

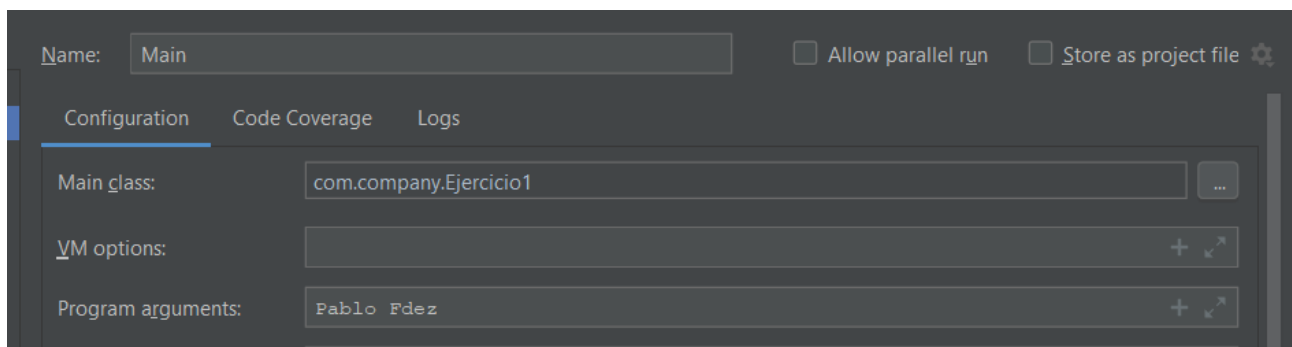
(Para ejecutar las clases definir como principal cuando se quiera ejecutar, de esta forma el main se ejecuta como principal independientemente del resto de clases.)

EJERCICIO 1

Realiza un programa en Java, que introduzca como argumentos el nombre y apellidos de alguien y posteriormente muestre el siguiente mensaje:

Hola nombre apellidos

```
public class Ejercicio1 {  
    public static void main(String[] args) {  
        System.out.println("Hello " + args[0] + " " + args[1]);  
    }  
}
```



- Según el IDE que estemos utilizando puede variar. En IntelliJ Idea en “Run/Debug Configurations” nos saldrá esa pestaña. En Main class definimos la clase Ejercicio1 para que ejecute el main como el principal. Pasamos los argumentos Pablo (args[0]) Fdez (args[1])

```
"C:\Program Files\Java\jdk1.8.0_  
Hello Pablo Fdez
```

EJERCICIO 2

Busca información sobre la clase Scanner, usada para leer datos desde el teclado o desde un fichero y explica que realizan los siguientes métodos:

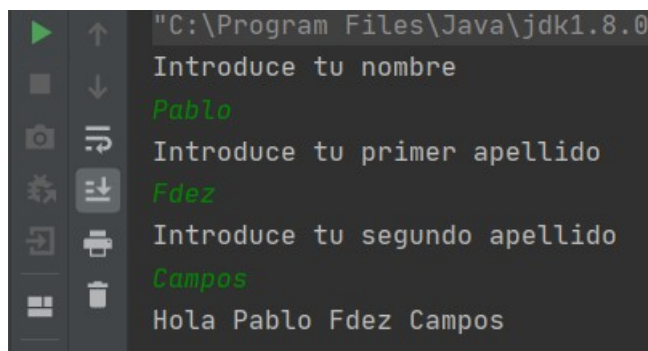
Scanner(System.in)
hasNext()
hasNextInt()
hasNextDouble()
next()
nextInt()
nextDouble()

- Todo explicado a través de comentarios en la clase Ejercicio2.
- Se puede ejecutar y ver los ejemplos explicados.

