# INSTALACION VISUAL STUDIO CODE

Datos del profe:

Paco Garcia Serrano

Usuario de Github: serraguti

Email de contacto: pacoserranox@gmail.com

Descargamos Visual Studio Code

<https://code.visualstudio.com/>

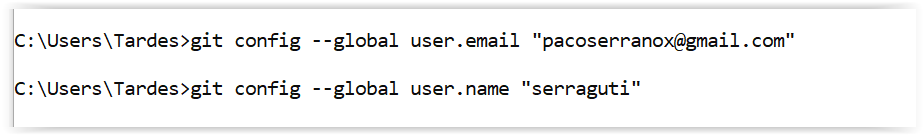
Instalamos Git for Windows

<https://git-scm.com/downloads/win>

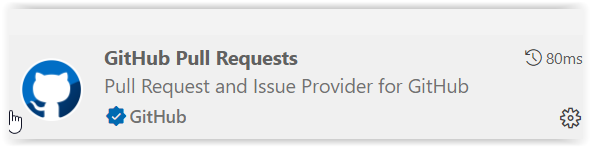
Incluimos, dentro de la línea de comandos (cmd) el usuario y email de Github

git config --global user.email "tuemail@example.com"

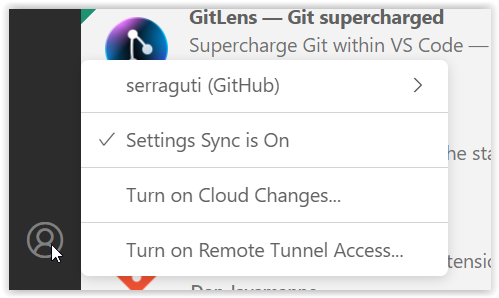
git config --global user.name "usuariogithub"



Dentro de Visual Studio, instalamos la Extensión de GitHub



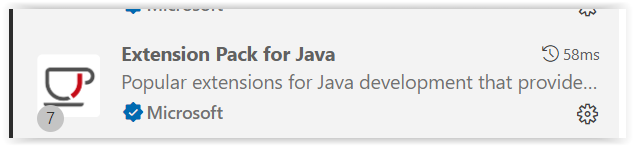
Nos validamos con nuestro usuario de Github dentro de Visual Studio



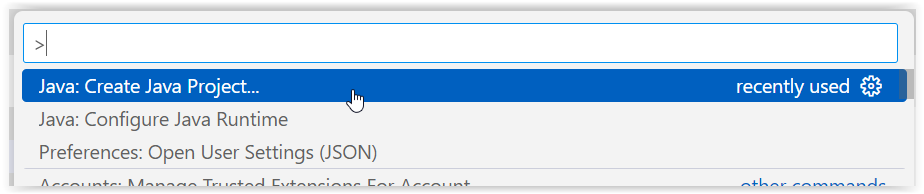
Pulsamos sobre **Backup and Sync**

Tendremos todo listo en casa para trabajar.

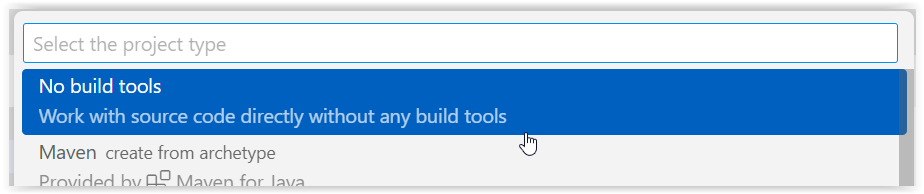
Por si algo falla, instalamos la extensión Java Extension Pack



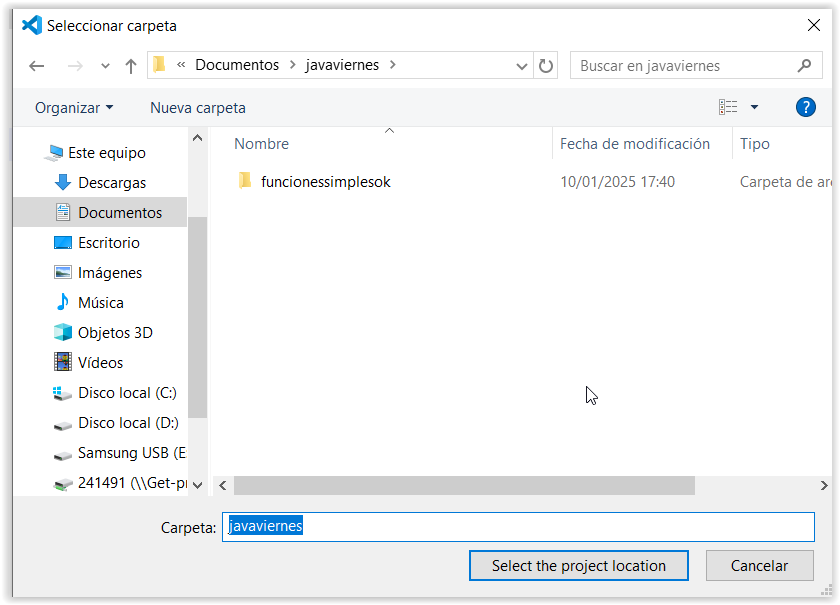
Para crear un nuevo proyecto de Java dentro de VS Code, debemos utilizar la siguiente combinación de Teclas: **CONTROL + SHIFT + P**

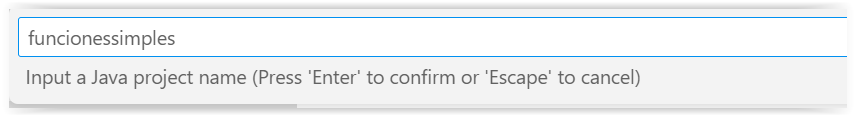


Seleccionamos **No Build Tools**

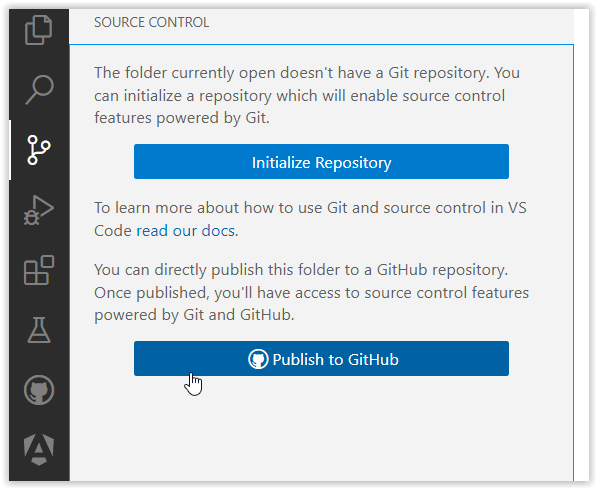


Indicamos la ubicación de nuestro proyecto nuevo.



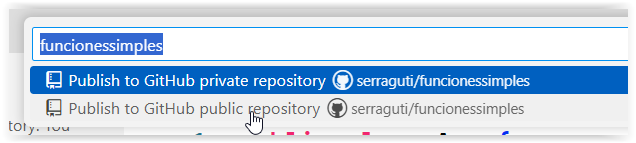


Para sincronizar con Github en Visual Studio, seleccionamos **Source Control**



Pulsamos sobre **Publish to GitHub**

Nos pide nombre del proyecto y visibilidad



Indicamos las carpetas que deseamos sincronizar



Existe un sistema de teclas para poder comentar código de forma automática dentro de Visual Studio Code.

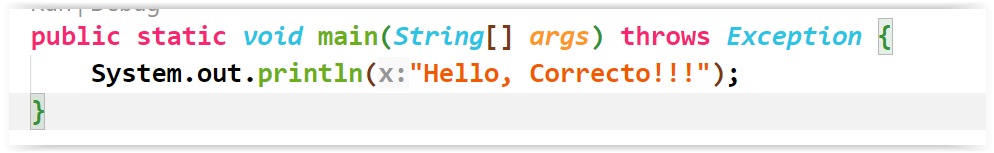
Para comentar código: Control + k + c

Descomentar: Control + k + u

**FUNCIONES DENTRO DE JAVA**

Llevamos utilizando funciones de Java desde el primer día.

Una función es un código que se ejecuta en una clase.



Las funciones nos permiten poder “dividir” códigos en funcionalidades para poder reutilizar el código desde múltiples puntos.

Las funciones no tienen que estar de forma obligatoria en la misma clase, podemos tener múltiples clases con múltiples funciones.

Actualmente, vamos a trabajar con funciones de tipo **static**

Una función **static** es una función que se llama directamente desde la clase.

