



Apellidos: Mier Montoto
Nombre: Juan Francisco

D.N.I.: 71777658V



INTRODUCCIÓN A LA PROGRAMACIÓN - E.P. DE INGENIERÍA DE GIJÓN

14 de Enero de 2021

1. (2 %) Escribe una expresión para declarar un vector de números reales cuyo tamaño se lee de un objeto Scanner llamado teclado, que supondremos definido previamente.

10 double [] vector = new double [teclado.nextInt()];

2. (5 %) Dada la siguiente definición de variables y sus valores iniciales,

```
int x=9; double f=2.0; char c='a';
```

indica para las siguientes expresiones, si son o no correctas (SI/NO), en caso de resultar incorrectas JUSTIFICA por qué lo son, y en el caso de ser correctas indica el TIPO y el VALOR que producen.

Nota: Los dígitos, al igual que las letras del alfabeto, ocupan posiciones consecutivas en la tabla de códigos.

		Tipo	Valor	Motivo
✓	f = f / f;	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	double	1.0
✓	f = 1 / y;	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	double	0.0
✓	1 / y	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	int	0
5/6	(char)(c+y)	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No		No se puede sumar un char a un int. SI
✓	c == 'c'	<input checked="" type="checkbox"/> Si <input type="checkbox"/> No	boolean	false
✓	y = c == 'a'	<input type="checkbox"/> Si <input checked="" type="checkbox"/> No		No se puede asignar un boolean a un entero.

3. (12 %) Implementa el método estático piAprox() para calcular (y retornar) de forma aproximada el valor de π . Para ello debes utilizar la serie conocida como *Producto de Wallis*:

$$\frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \cdot \frac{10}{9} \cdot \frac{10}{11} \cdots = \frac{\pi}{2}$$

El método debe recibir un parámetro $n \geq 1$ que indica el número de términos del producto que debemos utilizar para calcular la aproximación. Por ejemplo, si se invoca `double myPi = piAprox(5)` el método debería retornar el resultado de utilizar los 5 primeros términos del producto, es decir,

$$\pi = 2 \cdot \left(\frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \right)$$

public static double piAprox (int x) {
 double resultado = 2; // Siempre se empieza multiplicando por 2
 for (int i=3; i<=x; i+=2) {
 resultado *= (double)(i-1)/i;
 resultado *= (double)(i+1)/i;
 }
 resultado *= 2; // Como la aproximación es para $\frac{\pi}{2}$, se multiplica por 2.
 return resultado;

6
 }
 No se calcula correctamente porque no se
 usa el número de pasos que indica el
 parámetro, siempre se usa un número
 ímpar de términos y si quieras usar 100
 tu usarias 99

4. (12 %) Diseñar el método público y estático *Seguidos* que, dado un vector no vacío de números enteros, calcule la longitud de la subsecuencia más larga de números consecutivos e iguales.

Ejemplo. Para el vector 2, 1, 1, 2, 2, 5, 4, 4, 4 el método debe retornar 3 (ya que el número 4 aparece 3 veces seguidas, no hace falta retornar el 4, solamente el 3)

9 `public static int Seguidos(int[] x) {`

```

int reserva = x[0];
int contador = 1, max = 1;
for (int i = 1; i < x.length; i++) {
    if (x[i] == reserva) {contador++;} ↑
    else {
        reserva = x[i];
if (contador > max) max = contador;
        contador = 1;
    }
}
return max;
& pones esto dentro del if, sobre esto
if (contador > max) max = contador;

```

5. (8 %) Escribir un programa completo en el que se pidan por teclado tres números reales a modo de longitudes de lados y se indique si podrían formar un triángulo o no. Además, en caso afirmativo, el programa debe indicar qué tipo de triángulo sería: *equilátero* (tres lados iguales), *isósceles* (dos lados iguales) o *escaleno* (tres lados distintos).

