

PROGRAMACIÓN ORIENTADA A OBJETOS (Laboratorio de Prácticas)

Titulaciones de Grado en Ingeniería Informática y Grado en Sistemas de Información

Sesión 3 MODULARIDAD Y UTILIZACIÓN DE CLASES BÁSICAS EN JAVA (Soluciones a los ejercicios)

1. Modularidad

1. Realizar un programa que imprima todos los números de un rango de valores. El programa pedirá dos números (el menor y el mayor) y llamará a un método auxiliar encargado de imprimir todos los que se encuentren en su rango.

```
import java.util.Scanner;

public class Ej1 {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduce el menor:");
        int menor = entrada.nextInt();
        System.out.println("Introduce el mayor:");
        int mayor = entrada.nextInt();
        if (menor > mayor) {
            System.out.println("Error en los valores.");
        } else {
            rango(menor, mayor); //llamada a la función
        }
    }

    public static void rango(int menor, int mayor) {
        for (int i = menor; i <= mayor; i++) {
            System.out.println(i);
        }
    }
}
```

2. Reutilizando el programa anterior escribir solo los números pares en el rango de valores. El método auxiliar encargado de imprimir todos los valores que se encuentren en su rango tiene que hacer uso de otra función llamada esPar(int num) que devolverá un booleano indicando si el número es par o no.

```
import java.util.Scanner;

public class Ej2 {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduce el menor:");
        int menor = entrada.nextInt();
        System.out.println("Introduce el mayor:");
        int mayor = entrada.nextInt();
        if (menor > mayor) {
            System.out.println("Error en los valores.");
        }
    }
}
```

```

    } else {
        rango(menor, mayor); //llamada a la función
    }
}

public static void rango(int menor, int mayor) {
    for (int i = menor; i <= mayor; i++) {
        if (esPar(i)) {
            System.out.println(i);
        }
    }
}

public static boolean esPar(int num) {
    return num % 2 == 0;
}
}

```

3. Realizar un programa que determine cuál es el mayor de dos números. El programa tendrá un método mayor que recibirá dos valores de tipo entero y devolverá cual es el mayor. Desde el método main se pedirá al usuario los valores y se llamará al método mayor.

```

import java.util.Scanner;

public class Ej3 {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduce un número:");
        int x = entrada.nextInt();
        System.out.println("Introduce otro número:");
        int y = entrada.nextInt();
        if (x == y) {
            System.out.println("No pueden ser iguales.");
        } else {
            System.out.println("El mayor de los dos es: " + mayor(x, y));
        }
    }

    public static int mayor(int a, int b) {
        if (a > b) {
            return a;
        } else {
            return b;
        }
    }
}

```

4. Repetir el ejercicio anterior con una versión que devuelva el mayor de tres números pero reutilizando el método mayor de dos números.

```

import java.util.Scanner;

public class Ej4 {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduce número 1:");
        int x = entrada.nextInt();
        System.out.println("Introduce número 2:");
        int y = entrada.nextInt();
        System.out.println("Introduce número 3:");
        int z = entrada.nextInt();
        if (x == y && x == z && y == z) {
            System.out.println("No pueden ser iguales.");
        } else {
            System.out.println("El mayor de los dos es: " + mayor3(x, y, z));
        }
    }
}

```

```

public static int mayor(int a, int b) {
    if (a > b) {
        return a;
    } else {
        return b;
    }
}

public static int mayor3(int a, int b, int c) {
    int aux = mayor(a, b);
    return mayor(aux, c);
}
}