Existen funciones de objeto y funciones de clase (**static**).

Dentro de Java podemos crear objetos para trabajar con ellos.

Dichos objetos nos sirven para representar elementos personalizados.

Ejemplo: Actualmente estamos en el momento de “utilizar” los objetos.

Si yo quiero guardar un texto: String texto = “soy un texto”;

Si yo quiero guardar un número: int numero = 77;

Si yo quiero guardar una Persona?? (Nombre, Apellidos y Edad??)

Cuando tenemos un objeto, podemos incluirle propiedades y funciones.

Una propiedad es una característica de un objeto

Una función es una acción de un objeto.

Persona persona;

persona.nombre = “Alumno”;

persona.edad = 25;

persona.**comprobarEdad()**; //este método apunta al objeto Persona.

Quiero poder validar cualquier DNI de cualquier persona.

Elementos necesarios. Un código para validar un DNI

Dicho código debe estar en algún sitio (Clase)

Validaciones.**validarDni(DNI)**;

Validaciones.**validarMatricula(matricula)**;

Una función es un conjunto de instrucciones.

Dentro de una función tenemos dos opciones de trabajar con funciones:

1. **void:** Estas funciones son un conjunto de instrucciones

Sintaxis:

public static void nombreFuncion()

{

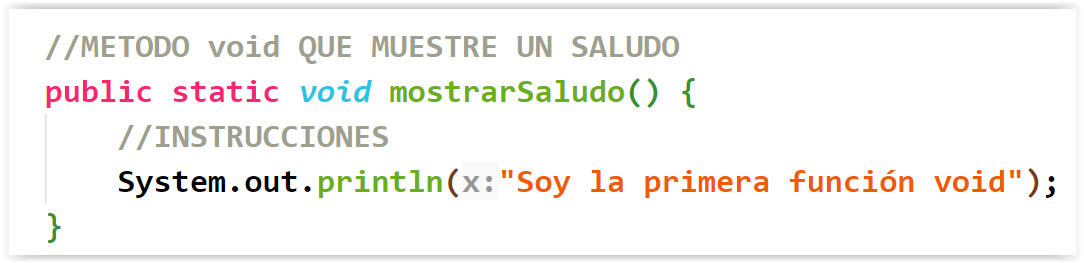
//INSTRUCCIONES

}

Debemos escribir siempre en la clase **App**

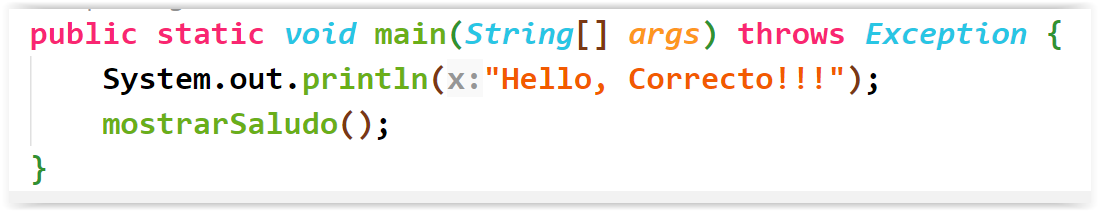
Esto es la declaración de una función **void**

Una función NO sirve de nada si no la llamamos.



Para llamar a una función **void** debemos hacerlo desde otra función (**main**)

La llamada a una función **static** DENTRO de la misma clase, simplemente se escribe el nombre de la función para la llamada con los paréntesis.



1. **return:** Sonfunciones con un conjunto de instrucciones y que devuelven un valor cuando se han ejecutado.

public static TIPODATO nombreFuncion()

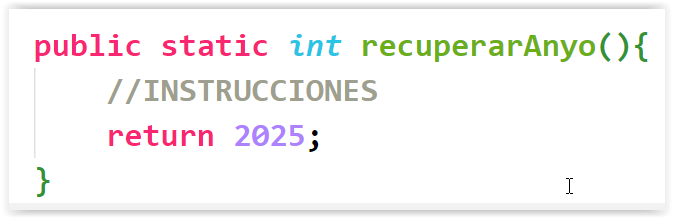
{

//INSTRUCCIONES

return VALOR\_DEL\_TIPO\_DATO;

}

Vamos a realizar una función que nos devolverá un número (2025).



La llamada se realiza igual pero se pierde en el olvido…

En la llamada a una función **return** debemos almacenar el valor que devuelve en la petición a la función.



**PARAMETROS EN LAS FUNCIONES/METODOS**

Una función puede recibir parámetro o parámetros para su ejecución.

Los parámetros son enviados entre los paréntesis de la declaración de la función y estarán separados por comas.

Los parámetros NO son opcionales, es decir, si declaramos una función con parámetros **debemos** enviar dichos parámetros en la llamada.

Es la misma sintaxis para todas las funciones (void, return)

public static void metodoParametros

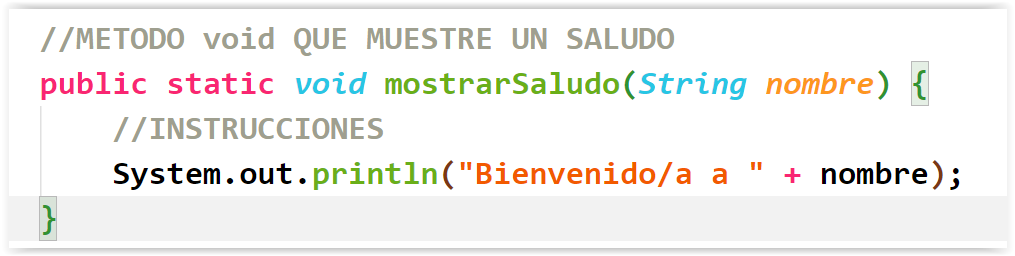
(**tipoDato nombreParam1, tipoDato nombreParam2**)

{

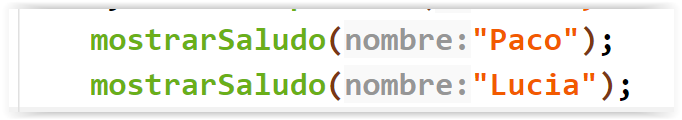
//INSTRUCCIONES

}

Por ejemplo, vamos a modificar el método **mostrarSaludo()** recibiendo un nombre para saludar.

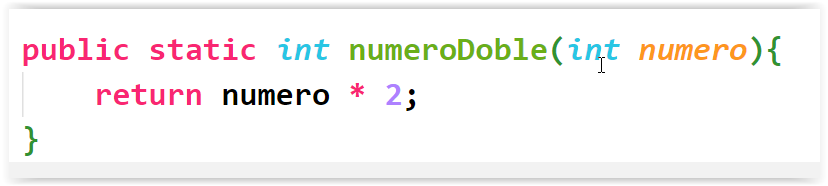


En la llamada debemos enviar el valor para los parámetros

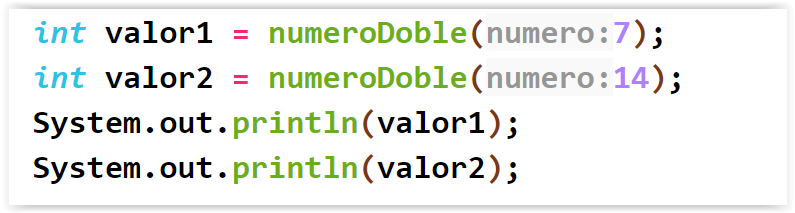


Lo mismo sucede si hacemos un método/función **return**

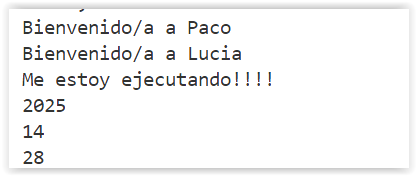
Vamos a realizar un método que devuelva el doble de un número que recibamos.



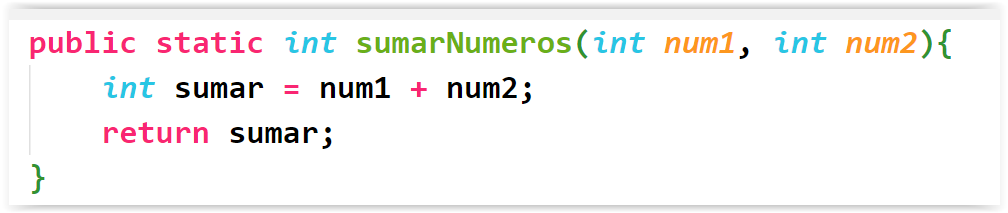
Las llamadas ya son dinámicas dependiendo del número



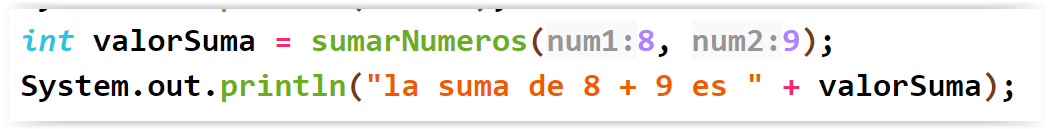
Salida del programa



Por ejemplo, vamos a realizar una función para devolver la suma de dos números.