10 `import static java.lang.System.out`
`import java.util.Scanner;`
`public class TipoTriangulo {`

```

public static void main(String[] args) {
    Scanner scr = new Scanner(System.in);
    out.print("Introduzca los tres lados del triángulo: ");
    double l1 = scr.nextDouble();
    double l2 = scr.nextDouble();
    double l3 = scr.nextDouble();
    scr.close();
    if (l1 + l2 > l3 && l1 + l3 > l2 && l2 + l3 > l1) {
        if (l1 == l2 || l2 == l3 || l1 == l3) {
            if (l1 == l2 && l2 == l3) out.print("Es equilátero.");
            else out.print("Es isósceles.");
        } else out.print("Es escaleno.");
    } else out.print("Esas medidas no se corresponden con la de
                    un triángulo.");
}

```


6. (8 %) Un almacén de fontanería ofrece descuentos en función del nivel de compra de los clientes, de acuerdo a la siguiente tabla:

	Herramientas	Materiales
Ocasional	5%	5%
Habitual	10%	20%
Profesional	15%	20%

Diseña el esquema condicional que determine el descuento a aplicar en función del tipo de cliente y el producto comprado. Puedes suponer que partes de una variable `cliente` cuyo valor puede ser 'O', 'H' o bien 'P', y de una variable `producto` cuyo valor puede ser 'H' o 'M'.

10

```

int descuento;
if (cliente == 'O') descuento = 5;
else if (producto == 'M') descuento = 20;
else if (cliente == 'H') descuento = 10;
else descuento = 15;
    
```

(
)

7. (15 %) Diseñar el método público y no estático `Alternadas` que dado un String que contiene una palabra devuelva `true` si las vocales y consonantes están alternadas, es decir, que no hay dos vocales, ni dos consonantes seguidas, y `false` en caso contrario. Se permite usar, sin programarlo, el método público `boolean esVocal(char)` que recibe un parámetro de tipo `char` y devuelve cierto si es una vocal y falso en caso contrario.

Ejemplos. "hola" → `true`, "adiós" → `false`, ya que hay 2 vocales seguidas, "palabra" → `false`, hay dos consonantes (`b` y `r`) seguidas.

lot

```

public boolean Alternadas (String s)
boolean check = true;
for (int i=0; i+1 < s.length(); i++) {
    if ((esVocal(s.charAt(i)) && !esVocal(s.charAt(i+1))) || 
        (!esVocal(s.charAt(i)) && !esVocal(s.charAt(i+1))))
        check = false;
}
return check;
}

int i=1;
while (i < s.length() && !esVocal(s.charAt(i)) != !esVocal(s.charAt(i+1))) i++;
return i==s.length();
    
```

88 check ... para salir en cuenta encuentras dos vocales o consonantes seguidas (problema de bucle)

zorra; basta comprobar si son del mismo tipo

8. (10 %) Escribir el método público y no estático `Abajo` que, dada una matriz de enteros no vacía, la modifica, desplazando hacia abajo todos los valores de cada columna que son distintos de 0, quedando por tanto abajo los valores distintos de 0 y arriba los ceros.

Antes					Después				
0	3	6	1	2					
2	3	6	0	0					
0	0	0	12	0					
0	0	0	0	0					

⇒

0	0	0	0	0
0	0	0	0	0
0	3	6	1	0
2	3	6	12	2

```

public void moverAbajo(int [][] tablero) {
    for (int i=0; i<1 < tablero.length; i++) {
        for (int j=0; j<tablero[i].length; j++) {
            if (tablero[i][j] != 0 && tablero[i+1][j]==0) {
                tablero[i+1][j]=tablero[i][j];
                tablero[i][j]=0;
            }
        }
    }
}

public void Abajo (int [][] tablero) {
    boolean check = false;
    while (!check) {
        check = true;
        for (int i=0; i<1 < tablero.length; i++) {
            for (int j=0; j<tablero[i].length; j++) {
                if (tablero[i+1][j]==0) {moverAbajo(tablero); check=false;}
            }
        }
    }
}

```

9. (16 %) Implementa la clase Colchón para representar colchones. La información que debe recoger cada objeto de esta clase consiste en las dimensiones en cm. (largo, ancho y grosor) así como el tipo de material, que puede ser un valor entero entre 1 y 4 (1=espuma, 2=visco, 3=látex, 4=muelles)