```

5. Realizar un programa que permita simular una calculadora. El programa tendrá un método calculadora que recibirá dos valores de tipo double y un carácter que indicará la operación a realizar (+,-,*,/). Desde el método main se pedirá al usuario los valores y la operación y se llamará al método calculadora.

```

import java.util.Locale;
import java.util.Scanner;

public class Ej5 {

    public static void main(String[] args) {
        Scanner entrada = new Scanner(System.in);
        entrada.useLocale(Locale.US);

        System.out.println("Introduce operando 1:");
        double op1 = entrada.nextInt();
        System.out.println("Introduce operando 2:");
        double op2 = entrada.nextInt();
        System.out.println("Introduce operación (+,-,*,/):");
        String dato = entrada.next();
        char ope = dato.charAt(0);
        System.out.println("El resultado es: " + calculadora(op1, op2, ope));
    }

    public static double calculadora(double op1, double op2, char ope) {
        double res;
        switch (ope) {
            case '+':
                res = op1 + op2;
                break;
            case '-':
                res = op1 - op2;
                break;
            case '*':
                res = op1 * op2;
                break;
            case '/':
                res = op1 / op2;
                break;
            default:
                res = 0;
                break;
        }
        return res;
    }
}

```

2. Ejercicios Arrays

1. Escribir un programa que genere 100 números enteros aleatorios entre el 1 y el 500, los almacene en un array, los ordene y a continuación genere un array de caracteres que contenga una 'p' en las posiciones donde hay números pares y una 'i' en los impares. Mostrar el contenido de ambos arrays en filas de 10 elementos formados por pares de número y letra.

```
import java.util.Arrays;
import java.util.Random;

public class Ses3ArraysEj1 {

    public static void main(String[] args) {
        Random rand = new Random();
        int numeros[] = new int[100];
        char paresImpares[] = new char[100];

        //generamos los números
        for (int i = 0; i < 100; i++) {
            numeros[i] = rand.nextInt(500) + 1;
        }

        //ordenamos el array
        Arrays.sort(numeros);

        //comprobamos y generamos las letras
        for (int i = 0; i < 100; i++) {
            if (esPar(numeros[i])) {
                paresImpares[i] = 'p';
            } else {
                paresImpares[i] = 'i';
            }
        }

        //imprimimos
        imprimir(numeros, paresImpares);
    }

    public static boolean esPar(int num) {
        return (num % 2 == 0);
    }

    public static void imprimir(int numeros[], char paresImpares[]) {
        int contador = 1;
        //imprimimos los arrays
        for (int i = 0; i < 100; i++) {
            System.out.print(numeros[i] + " " + paresImpares[i] + " - ");
            if (contador++ % 10 == 0) {
                System.out.println("\n");
            }
        }
    }
}
```

2. Escribir un programa que genere 50 números enteros aleatorios entre el 97 y el 123 que representarán las letras del alfabeto (exceptuando la ñ), los almacene en un array de caracteres, y cuente cuantas vocales se han generado.

```
import java.util.Random;

public class Ses3ArraysEj2 {

    public static void main(String[] args) {
        Random rand = new Random();
        char letras[] = new char[50];
        int contador = 0;

        //generamos las letras
        for (int i = 0; i < letras.length; i++) {
```

```

        int numero = rand.nextInt(27)+97;
        letras[i] = (char) numero;
    }
    //contamos las vocales
    for (int i = 0; i < letras.length; i++) {
        if (esVocal(letras[i])){
            contador++;
        }
    }

    System.out.println("Número de vocales: " + contador);

    //imprimimos el array
    for (int i = 0; i < letras.length; i++) {
        System.out.print(letras[i] + " ");
    }
}

public static boolean esVocal(char c) {
    return (c=='a' || c=='e' || c=='i' || c=='o' || c=='u');
}
}

```

3. Escribir un programa que sea capaz de calcular la letra de un NIF a partir del número del DNI. El programa debe poseer al menos un método encargado de pedir al usuario el DNI de 8 dígitos y otro que calculará la letra del NIF (se pueden añadir más métodos auxiliares). Al finalizar el programa se debe presentar el NIF completo con el formato: ocho dígitos, un guion y la letra en mayúscula; por ejemplo: 00395469-F. La letra se calculará de la siguiente forma: Se obtiene el resto de la división entera del número del DNI entre 23 y se usa la siguiente tabla para obtener la letra que corresponde, esta tabla debe estar almacenada en un array para buscar la letra por su posición.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	R	W	A	G	M	Y	F	P	D	X	B	N	J	Z	S	Q	V	H	L	C	K	E