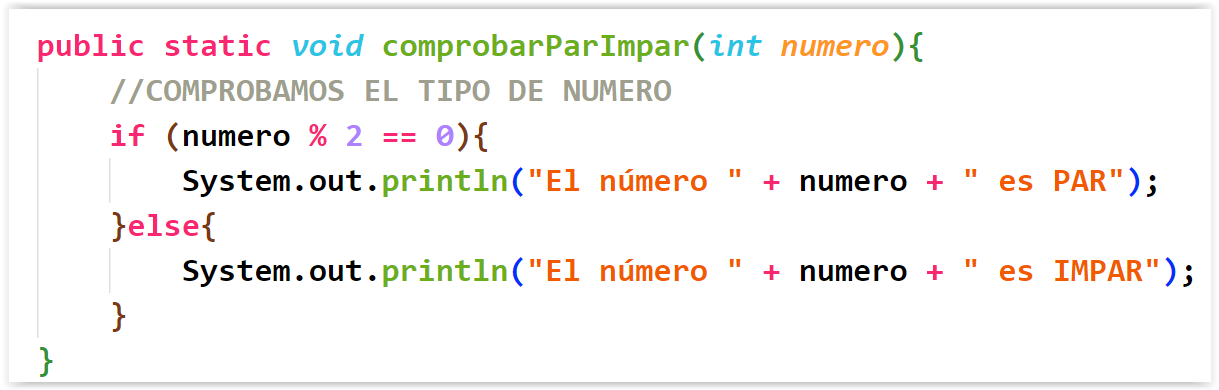


Llamada al método:

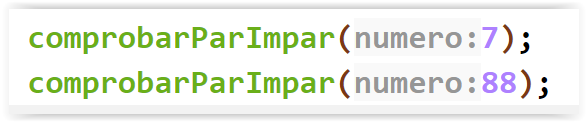


Vamos a realizar una función/método que recibirá un número y **mostrará** si el número es par o impar.

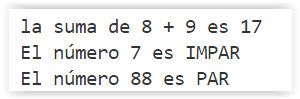
El símbolo para el resto en Java es **%**



Realizamos la llamada:



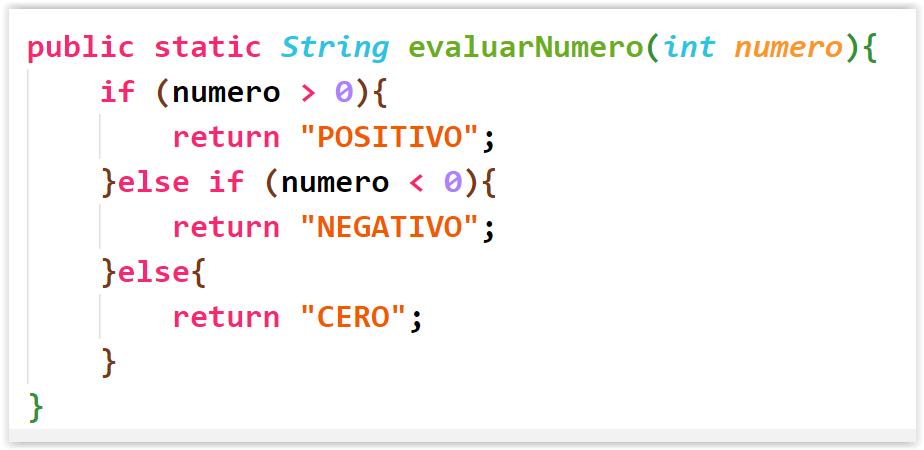
Y veremos el resultado en pantalla



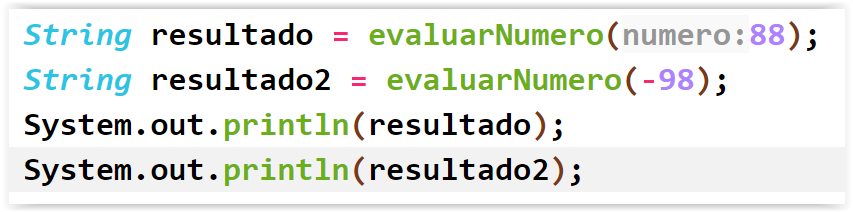
Ejemplo:

Creamos una nueva función (en la misma clase) que recibirá un número.

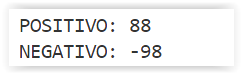
Debemos **devolver** si el número recibido es POSITIVO, NEGATIVO o CERO



Realizamos la llamada

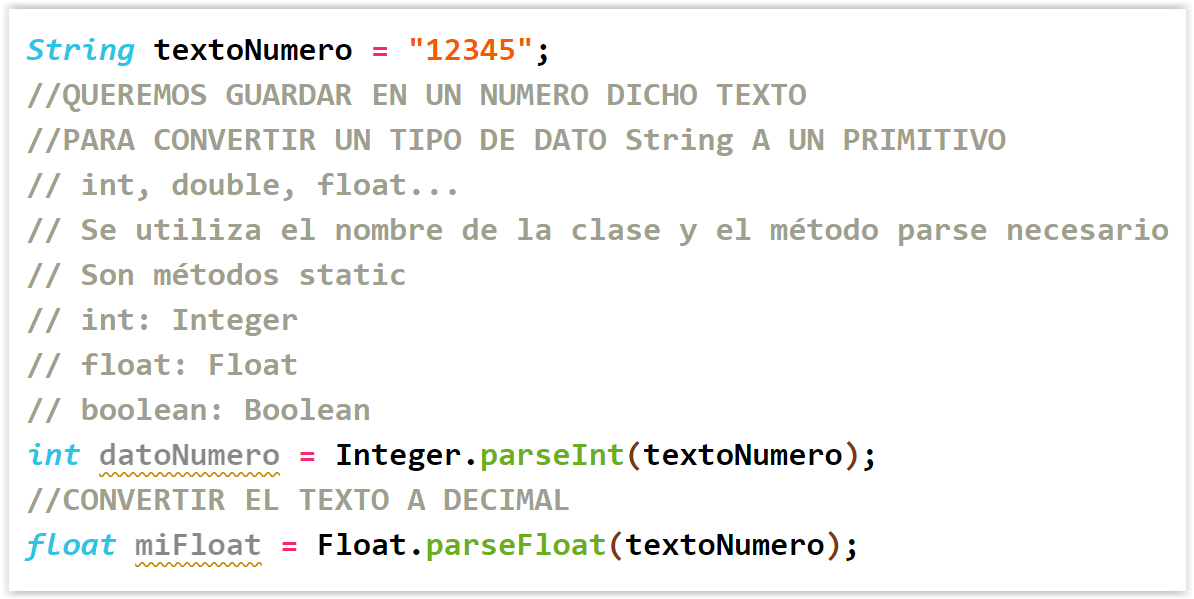


Resultado de la ejecución



Sabemos pedir datos por teclado? **Scanner?**

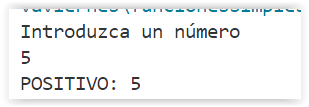
Ejemplo de conversión: Convertir un texto de tipo numérico a int



Vamos a llevarnos ahora mismo el código que tenemos de **evaluarNumero()** a una nueva clase.

Lo que haremos será pedir al usuario por teclado (Scanner) el número que quiera evaluar.