EJERCICIO 3

Realiza un programa en Java, que lea desde la entrada estándar tu nombre y apellidos y posteriormente presente por pantalla el siguiente mensaje:
Hola nombre apellidos

```
public class Ejercicio3 {  
    public static void main (String[] args){  
        System.out.println("Introduce tu nombre");  
        Scanner sc = new Scanner(System.in);  
        String name = sc.next();  
        System.out.println("Introduce tu primer apellido");  
        Scanner entrada = new Scanner(System.in);  
        String surname1 = entrada.next();  
        System.out.println("Introduce tu segundo apellido");  
        Scanner entrada2 = new Scanner(System.in);  
        String surname2 = entrada2.next();  
        System.out.println("Hola " + name + " " + surname1 + " " + surname2);  
    }  
}
```



EJERCICIO 4

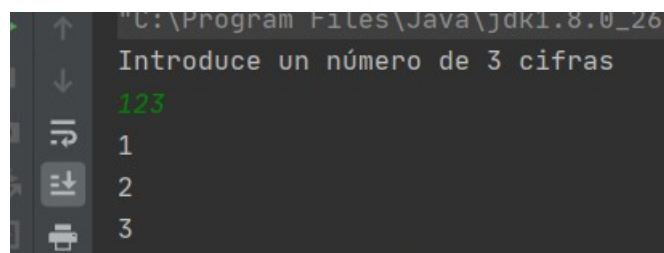
Realiza un programa en Java, que lea desde la entrada (teclado o argumentos) un número entero de tres cifras y posteriormente muestre sus cifras por separado. Haz todo en un método main de una clase denominada Cifras. Ejemplo para el número 457:

Cifra1 4

Cifra2 5

Cifra3 7

```
public class Ejercicio4 {  
    public static void main (String[] args){  
        int third, second, first;  
        System.out.println("Introduce un número de 3 cifras");  
        Scanner sc = new Scanner(System.in);  
        int number = sc.nextInt();  
        third = number % 10;  
        number = number / 10;  
        second = number % 10;  
        number = number / 10;  
        first = number % 10;  
        System.out.printf("%d\n%d\n%d", first, second, third);  
    }  
}
```



```
C:\Program Files\Java\jdk1.8.0_261  
Introduce un número de 3 cifras  
123  
1  
2  
3
```

EJERCICIO 5 – Clase Time

Programa que se le pase como argumentos tres números enteros H, M, S que contienen hora, minutos y segundos respectivamente. Comprueba si la hora que indican es una hora válida.

Ejemplo de ejecución:

java Tiempo 22 10 15

Hora válida

java Tiempo 25 10 15

Hora no válida

```
import java.util.Scanner;
public class Time {
    public static void main(String[] args){
        int H, M, S;
        System.out.println("Introduce la hora (0-23)");
        Scanner sc = new Scanner(System.in);
        H = sc.nextInt();
        System.out.println("Introduce los minutos (0-59)");
        Scanner scanner = new Scanner(System.in);
        M = scanner.nextInt();
        System.out.println("Introduce los segundos (0-59)");
        Scanner scan = new Scanner(System.in);
        S = scan.nextInt();
        if(H < 0 || H > 23 || M < 0 || M > 59 || S < 0 || S > 59) {
            System.out.println("Hora introducida no válida");
        } else {
            System.out.println("Hora introducida válida: " + H + " " + M + " " + S);
        }
    }
}
```

```
"C:\Program Files\Java\jdk1.8.0_261\
Introduce la hora (0-23)
222
Introduce los minutos (0-59)
22
Introduce los segundos (0-59)
2
Hora introducida no válida
```

```
"C:\Program Files\Java\jdk1.8.0_261\bin\jav
Introduce la hora (0-23)
22
Introduce los minutos (0-59)
45
Introduce los segundos (0-59)
23
Hora introducida válida: 22 45 23
```

EJERCICIO 6

Programa que solicite un número y un número de columnas, de manera que imprima desde el 1 al número solicitado, en el número de columnas pedido.

Ejemplo:

Introduce nº de columnas: 3

Introduce un nº: 15

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

Otro ejemplo:

Introduce nº de columnas: 7

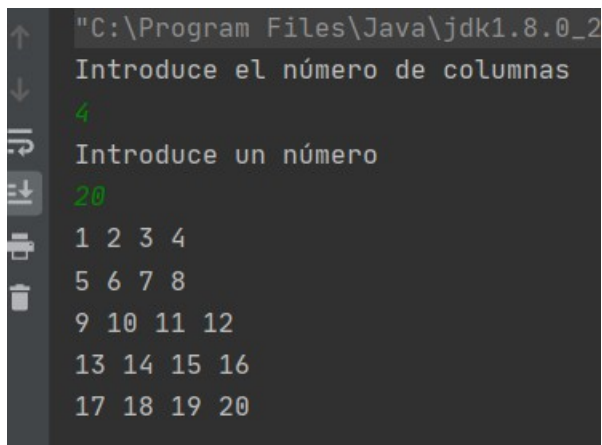
Introduce un nº: 100

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56
57	58	59	60	61	62	63
64	65	66	67	68	69	70
71	72	73	74	75	76	77
78	79	80	81	82	83	84
85	86	87	88	89	90	91
92	93	94	95	96	97	98
99	100					

```

public class Ejercicio6 {
    public static void main (String[] args){
        System.out.println("Introduce el número de columnas");
        Scanner sc = new Scanner(System.in);
        int column = sc.nextInt();
        System.out.println("Introduce un número");
        Scanner scanner = new Scanner(System.in);
        int number = scanner.nextInt();
        for (int i = 1; i <= number;){
            int matrix [][] = new int [1][column];
            for (int j = 0; j < matrix.length; j++){
                for (int k = 0; k < matrix[j].length; k++){
                    if(i <= number){
                        matrix[j][k] = i;
                        System.out.print(matrix[j][k] + " ");
                    }
                    i++;
                }
                System.out.println(" ");
            }
        }
    }
}

```



```

"C:\Program Files\Java\jdk1.8.0_2
Introduce el número de columnas
4
Introduce un número
20
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16
17 18 19 20

```

EJERCICIO 7 – Clase TestEquation

Los sistemas de ecuaciones lineales, en el caso de dos ecuaciones con dos incógnitas se pueden representar de forma genérica de la siguiente forma:

Un sistema de ecuaciones tiene solución si:

La solución de un sistema de dos ecuaciones con dos incógnitas, en el caso que lo tenga, viene dado por la siguiente regla, derivada de la regla de Cramer:

Se quiere realizar una clase denomina Ecuacion que contenga dos métodos:

Un método boolean que devuelva verdadero o falso si el sistema es resoluble o no

Dos métodos que devuelvan el valor de x e y. Usa double como tipos, al menos en los valores de x e y

Los argumentos en los métodos serán los coeficientes que acompañan a las incógnitas, además del término independiente (a, b, c, . . .)

Crea una clase TestEcuacion, con el método main y que resuelva las siguientes ecuaciones:

Debe indicar por pantalla, si el sistema es resoluble o no. Y en el caso que sea resoluble debe mostrar la resolución de dicho sistema de ecuaciones.

```

import java.util.Scanner;
class Number {
    public static boolean check(int a, int b, int c, int d){
        boolean operation;
        if ((a * d) - (b * c) != 0){
            operation = true;
        } else {
            operation = false;
        }
        return operation;
    }
    public static double solutionX(double a, double b, double c, double d,
double e, double f){
        double numerator = (e * d) - (b * f);
        double denominator = (a * d) - (b * c);
        double x = numerator / denominator;
        return x;
    }
    public static double solutionY(double a, double b, double c, double d,
double e, double f){
        double numerator = (a * f) - (e * c);
        double denominator = (a * d) - (b * c);
        double y = numerator / denominator;
        return y;
    }
}
public class TestEquation {
    public static void main(String[] args) {
        System.out.print("Con este programa puedes a resolver  ecuaciones
lineal: \n");
        System.out.print("ax + by = e \n");
        System.out.print("cx + dy = f \n");
        if (Number.check(1, 1, 2, 2) == true) {
            int i = 0;
            do {
                i++;
                System.out.println("x = " + Number.solutionX(1,1,2,2,1,2));
                System.out.println("y = " + Number.solutionY(2,1,-1,2, 7,-1));
            } while (i <= 1);
        } else {
            System.out.println("No tiene solución");
        }
    }
}

```

- Pasamos como parametros los valores de los coeficientes.

```

"C:\Program Files\Java\jdk1.8.0_261\bin\java.exe" ...
Con este programa puedes a resolver  ecuaciones lineal:
ax + by = e
cx + dy = f
No tiene solución

```


- Está subido en github el proyecto por si le resulta más fácil la corrección. En la rama master solo está el README.md y el pdf, en la rama main se encuentran todos los java.
Url : <https://github.com/pabofdezcampos/APIs-Java/tree/main>