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Ses3ArraysEj3 {

    public static void main(String[] args) throws IOException {
        String dni = pideDNI();
        int dninum = Integer.parseInt(dni);
        char letra = calculaLetraNIF(dninum);
        System.out.println("NIF: " + dni + "-" + letra);
    }

    public static String pideDNI() throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        String dni;
        do {
            System.out.println("Introduzca un número de DNI:");
            dni = entrada.readLine();
        } while (dni.length()!=8);
        return dni;
    }

    public static char calculaLetraNIF(int dninum) {
        char letras[] = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'};
        int pos = dninum % 23;
        return letras[pos];
    }
}

```

4. Realizar un programa que se encargue de crear una matriz de 3 filas por 20 columnas que representará los sueldos de los empleados de una empresa. En la primera fila se deben generar 20

números decimales aleatorios entre 0 y 300.000 € representando los salarios brutos de los empleados, la segunda fila representará la retención de estos salarios, calculada a partir de la siguiente tabla y por último la tercera fila representará el salario neto, es decir, el salario bruto menos la retención. Presentar por pantalla los datos de la matriz con la información de cada empleado. Ejemplo: *Sueldo del empleado 0: SB: 113763.05745992783 - R: 53468.63700616608 - SN: 60294.420453761755*

Salario Bruto	Retención
Desde 0 €a 17.707	24,00%
Desde 17.708 a 33.007	30,00%
Desde 33.008 a 53.407	40,00%
Desde 53.408 a 120.000	47,00%
Desde 120.001 a 175.000	49,00%
Desde 175.001 a 300.000	51,00%

Nota: De forma voluntaria intentar presentar los sueldos con separadores de miles y solo dos decimales. Ejemplo: Sueldo del empleado 0: SB: 40.185,83 - R: 16.074,33 - SN: 24.111,5

```
import java.util.Random;

public class Ses3ArraysEj4 {

    private static double sueldos[][] = new double[3][20];

    public static void main(String[] args) {
        generaSalariosBrutos();
        generaRetenciones();
        generaSalariosNetos();
        imprimir();
    }

    public static void generaSalariosBrutos() {
        Random rand = new Random();
        for (int i = 0; i < sueldos[0].length; i++) {
            sueldos[0][i] = rand.nextDouble()*300000;
        }
    }

    public static void generaRetenciones() {
        for (int i = 0; i < sueldos[0].length; i++) {
            if (sueldos[0][i] <= 17707) {
                sueldos[1][i] = sueldos[0][i]*0.24;
            } else if (sueldos[0][i] >= 17708 && sueldos[0][i] <= 33007) {
                sueldos[1][i] = sueldos[0][i]*0.3;
            } else if (sueldos[0][i] >= 33008 && sueldos[0][i] <= 53407) {
                sueldos[1][i] = sueldos[0][i]*0.4;
            } else if (sueldos[0][i] >= 53408 && sueldos[0][i] <= 120000) {
                sueldos[1][i] = sueldos[0][i]*0.47;
            } else if (sueldos[0][i] >= 120001 && sueldos[0][i] <= 175000) {
                sueldos[1][i] = sueldos[0][i]*0.49;
            } else if (sueldos[0][i] >= 175001 && sueldos[0][i] <= 300000) {
                sueldos[1][i] = sueldos[0][i]*0.51;
            }
        }
    }

    public static void generaSalariosNetos() {
        for (int i = 0; i < sueldos[0].length; i++) {
            sueldos[2][i] = sueldos[0][i] - sueldos[1][i];
        }
    }

    public static void imprimir() {
        DecimalFormat df = new DecimalFormat("###,###.##");
        for (int i = 0; i < sueldos[0].length; i++) {
            String sueldo = "SB: " + df.format(sueldos[0][i]) + " - R: " +
df.format(sueldos[1][i]) + " - SN: " + df.format(sueldos[2][i]);
            System.out.println("Sueldo del empleado " + i + ": " + sueldo);
        }
    }
}
```