Sobre la carpeta **src** le indicamos con el botón derecho **New Java File 🡪 Class** y creamos una clase llamada **FinViernes**





**CODIGO JAVA**

**import java.util.Scanner;**

**public class FinViernes {**

**//psvm**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca un número");**

**//CAPTURAMOS EL String numérico que el usuario ha introducido**

***String* dato = teclado.nextLine();**

**//CONVERTIMOS EL TEXTO A NUMERO**

***int* numero = Integer.parseInt(dato);**

**//LLAMAMOS A EVALUAR NUMERO**

***String* respuesta = evaluarNumero(numero);**

**System.out.println(respuesta);**

**}**

**public static *String* evaluarNumero(*int* *numero*){**

**if (numero > 0){**

**return "POSITIVO: " + numero;**

**}else if (numero < 0){**

**return "NEGATIVO: " + numero;**

**}else{**

**return "CERO: " + numero;**

**}**

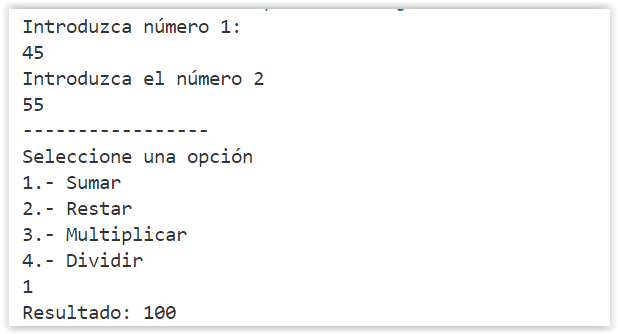
**}**

**}**

Vamos a realizar, en una nueva clase llamada **InicioLunes** un método que recibirá un número y nos devolverá si dicho número es POSITIVO, NEGATIVO o CERO.

Dicha clase será una clase ejecutable, es decir, una clase con **main()**

Creamos una clase llamada **Ejecución** que tendrá un método **main()** que pedirá dos números al Usuario



Tendremos otra clase llamada **Calculadora** que tendrá un menú y las operaciones de Sumar, Restar, Multiplicar y dividir.

**CALCULADORA**

**public class Calculadora {**

**//ESTA CLASE ES UNA HERRAMIENTA**

**//POR SI MISMA, NO SIRVE DE NADA**

**//TENDRA UN METODO/S PARA HACER OPERACIONES**

**//MATEMATICAS**

**public static *void* mostrarMenu(){**

**System.out.println("-----------------");**

**System.out.println("Seleccione una opción");**

**System.out.println("1.- Sumar");**

**System.out.println("2.- Restar");**

**System.out.println("3.- Multiplicar");**

**System.out.println("4.- Dividir");**

**}**

**public static *int* sumarNumeros(*int* *num1*, *int* *num2*){**

***int* operacion = num1 + num2;**

**return operacion;**

**}**

**public static *int* restarNumeros(*int* *num1*, *int* *num2*){**

***int* operacion = num1 - num2;**

**return operacion;**

**}**

**public static *int* dividirNumeros(*int* *num1*, *int* *num2*){**

***int* operacion = num1 / num2;**

**return operacion;**

**}**

**public static *int* multiplicarNumeros(*int* *num1*, *int* *num2*){**

***int* operacion = num1 \* num2;**

**return operacion;**

**}**

**}**

**EJECUCION**

**import java.util.Scanner;**

**public class Ejecucion {**

**public static *void* main(*String*[] *args*) {**

**//CREAMOS EL TECLADO**

***Scanner* teclado = new Scanner(System.in);**

**//MENSAJES PARA PEDIR NUMEROS**

**System.out.println("Introduzca número 1:");**

**//ALMACENAMOS EL DATO QUE HA ESCRITO**

***String* entrada = teclado.nextLine();**

**//ALMACENAMOS EL NUMERO 1**

***int* numero1 = Integer.parseInt(entrada);**

**System.out.println("Introduzca el número 2");**

**//ALMACENAMOS DE NUEVO LO QUE HA ESCRITO EL USUARIO**

**entrada = teclado.nextLine();**

**//ALMACENAMOS EL NUMERO 2**

***int* numero2 = Integer.parseInt(entrada);**

**//CREAMOS UN MENU PARA LAS OPCIONES DEL USUARIO**

**Calculadora.mostrarMenu();**

**//CAPTURAMOS LO QUE HA ESCRITO EL USUARIO**

**entrada = teclado.nextLine();**

**//CAPTURAMOS LA OPCION SELECCIONADA**

***int* opcion = Integer.parseInt(entrada);**

***int* resultado = 0;**

**if (opcion == 1){**

**resultado = Calculadora.sumarNumeros(numero1, numero2);**

**}else if (opcion == 2){**

**resultado = Calculadora.restarNumeros(numero1, numero2);**

**}else if (opcion == 3){**

**resultado = Calculadora.multiplicarNumeros(numero1, numero2);**

**}else if (opcion == 4){**

**resultado = Calculadora.dividirNumeros(numero1, numero2);**

**}**

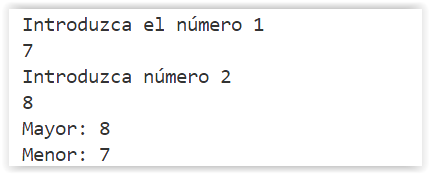
**System.out.println("Resultado: " + resultado);**

**}**

**}**

Vamos a tener dos clases.

1. **EvaluacionNumeros**: Esta clase tendrá dos métodos
   1. **mayorDosNumeros**(num1, num2): Devolverá el mayor de dos números
   2. **menorDosNumeros**(num1, num2): Devolverá el menor de dos números.
   3. **evaluarTresNumeros(num1, num2, num3):** Método que recibirá tres números y nos **mostrará** el número mayor, el número menor y el intermedio.
2. **EjecucionNumeros**: Esta clase tendrá el método **main()** dónde pediremos dos números y llamaremos a los métodos de la clase **EvaluacionNumeros**.



**EVALUACIONNUMEROS**

**public class EvaluacionNumeros {**

**public static *int* mayorDosNumeros(*int* *num1*, *int* *num2*){**

**if (num1 > num2){**

**return num1;**

**}else{**

**return num2;**

**}**

**}**

**public static *int* menorDosNumeros(*int* *num1*, *int* *num2*){**

**if (num1 < num2) {**

**return num1;**

**}else{**

**return num2;**

**}**

**}**

**}**

**EJECUCIONNUMEROS**

**import java.util.Scanner;**

**public class EjecucionNumeros {**

**public static *void* main(*String*[] *args*) {**

**//CREAMOS EL OBJETO TECLADO**

***Scanner* teclado = new Scanner(System.in);**

**//MENSAJES PARA PEDIR DATOS**

**System.out.println("Introduzca el número 1");**

**//CAPTURAMOS LO QUE EL USUARIO HA ESCRITO**

***String* entrada = teclado.nextLine();**

**//CAPTURAMOS EL PRIMER NUMERO**

***int* numero1 = Integer.parseInt(entrada);**

**System.out.println("Introduzca número 2");**

**entrada = teclado.nextLine();**

***int* numero2 = Integer.parseInt(entrada);**

***int* mayor = EvaluacionNumeros.mayorDosNumeros(numero1, numero2);**

***int* menor = EvaluacionNumeros.menorDosNumeros(numero1, numero2);**

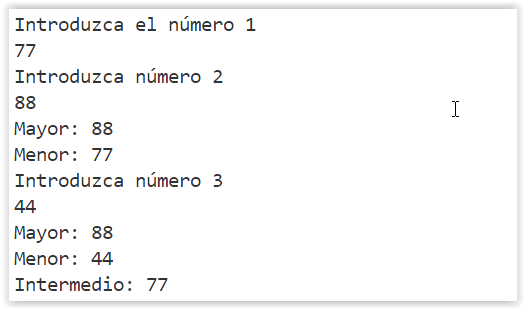
**System.out.println("Mayor: " + mayor);**

**System.out.println("Menor: " + menor);**

**}**

**}**

A continuación, incluimos un nuevo método dentro de **EvaluacionNumeros**



**public static *void* evaluarTresNumeros(*int* *num1*, *int* *num2*, *int* *num3*){**

***int* mayor, menor, intermedio;**

**if (num1 >= num2 && num1 >= num3){**

**mayor = num1;**

**}else if (num2 >= num1 && num2 >= num3){**

**mayor = num2;**

**}else {**

**mayor = num3;**

**}**

**if (num1 <= num2 && num1 <= num3){**

**menor = num1;**

**}else if (num2 <= num1 && num2 <= num3){**

**menor = num2;**

**}else{**

**menor = num3;**

**}**

***int* suma = num1 + num2 + num3;**

**intermedio = suma - mayor - menor;**

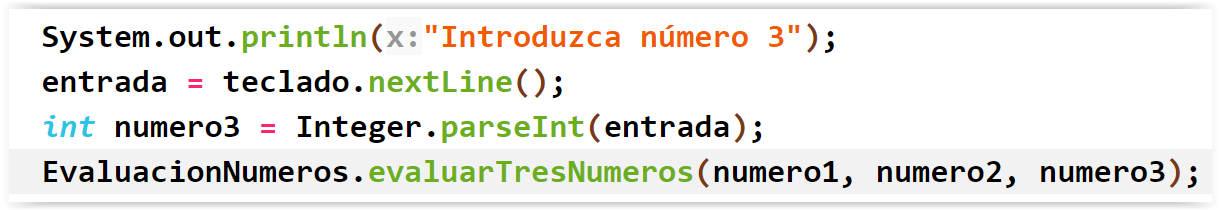
**System.out.println("Mayor: " + mayor);**

**System.out.println("Menor: " + menor);**

**System.out.println("Intermedio: " + intermedio);**

**}**

Realizamos la llamada desde **EjecucionNumeros**



En el siguiente ejemplo vamos a realizar una clase que contendrá **tres** métodos.

