor mediante el algoritmo de inserción directa.

SOLUCIÓN:

```
public static void insercionDirecta (double [] lista){
      for (int i=1; i<lista.length; i++){
            double auxiliar = lista[i];
            int j=i-1;
            while (j>=0 && auxiliar>lista[j]){
                  lista [j+1]=lista[j];
                  j--;
            }// fin while
            lista[j+1]=auxiliar;
      }// fin for
```

```
}// fin insercionDirecta
```

Problema 4 (1 Punto).- Crear un método denominado `numeros` que no reciba parámetros y que devuelva un array de `int` con 30 números aleatorios en el intervalo [2, 20).

SOLUCIÓN:

```
public static int [] numeros (){
      int result [] = new int [30];
      for (int j=0; j<30; j++)
            result [j] = (int) (Math.random()*18+2);
      return result;
}
```

Problema 5 (1 Punto).- Crear un método denominado `contiene` que reciba una matriz de `float` de cualquier dimensión (Ejemplo 3x2, 6x23, etc.) y un número y devuelva un `boolean` indicando si el número está en la matriz o no.

SOLUCIÓN:

```
public static boolean contiene (float a[][], float b){
      boolean resultado = false;
      for (int i=0;i<a.length;i++)
            for (int j=0;j<a[i].length;j++)
                  if (a[i][j]==b) resultado=true;
                  return resultado;
}
```

Problema 6 (1 Punto).- Crear un método, denominado `elMayor`, que no reciba parámetros, que pida números enteros positivos por teclado hasta que se introduzca -1 y que devuelva el mayor de todos los introducidos.

SOLUCIÓN: Hay que hacer un import java.io.*; en la clase.

```
public static int mayor () throws IOException {
      BufferedReader br = new BufferedReader (new InputStreamReader
       (System.in));
      int numero = 1;
      int maximo = 0;
      while (numero!=-1){
            numero = Integer.parseInt(br.readLine());
            if (numero>maximo) maximo = numero;
      }
      return maximo;
}
```

Problema 7 (1 Punto).- Crear un programa, cuyo método `main` reciba como argumentos un número `double` (que llamaremos `x`) y varios números enteros (que llamaremos coeficientes) y que calcule e imprima por pantalla el valor del polinomio definido por los coeficientes para el valor `x`.

Ejemplo:

Si llamamos al programa con esta sintaxis: `java Problema7 1.8  4 2 3`
el polinomio definido por los coeficientes sería 4 + 2x + 3x^2

y el resultado imprimido por pantalla sería 17.32 (puesto que 4 + 2*1.8 + 3*1.8^2 = 17.32)

El programa debe funcionar para cualquier valor x y para cualquier número de coeficientes.

SOLUCIÓN:

```
public static void main(String [] args) {
	//Comprobamos que hay al menos 2 argumentos
	if(args.length < 2) {
		System.out.println("[ERROR] Sintaxis: ");
		System.out.println(
		"java Problem7 <vaor x> <coeficiente x^0> <coeficiente x^1> ...");
		System.exit(-1);
	}
	// x
	double x = 0;
	x = Double.parseDouble(args[0]);
	// coeficientes
	int [] c = new int[args.length - 1];
	for(int i=1; i<args.length; i++)
		c[i-1] = Integer.parseInt(args[i]);
	double r = 0;
	for(int i=0; i<c.length; i++) {
		double aux = 1;
		for(int j=0; j < i; j++) {
			aux = aux * x;
		}
		r = r + c[i] * aux;
	}
	System.out.println("Result: " + r);
}
```