```

    }
}

```

Cadenas de caracteres

- Haciendo uso de la clase **String** hacer los siguientes ejercicios:
 1. Escribir un programa que pida una palabra y un entero n y vaya rotando el carácter inicial de la palabra, al final de la misma, tantas veces como indique n. (Por ejemplo, “monja”,3 debe devolver “jamon” y “monja”;5 “monja”).
 2. Escribir un programa que solicite una cadena y una letra y nos devuelva las posiciones que ocupa esa letra en la cadena.
 3. Escribir un programa que nos diga cuantas veces se repiten cada una de las vocales en una cadena que el usuario introduce por teclado.
 4. Escribir un programa que reciba un NIF con 9 caracteres (ej. 00395469F) y nos diga si la letra es correcta.
- Haciendo uso de la clase **StringTokenizer** hacer los siguientes ejercicios:
 1. Escribir un programa al que se le pasa un número de teléfono de la forma: 91-8885566. El programa deberá usar la clase StringTokenizer para extraer el código de la comunidad y el resto del número, convertir el código de la comunidad en int y el resto en long, y presentarlo por pantalla.
 2. Escribir un programa que lea una frase y nos diga cuántas palabras la componen.
 3. Escribir un programa que pida una cadena y la divida en palabras y presente las palabras en orden inverso. Consejo: Utilizar la clase **StringBuffer**.
- Haciendo uso del método **split** de la clase **String** repetir ejercicios de la clase StringTokenizer.

```

/*
Escribir un programa que pida una palabra y un entero n y vaya rotando el carácter inicial
de la palabra,
al final de la misma, tantas veces como indique n.
*/
import java.io.*;
public class Ses3Ejercicio1 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee una palabra o frase (Intro para salir): ");
            String p=entrada.readLine();
            if (p.equals("")) return;
            System.out.println("\nTeclee el número de posiciones a rotar: ");
            int n = Integer.parseInt(entrada.readLine());
            System.out.println(p + " rota " + n + " --> " + rotar(p,n));
        }
    }

    public static String rotar(String s, int n) {
        for (int i = 0; i < n; i++) {
            s=s.substring(1).concat(s.substring(0, 1));
        }
        return s;
    }
}

```

```

/*
Escribir un programa que solicite una cadena y una letra y nos devuelva las posiciones que
ocupa esa letra en la cadena
*/
import java.io.*;
public class Ses3Ejercicio2 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
    }
}

```

```

        while (true) {
            System.out.println("\nTeclee una palabra o frase (Intro para salir): ");
            String palabra = entrada.readLine();
            if (palabra.equals("")) return;
            System.out.println("\nTeclee la letra a buscar: ");
            char letra = entrada.readLine().charAt(0);
            System.out.print("La letra "+letra+" se encuentra en las posiciones: ");
            for (int i=0; i<palabra.length(); i++){
                if (palabra.charAt(i)==letra) System.out.print(i+1+", ");
            }
        }
    }
}

```

```

/*
Escribir un programa que nos diga cuantas veces se repiten
cada una de las vocales en una cadena que el usuario introduce por teclado.
*/
import java.io.*;
public class Ses3Ejercicio3 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee una palabra o frase (Intro para salir): ");
            String palabra = entrada.readLine().toLowerCase();
            if (palabra.equals("")) return;
            System.out.println("Vocal a --> " + numVocal(palabra,'a') + " veces");
            System.out.println("Vocal e --> " + numVocal(palabra,'e') + " veces");
            System.out.println("Vocal i --> " + numVocal(palabra,'i') + " veces");
            System.out.println("Vocal o --> " + numVocal(palabra,'o') + " veces");
            System.out.println("Vocal u --> " + numVocal(palabra,'u') + " veces");
        }
        public static int numVocal(String s, char v){
            int veces=0;
            for (int i=0; i<s.length();i++) if (s.charAt(i)==v) veces++;
            return veces;
        }
    }
}

```

```

/*
Escribir un programa que reciba un NIF con 9 caracteres (ej. 00395469F) y nos diga si la
letra es correcta
*/
import java.io.*;
public class Ses3CadEjercicio4 {
    public static void main(String[] args) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        System.out.println("Introduce un NIF:");
        String nif = entrada.readLine().toUpperCase();
        char letraNif = nif.charAt(8);
        char letraCalculada = calculaLetraNIF(Integer.parseInt(nif.substring(0, 8)));
        if (letraNif == letraCalculada) {
            System.out.println("Letra correcta.");
        } else {
            System.out.println("Letra incorrecta.");
        }
    }
    public static char calculaLetraNIF(int dninum) {
        char letras[] = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X', 'B', 'N',
            'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'};
        int pos = dninum % 23;
        return letras[pos];
    }
}

```