En estos **tres** métodos vamos a devolver información sobre algo de Personas.

Vamos a crear dos clases:

* **EjecucionPersonas (main)**
* **HerramientasPersonas**

Tendremos un método que nos devolverá la letra del DNI a partir de el número.

* **getLetraNIF(int numeroDni)**
* **validarNIF(String dni)**
* Realizar una aplicación para conocer la letra del Documento Nacional de Identidad a través del número de DNI.
* Pediremos al usuario su número de DNI sin la letra.
* Mostraremos la letra de dicho número.
* La fórmula para calcular la letra del número del DNI se halla de la siguiente manera:
* Se calcula el valor de la siguiente ecuación

( nº DNI - ((nº DNI / 23) \* 23))

Se mira la equivalencia en la siguiente tabla

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0=T | 4=G | 8=P | 12=N | 16=Q | 20=C |
| 1=R | 5=M | 9=D | 13=J | 17=V | 21=K |
| 2=W | 6=Y | 10=X | 14=Z | 18=H | 22=E |
| 3=A | 7=F | 11=B | 15=S | 19=L | 23=T |

**HERRAMIENTASPERSONAS**

**public class HerramientasPersonas {**

**public static *String* getLetraNIF(*int* *numeroDni*){**

***int* resultado = (numeroDni - ((numeroDni / 23) \* 23));**

***String* letra = "";**

**if (resultado == 0){**

**letra = "T";**

**}else if (resultado == 1){**

**letra = "R";**

**}else if (resultado == 2){**

**letra = "W";**

**}else if (resultado == 3){**

**letra = "A";**

**}else if (resultado == 4){**

**letra = "G";**

**}else if (resultado == 5){**

**letra = "M";**

**}else if (resultado == 6){**

**letra = "Y";**

**}else if (resultado == 7){**

**letra = "F";**

**}else if (resultado == 8){**

**letra = "P";**

**}else if (resultado == 9){**

**letra = "D";**

**}else if (resultado == 10){**

**letra = "X";**

**}else if (resultado == 11){**

**letra = "B";**

**}else if (resultado == 12){**

**letra = "N";**

**}else if (resultado == 13){**

**letra = "J";**

**}else if (resultado == 14){**

**letra = "Z";**

**}else if (resultado == 15){**

**letra = "S";**

**}else if (resultado == 16){**

**letra = "Q";**

**}else if (resultado == 17){**

**letra = "V";**

**}else if (resultado == 18){**

**letra = "H";**

**}else if (resultado == 19){**

**letra = "L";**

**}else if (resultado == 20){**

**letra = "C";**

**}else if (resultado == 21){**

**letra = "K";**

**}else if (resultado == 22){**

**letra = "E";**

**}else if (resultado == 23){**

**letra = "T";**

**}**

**return letra;**

**}**

**}**

**EJECUCIONPERSONAS**

**import java.util.Scanner;**

**public class EjecucionPersonas {**

**public static *void* main(*String*[] *args*) {**

**//12345678 -- Z**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca número de NIF");**

***String* entrada = teclado.nextLine();**

**//RECUPERAMOS EL NUMERO DE DNI**

***int* numeroDni = Integer.parseInt(entrada);**

***String* letra = HerramientasPersonas.getLetraNIF(numeroDni);**

**System.out.println("Su letra de NIF es " + letra);**

**}**

**}**

A continuación vamos a realizar la versión 2, que es utilizando métodos de la clase String.

String secuencia = “TRWA**G**…”;

Recuperamos la letra que corresponde al número de la secuencia.

**VERSION 2**

**public static *String* getLetraNIF(*int* *numeroDni*){**

***int* resultado = (numeroDni - ((numeroDni / 23) \* 23));**

***String* secuencia = "TRWAGMYFPDXBNJZSQVHLCKET";**

***char* letra = secuencia.charAt(resultado);**

***String* letraConvertida =**

**String.valueOf(letra);**

**return letraConvertida;**

**}**

El siguiente paso es implementar un método que recibirá el DNI completo con letra y nos devolverá si es correcto el dni o no respecto a la letra

**public class HerramientasPersonas {**

**public static *String* getLetraNIF(*int* *numeroDni*){**

***int* resultado = (numeroDni - ((numeroDni / 23) \* 23));**

***String* secuencia = "TRWAGMYFPDXBNJZSQVHLCKET";**

***char* letra = secuencia.charAt(resultado);**

***String* letraConvertida =**

**String.valueOf(letra);**

**return letraConvertida;**

**}**

**//QUISIERA UN METODO QUE RECIBA UN NIF/DNI**

**//Y NOS DEVUELVA SI ES VALIDO O NO**

**public static *boolean* validarNIF(*String* *nif*){**

***int* longitud = nif.length() - 1;**

**//RECUPERAR EL ULTIMO CARACTER**

***char* ultimaLetra = nif.charAt(longitud);**

**//ELIMINAMOS EL ULTIMO CARACTER**

***String* soloNumeros =**

**nif.substring(0, longitud);**

**//NECESITAMOS CONVERTIR LOS TIPOS**

***int* numeroDni = Integer.parseInt(soloNumeros);**

**//CONVERTIMOS LA LETRA A STRING PARA COMPARAR**

***String* letraComparacion =**

**String.valueOf(ultimaLetra);**

**//LLAMAMOS AL METODO QUE NOS DEVUELVE UNA LETRA**

***String* resultado = getLetraNIF(numeroDni);**

**return resultado.equals(letraComparacion);**

**}**

**}**

Probamos la aplicación en **EjecucionPersonas**

**import java.util.Scanner;**

**public class EjecucionPersonas {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca número de NIF");**

***String* entrada = teclado.nextLine();**

**//RECUPERAMOS EL NUMERO DE DNI**

***int* numeroDni = Integer.parseInt(entrada);**

***String* letra = HerramientasPersonas.getLetraNIF(numeroDni);**

**System.out.println("Su letra de NIF es " + letra);**

**System.out.println("Introduzca un DNI completo");**

***String* dni = teclado.nextLine();**

***boolean* respuesta =**

**HerramientasPersonas.validarNIF(dni);**

**System.out.println("El dni " + dni**

**+ " es correcto: " + respuesta);**

**}**

**}**

El siguiente método que debemos realizar sobre **HerramientasPersona** será un método que nos devolverá el día de nacimiento de una persona.

Dicho método, para su funcionalidad necesita un día, un mes y un año.

* **getDiaNacimiento(int dia, int mes, int anyo):** Devolverá un String con el día de la semana que corresponda

Pedir una fecha al usuario para calcular el día de la semana que nació. Tenemos que tener la tabla de días de la semana para la correspondencia comenzando en sábado:

|  |  |
| --- | --- |
| **DIA** | **NÚMERO** |
| **Sábado** | 0 |
| **Domingo** | 1 |
| **Lunes** | 2 |
| **Martes** | 3 |
| **Miércoles** | 4 |
| **Jueves** | 5 |
| **Viernes** | 6 |

Debemos tener el día, el número de mes y el año que el usuario haya nacido.

A partir de esto datos hay que calcular lo siguiente para averiguar el día de la semana de nacimiento:

Ejemplo 🡪 22/06/1983

Hay que tener en cuenta el mes para realizar el cálculo, si el mes es Enero, el Mes será 13 y restaremos uno al año. Si el Mes es Febrero, el Mes será 14 y restaremos uno al año.

Para poder calcular el número final de la semana debemos seguir los siguientes pasos:

1. Multiplicar el Mes más 1 por 3 y dividirlo entre 5

((6 + 1) \* 3) / 5 🡪 4

1. Dividir el año entre 4

1983 / 4 🡪 495

1. Dividir el año entre 100

1983 / 100 🡪 19

1. Dividir el año entre 400

1983 / 400 🡪 4

1. Sumar el dia, el doble del mes, el año, el resultado de la operación 1, el resultado de la operación 2, menos el resultado de la operación 3 más la operación 4 más 2.