- Constructores: por defecto (con los siguientes valores para cada atributo: largo=190, ancho=90, grosor=21, tipo=1), el de copia, con 1 entero para inicializar el tipo (los otros tres atributos tomarán los valores por defecto antes indicados), con 4 enteros para poder inicializar los 4 atributos.
- Métodos `get()` y `set()`. Ten en cuenta que el largo sólo puede ser 180, 190 o 200, el ancho sólo puede ser 90, 135 o 200, el grosor será de, al menos, 10 cm. y el tipo sólo puede ser 1, 2, 3 o 4.
- Método `calculaVolumen()`, que calcula el volumen del colchón.
- Método `esMásVoluminoso()` que recibe un objeto `Colchón` y devuelve cierto si el objeto con el que se llama al método tiene un volumen mayor que el objeto que se pasa como parámetro, y falso en caso contrario.
- Método `toString()`, que devuelve un String con la información del objeto `Colchón` con el siguiente formato: "Tipo: espuma. Medidas: 190 cm. x 135 cm. x 32 cm.". (largo x ancho x grosor)

10. (12 %) Se pide implementar la clase Préstamo, que debe registrar el usuario, el libro y la fecha (día, mes y año) del préstamo realizado en una biblioteca. Teniendo en cuenta que las clases Usuario y Libro ya están implementadas y se corresponden con los diagramas adjuntos, implementa la clase Préstamo con dos constructores y el método `toString()` para que el programa que se incluye a modo de ejemplo funcione correctamente y produzca la salida mostrada más abajo (no es necesario que implementes los `get()` y `set()`, aunque puedes usarlos como si estuviesen implementados). Tampoco es necesario controlar si la fecha es correcta).

```

public class ProgramaBiblioteca {

    public static void main(String[] args) {
        Usuario u1 = new Usuario("Pepito Pérez", 647593);
        Libro l1 = new Libro("Patria", "Fernando Aramburu", 9788490663196L);

        Préstamo p1 = new Préstamo(u1, l1, 10, 1, 2021);
        Préstamo p2 = new Préstamo("Manuela Felgueroso", 456753,
            "Los pilares de la tierra", "Ken Follett", 9788466341783L,
            14, 1, 2021);

        System.out.printf("PRÉSTAMOS:\n%s\n%s\n", p1, p2);
    }
}

```

PRÉSTAMOS:

Préstamo del día 10/01/2021 a Pepito Pérez (647593):

 Título: Patria. Autor: Fernández Aramburu. ISBN: 9788490663196

Préstamo del día 14/01/2021 a Manuela Felgueroso (456753):

 Título: Los pilares de la tierra. Autor: Ken Follett. ISBN: 9788466341783

10

```

public class Préstamo {
    private Usuario usuario;
    private Libro libro;
    private short día;
    private short mes;
    private short año;

    public Préstamo(Usuario u, Libro l, short día, short mes, short año) {
        this.usuario = u;
        this.libro = l;
        this.setFecha(día, mes, año);
    }

    public Préstamo(String nombre, int num, String título, String autor, long ISBN,
                    short día, short mes, short año) {
        this.usuario = new Usuario(nombre, num);
        this.libro = new Libro(título, autor, ISBN);
        this.setFecha(día, mes, año);
    }

    public void setFecha(short d, short m, short a) {this.día=d; this.mes=m;
        this.año=a;}
}

@Override
public String toString() {return String.format("Préstamo del día %02d/%
%02d/%04d a %02d/%02d/%02d", this.getDía(), this.getMes(), this.getAño(),
this.usuario, this.libro);}


```

Libro	Usuario
- título: String - autor: String - ISBN: long + Libro(String, String, long) + Libro(Libro) + setTítulo(String) + getTítulo(): String + setAutor(String) + getAutor(): String + setISBN(long) + getISBN(): long + toString(): String	- nombre: String - num_socio: int + Usuario(String, int) + Usuario(Usuario) + setNombre(String) + getNombre(): String + setNúmero(int) + getNúmero(): int + toString(): String