```

/*
Escribir un programa al que se le pasa un número de teléfono de la forma: 91-8885566.
El programa deberá usar la clase StringTokenizer para extraer el prefijo y el resto del
número,

```



```

convertir el prefijo en int y el resto en long, y presentarlo por pantalla.
*/
import java.io.*;
import java.util.*;
public class Ses3Ejercicio5 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee un número de telefono en forma 91-8885566 (Intro
para salir): ");
            String p = entrada.readLine().toLowerCase();
            if (p.equals("")) return;
            StringTokenizer tfno = new StringTokenizer(p, "-");
            int prefijo=Integer.parseInt(tfno.nextToken());
            long resto=Long.parseLong(tfno.nextToken());
            System.out.println("Prefijo: " + prefijo + " resto: " + resto);
        }
    }
}

/* Método split */
import java.io.*;
public class Ses3Ejercicio5Split {
    public static void main(String[] args) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee un número de telefono en forma 91-8885566 (Intro
para salir): ");
            String p = entrada.readLine().toLowerCase();
            if (p.equals("")) {
                return;
            }
            String num[] = p.split("-");
            int prefijo = Integer.parseInt(num[0]);
            long resto = Long.parseLong(num[1]);
            System.out.println("Prefijo: " + prefijo + " resto: " + resto);
        }
    }
}

```

```

/*
Escribir un programa que lea una frase y nos diga cuántas palabras la componen.
*/
import java.io.*;
import java.util.*;
public class Ses3Ejercicio6 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee una frase (Intro para salir): ");
            String frase = entrada.readLine();
            if (frase.equals("")) return;
            StringTokenizer pal = new StringTokenizer(frase, " ");
            System.out.println("Número de palabras: " + pal.countTokens());
            // Podemos imprimirlas
            while(pal.hasMoreTokens()){
                System.out.println(pal.nextToken());
            }
        }
    }
}

/* Método split */
public class Ses3Ejercicio6Split {
    public static void main(String[] args) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee una frase (Intro para salir): ");
            String frase = entrada.readLine();
            if (frase.equals("")) {
                return;
            }
            String palabras[] = frase.split(" ");

```

```

        System.out.println("Número de palabras: " + palabras.length);
        // Podemos imprimirlas
        for (int i = 0; i < palabras.length; i++) {
            System.out.println(palabras[i]);
        }
    }
}

```

/*
Escribir un programa que pida una cadena y la divida en palabras y presente las palabras en orden inverso.

Consejo: Utilizar la clase StringBuilder

```

*/
import java.io.*;
import java.util.*;
public class Ses3Ejercicio7 {
    public static void main(String args[]) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (true) {
            System.out.println("\nTeclee una frase (Intro para salir): ");
            String frase = entrada.readLine();
            if (frase.equals("")) return;
            StringTokenizer pal = new StringTokenizer(frase, " ");
            StringBuilder fraseReves = new StringBuilder();
            while(pal.hasMoreTokens()){
                fraseReves.insert(0, " ");
                fraseReves.insert(0,pal.nextToken());
            }
            System.out.println(fraseReves);
        }
    }
}

```

/* Método split */

```

import java.io.*;
public class Ses3Ejercicio7Split {

    public static void main(String[] args) throws IOException {
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));

        while (true) {
            System.out.println("\nTeclee una frase (Intro para salir): ");
            String frase = entrada.readLine();
            if (frase.equals("")) {
                return;
            }
            String palabras[] = frase.split(" ");
            StringBuilder fraseReves = new StringBuilder();
            for (int i = 0; i < palabras.length; i++) {
                fraseReves.insert(0, palabras[i] + " ");
            }

            System.out.println(fraseReves);
        }
    }
}

```