22 + (6 \* 2) + 1983 + 4 + 495 - 19 + 4 + 2 🡪 2503

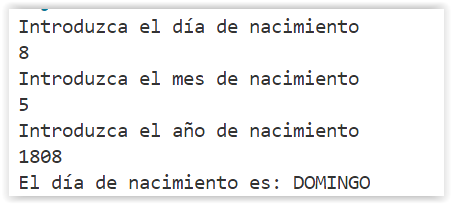
1. Dividir el resultado anterior entre 7.

2503 / 7 🡪 357

1. Restar el número del paso 5 con el número del paso 6 por 7.

2503 – (357 \* 7) 🡪 4

1. Miramos la tabla y vemos que el número 4 corresponde a Miércoles



Vamos a utilizar la clase **HerramientasPersonas** y para la ejecución nos creamos otra distinta para las peticiones y las pruebas.

**HERRAMIENTASPERSONAS**

**public static *String* getDiaNacimiento(*int* *dia*, *int* *mes*, *int* *anyo*){**

**//COMPROBAMOS ENERO Y FEBRERO**

**if (mes == 1){**

**mes = 13;**

**anyo--;**

**}else if (mes == 2){**

**mes = 14;**

**anyo--;**

**}**

**//ALMACENAMOS EN VARIABLES DIFERENTES CADA OPERACION MATEMATICA**

***int* op1 = ((mes + 1) \* 3) / 5;**

***int* op2 = anyo / 4;**

***int* op3 = anyo / 100;**

***int* op4 = anyo / 400;**

**//Sumar el dia,**

**//el doble del mes, el año, el resultado de la operación 1**

**//, el resultado de la operación 2,**

**//menos el resultado de la operación 3 más la operación 4 más 2.**

***int* op5 = dia + (mes \* 2) + anyo + op1 + op2 - op3 + op4 + 2;**

***int* op6 = op5 / 7;**

***int* resultado = op5 - (op6 \* 7);**

***String* diaSemana = "";**

**if (resultado == 0){**

**diaSemana = "SABADO";**

**}else if (resultado == 1){**

**diaSemana = "DOMINGO";**

**}else if (resultado == 2){**

**diaSemana = "LUNES";**

**}else if (resultado == 3){**

**diaSemana = "MARTES";**

**}else if (resultado == 4){**

**diaSemana = "MIERCOLES";**

**}else if (resultado == 5){**

**diaSemana = "JUEVES";**

**}else if (resultado == 6){**

**diaSemana = "VIERNES";**

**}**

**return diaSemana;**

**}**

**EJECUCIONDIANACIMIENTO**

**import java.util.Scanner;**

**public class EjecucionDiaNacimiento {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca el día de nacimiento");**

***String* entrada = teclado.nextLine();**

***int* dia = Integer.parseInt(entrada);**

**System.out.println("Introduzca el mes de nacimiento");**

**entrada = teclado.nextLine();**

***int* mes = Integer.parseInt(entrada);**

**System.out.println("Introduzca el año de nacimiento");**

**entrada = teclado.nextLine();**

***int* anyo = Integer.parseInt(entrada);**

***String* diaSemana =**

**HerramientasPersonas.getDiaNacimiento(dia, mes, anyo);**

**System.out.println("El día de nacimiento es: "**

**+ diaSemana);**

**}**

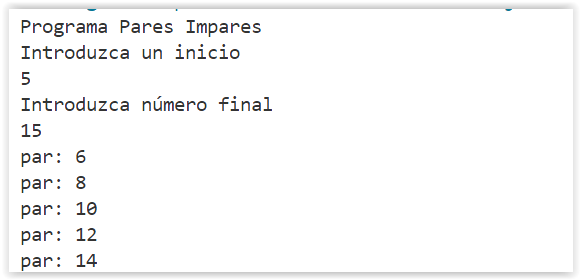
**}**

El siguiente paso será crearnos otro proyecto distinto.

Creamos un nuevo proyecto de Java llamado **funcionescomplejas**

Vamos a crear una nueva clase para **mostrar** los números pares que existen entre dos números.

Vamos a enviar, a un método, un inicio y un final y lo que haremos es **mostrar** los números pares.



Creamos una nueva clase llamada **ClaseParesImpares**

Creamos un nuevo método llamado **getNumerosPares(inicio, fin)**

**CLASEPARESIMPARES**

**public class ClaseParesImpares {**

**public static *void* getNumerosPares(*int* *inicio*, *int* *fin*){**

**//BUCLE DESDE INICIO HASTA FIN**

**for (*int* i = inicio; i <= fin; i++){**

**//PREGUNTAMOS POR LOS PARES**

**if (i % 2 == 0){**

**System.out.println("par: " + i);**

**}**

**}**

**}**

**}**

Creamos una clase para probar las acciones. **EjecucionParesImpares**

**EJECUCIONPARESIMPARES**

**import java.util.Scanner;**

**public class EjecucionParesImpares {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Programa Pares Impares");**

**System.out.println("Introduzca un inicio");**

***String* entrada = teclado.nextLine();**

***int* inicio = Integer.parseInt(entrada);**

**System.out.println("Introduzca número final");**

**entrada = teclado.nextLine();**

***int* fin = Integer.parseInt(entrada);**

**ClaseParesImpares.getNumerosPares(inicio, fin);**

**}**

**}**

A continuación, vamos a crear otro método para mostrar los números impares entre un rango. Dentro de la clase **ClaseParesImpares**

* getNumerosImpares(inicio, fin)

**CLASEPARESIMPARES**

**public static *void* getNumerosImpares(*int* *inicio*, *int* *fin*){**

**//BUCLE DESDE INICIO HASTA FIN**

**for (*int* i = inicio; i <= fin; i++){**

**//PREGUNTAMOS POR LOS IMPARES**

**if (i % 2 != 0){**

**System.out.println("impar: " + i);**

**}**

**}**

**}**

Podríamos tener un método SOLAMENTE donde el usuario nos esté enviando un **inicio**, un **final** y que nos diga que es lo que quiere.

Por ejemplo: **getRangoNumeros(inicio, fin, pares)**

Vamos a mantener los dos métodos que ya tenemos.

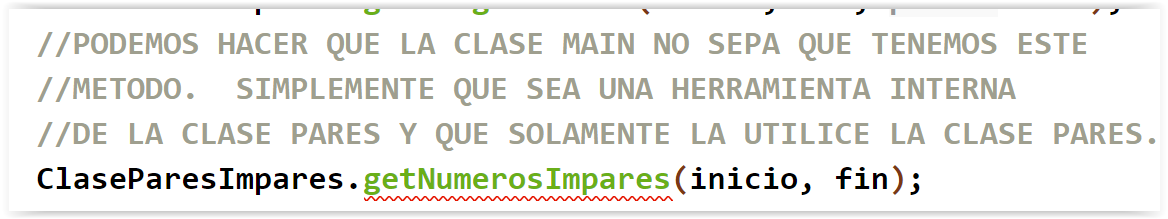
Lo que haremos será un método “intermedio” que llamará a los otros dos, dependiendo lo que el usuario decida.

Si hemos realizado el método **getRangoNumeros**, ya no tiene sentido dejar que la clase pares deje visualizar sus otros dos métodos ya que, en realidad, estamos dejándole que lo haga mediante el método del rango.

Cuando hablamos de visibilidad de elementos/métodos/funciones dentro de las clases, podemos hacer que sean visibles o no para el resto de clases.

Si deseamos que un método sea visible por otras clases, se utiliza **public**

Si deseamos que un método NO sea visible por otras clases, utilizamos **private**



**CLASEPARESIMPARES**

**public class ClaseParesImpares {**

**public static *void* getRangoNumeros(*int* *inicio*, *int* *fin*, *boolean* *pares*){**

**if (pares == true){**

**getNumerosPares(inicio, fin);**

**}else{**

**getNumerosImpares(inicio, fin);**

**}**

**}**

**private static *void* getNumerosPares(*int* *inicio*, *int* *fin*){**

**//BUCLE DESDE INICIO HASTA FIN**

**for (*int* i = inicio; i <= fin; i++){**

**//PREGUNTAMOS POR LOS PARES**

**if (i % 2 == 0){**

**System.out.println("par: " + i);**

**}**

**}**

**}**

**private static *void* getNumerosImpares(*int* *inicio*, *int* *fin*){**

**//BUCLE DESDE INICIO HASTA FIN**

**for (*int* i = inicio; i <= fin; i++){**

**//PREGUNTAMOS POR LOS IMPARES**

**if (i % 2 != 0){**

**System.out.println("impar: " + i);**

**}**

**}**

**}**

**}**

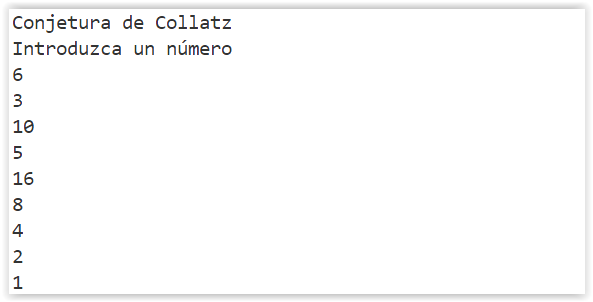
Un método private es una herramienta de la clase y no tiene porqué utilizarla un clase de ejecución.

Vamos a realizar una clase que nos **mostrará** los números de la conjetura de Collatz.

La conjetura de Collatz indica que todo número positivo siempre llegará a ser 1 siguiendo una secuencia:

* Si el número es par, dividimos entre 2
* Si el número es impar, multiplicamos por 3 y le sumamos 1

6, 3, 10, 5, 16, 8, 4, 2, **1**



Vamos a realizar un código para hacer esta teoría y, después, vamos a cambiar y crear algún método private.

Creamos una clase llamada **HerramientasNumeros**

Tendremos un método que llamaremos **mostrarCollatz(numero)**

**HERRAMIENTASNUMEROS**

**public class HerramientasNumeros {**

**public static *void* mostrarCollatz(*int* *numero*){**

**while (numero != 1){**

**//COMPROBAMOS EL NUMERO**

**if (numero % 2 == 0){**

**//DIVIDIMOS ENTRE 2**

**numero = numero / 2;**

**}else{**

**//MULTIPLICAMOS POR 3 + 1**

**numero = numero \* 3 + 1;**

**}**

**System.out.println(numero);**

**}**

**}**

**}**

Creamos una clase llamada **EjecucionNumeros**

**EJECUCIONNUMEROS**

**import java.util.Scanner;**

**public class EjecucionNumeros {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Conjetura de Collatz");**

**System.out.println("Introduzca un número");**

***String* entrada = teclado.nextLine();**

***int* numero = Integer.parseInt(entrada);**

**HerramientasNumeros.mostrarCollatz(numero);**

**}**

**}**

Modificamos el código de la conjetura de Collatz para llevarnos a un método interno la ejecución de la división o multiplicación del número.

**HERRAMIENTASNUMEROS**

**public class HerramientasNumeros {**

**public static *void* mostrarCollatz(*int* *numero*) {**

**//6**

**while (numero != 1) {**

**//String mensaje = getMensajeCollatz(numero);**

**//System.out.println(mensaje);**

***int* num = getNumeroCollatz(numero);**

**numero = num;**

**System.out.println(num);**

**}**

**}**

**// VAMOS A CREAR UN METODO private QUE NOS DEVOLVERA EL NUMERO**

**//8, 15**

**private static *int* getNumeroCollatz(*int* *numero*) {**

**// COMPROBAMOS EL NUMERO**

**if (numero % 2 == 0) {**

**// DIVIDIMOS ENTRE 2**

**numero = numero / 2;**

**} else {**

**// MULTIPLICAMOS POR 3 + 1**

**numero = numero \* 3 + 1;**

**}**

**return numero;**

**}**

**// VAMOS A CREAR UN METODO private QUE NOS DEVOLVERA EL MENSAJE**

**// "par: " 8, "impar: " + 15**

**private static *String* getMensajeCollatz(*int* *numero*) {**

**// COMPROBAMOS EL NUMERO**

***String* mensaje = "";**

**if (numero % 2 == 0) {**

**// DIVIDIMOS ENTRE 2**

**numero = numero / 2;**

**mensaje = "Par: " + numero;**

**} else {**

**// MULTIPLICAMOS POR 3 + 1**

**numero = numero \* 3 + 1;**

**mensaje = "Impar: " + numero;**

**}**

**return mensaje;**

**}**

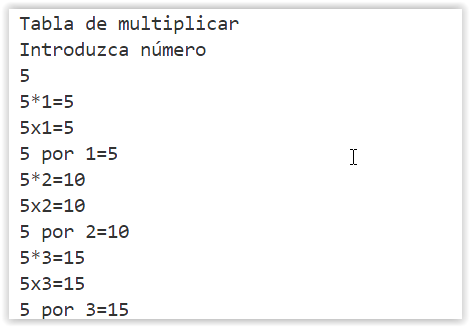
**}**

Necesitamos una clase llamada **HerramientasTabla** y otra clase llamada **EjecucionTabla**

En herramientas tabla tendremos un método que nos mostrará la tabla de multiplicar de un número que enviemos al método.

* **mostrarTablaMultiplicar**(int numero)

Ejecución tabla llamará a mostrar tabla y veremos los resultados.



Si nos dan el número 5:

5 por 1=5

5 por 2=10

5 por 3=15

...

5 por 10=50

5x1=5

5x2=10

5x3=15

…

5x10=50

**HERRAMIENTASTABLA**

**public class HerramientasTabla {**

**public static *void* mostrarTablaMultiplicar(*int* *numero*){**

**for (*int* i = 1; i <= 10; i++){**

**//PODEMOS REPRESENTAR TODOS LOS FORMATOS**

**System.out.println(getMensajeTabla(numero, i, "\*"));**

**System.out.println(getMensajeTabla(numero, i, "x"));**

**System.out.println(getMensajeTabla(numero, i, " por "));**

**}**

**}**

**private static *String* getMensajeTabla(*int* *num*, *int* *multi*, *String* *sep*){**

***int* operacion = num \* multi;**

**return num + sep + multi + "=" + operacion;**

**}**

**}**

**EJECUCIONTABLA**

**import java.util.Scanner;**

**public class EjecucionTabla {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Tabla de multiplicar");**

**System.out.println("Introduzca número");**

***String* entrada = teclado.nextLine();**

***int* numero = Integer.parseInt(entrada);**

**HerramientasTabla.mostrarTablaMultiplicar(numero);**

**}**

**}**

Una clase llamada **EjecucionArrays** con un método main.

1. En el método main creamos un Array con 3 posiciones.
2. Guardamos números en cada posición (los que sean)
3. Creamos un bucle para recorrer cada elemento del Array y sumamos cada número
4. Mostramos la suma total de todos los números del Array

**EJEMPLO SIMPLE**

**public class EjecucionArrays {**

**public static *void* main(*String*[] *args*) {**

**//NECESITAMOS UN ARRAY DE 3 POSICIONES**

**//CON TIPO int**

***int*[] numeros = new *int*[3];**

**numeros[0] = 99;**

**numeros[1] = 2;**

**numeros[2] = 1;**

***int* suma = 0;**

**for (*int* i = 0; i < numeros.length; i++){**

**suma += numeros[i];**

**}**

**System.out.println("La suma es: " + suma);**

**}**

**}**

El siguiente ejemplo será una clase que nos permitirá jugar con un Array de números. Vamos a llamar a la clase **JuegoArrays**

En esta clase, tendremos un método que recibirá un Array de números y **devolverá** la suma de dichos números.

* **sumarNumerosArray(int[] numeros)**

Creamos una clase llamada **EjecucionArrays** y lo probamos ahí.

**JUEGOARRAYS**

**public class JuegoArrays {**

**public static *int* sumarNumerosArray(*int*[] *numeros*){**

***int* suma = 0;**

**for (*int* i = 0; i < numeros.length; i++){**

**suma += numeros[i];**

**}**

**return suma;**

**}**

**}**

**EJECUCIONARRAYS**

**public class EjecucionArrays {**

**public static *void* main(*String*[] *args*) {**

**//NECESITAMOS UN ARRAY DE 3 POSICIONES**

**//CON TIPO int**

***int*[] numeros = new *int*[3];**

**numeros[0] = 99;**

**numeros[1] = 2;**

**numeros[2] = 1;**

***int* suma = JuegoArrays.sumarNumerosArray(numeros);**

**System.out.println("La suma es: " + suma);**

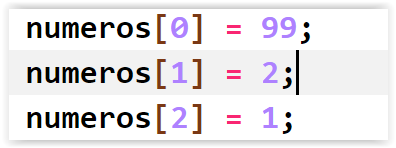
**}**

**}**

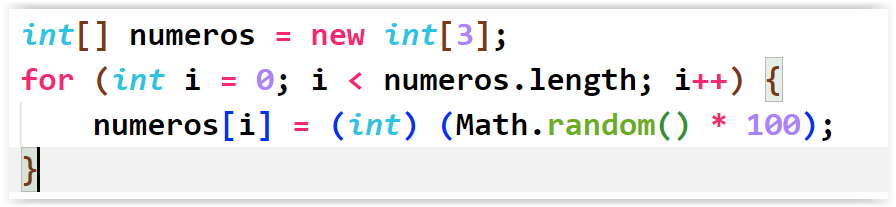
Versión 2:

En la clase **EjecucionArrays**, en lugar de rellenar un Array de números de forma manual, lo que haremos será rellenar un Array con números Random.

Recorremos todo el Array y lo rellenamos con valores Random.



Código de ejemplo:



El siguiente paso es poder tener un Array en el que le indiquemos el número de elementos que deseamos almacenar.

Vamos a dinamizar el código mediante la clase **JuegoArrays**

En la clase **JuegoArrays** lo que haremos será tener un método que nos pedirá el número de elementos que necesita para el Array y nos **devolverá** un Array relleno con números aleatorios.

Creamos un nuevo método llamado **getArrayRandom(int elementos)**

**JUEGOARRAYS**

**public class JuegoArrays {**

**public static *int*[] getArrayRandom(*int* *elementos*){**

***int*[] numeros = new *int*[elementos];**

**for (*int* i = 0; i < numeros.length; i++){**

**numeros[i] = (*int*) (Math.random() \* 100);**

**System.out.println("Random: " + numeros[i]);**

**}**

**return numeros;**

**}**

**public static *int* sumarNumerosArray(*int*[] *numeros*){**

***int* suma = 0;**

**for (*int* i = 0; i < numeros.length; i++){**

**suma += numeros[i];**

**}**

**return suma;**

**}**

**}**

**EJECUCIONARRAYS**

**public class EjecucionArrays {**

**public static *void* main(*String*[] *args*) {**

**//RECUPERAMOS UN ARRAY CON EL TAMAÑO QUE DESEEMOS**

***int*[] numeros = JuegoArrays.getArrayRandom(3);**

***int* suma = JuegoArrays.sumarNumerosArray(numeros);**

**System.out.println("La suma es: " + suma);**

**}**

**}**

Vamos a crear una clase llamada **EjecucionTextos** con un método **main**.

Dicha clase tendrá una variable **String** llamada **texto** que contendrá un texto numérico: **1234**

Debemos recorrer cada **letra** numérica del String y sumar sus cifras.

1234 = 10



¿Sabemos crear el String? Si

¿Sabemos recorrer un String letra a letra? Puede… (for)

¿Sabemos recuperar una letra en una posición de un String? Si (charAt(posición))

char letra = ‘1’

Necesitamos que el carácter ‘**1**’ sea un **int** 1

Debemos convertir la clase **char** a **int**

¿Sabemos convertir un char a int? Puede…Casting

¿Sabemos convertir un valor en una variable a String? String.**valueOf**

**EJECUCIONTEXTOS**

**public class EjecucionTextos {**

**public static *void* main(*String*[] *args*) {**

***String* texto = "1234";**

***int* suma = 0;**

**for (*int* i = 0; i < texto.length(); i++){**

***char* letra = texto.charAt(i);**

**//TRUCO: LOS DE MINUSCULAS SON CON CASTING**

**//CONVERTIMOS NUESTRO char A int**

**//ESTA CONVERSION PRIMITIVA RECUPERA EL VALOR**

**//ASCII DEL char QUE ESTAMOS RECORRIENDO**

**//DEBEMOS CONVERTIR EL char A String**

***String* temporal = String.valueOf(letra);**

**//SABEMOS CONVERTIR UN String A int?**

***int* numero = Integer.parseInt(temporal);**

**suma += numero;**

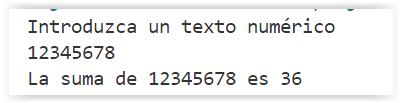
**}**

**System.out.println("La suma de " + texto + " es " + suma);**

**}**

**}**

Versión 2:



Creamos una clase llamada **JuegoTextos** y debemos crear un método que recibirá un String numérico y nos **devolverá** la **suma** de todos sus caracteres.

* **getSumarTexto(String texto)**

**JUEGOTEXTOS**

**public class JuegoTextos {**

**public static *int* getSumarTexto(*String* *texto*){**

***int* suma = 0;**

**for (*int* i = 0; i < texto.length(); i++){**

***char* letra = texto.charAt(i);**

***String* temporal = String.valueOf(letra);**

**//SABEMOS CONVERTIR UN String A int?**

***int* numero = Integer.parseInt(temporal);**

**suma += numero;**

**}**

**return suma;**

**}**

**}**

Lo probamos en la clase **EjecucionTextos**

**EJECUCIONTEXTOS**

**import java.util.Scanner;**

**public class EjecucionTextos {**

**public static *void* main(*String*[] *args*) {**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca un texto numérico");**

***String* texto = teclado.nextLine();**

***int* suma = JuegoTextos.getSumarTexto(texto);**

**System.out.println("La suma de " + texto + " es " + suma);**

**}**

**}**

Este ejemplo es MUY parecido, simplemente tiene algún elemento más.

Tenéis que pensar, de todo el problema, paso a paso que es lo que sabéis y lo que no.

EJEMPLO DE NUMERO ISBN QUE ESTÁ BIEN: **8441513929**

1.- Se descompone la cadena y se multiplica cada número por la posición que ocupa en la cadena:

8 \* 1

4 \* 2

4 \* 3

1 \* 4

5 \* 5

.

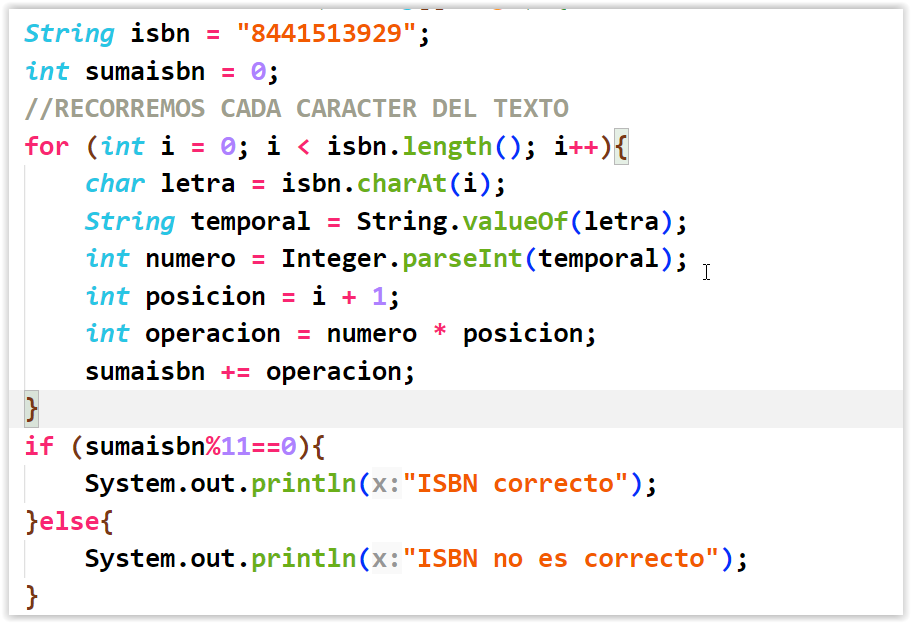
.

9 \* 10

2.- La suma de todas estas multiplicaciones se divide entre 11, y si el resto es cero, el número ISBN es correcto.

Sobre **EjecucionTextos** probamos el ejemplo (**main**) y después un método llamado **validarISBN(String isbn)** en **JuegoTextos** que nos **devolverá** true/false

**EJECUCIONTEXTOS**



Versión 2 con método en la clase **JuegoTextos**

**JUEGOTEXTOS**

**public static *boolean* validarISBN(*String* *isbn*){**

***int* sumaisbn = 0;**

**//RECORREMOS CADA CARACTER DEL TEXTO**

**for (*int* i = 0; i < isbn.length(); i++){**

***char* letra = isbn.charAt(i);**

***String* temporal = String.valueOf(letra);**

***int* numero = Integer.parseInt(temporal);**

***int* posicion = i + 1;**

***int* operacion = numero \* posicion;**

**sumaisbn += operacion;**

**}**

**if (sumaisbn%11==0){**

**return true;**

**}else{**

**return false;**

**}**

**}**

**EJECUCIONTEXTOS**

**import java.util.Scanner;**

**public class EjecucionTextos {**

**public static *void* main(*String*[] *args*) {**

***String* isbn = "8441513929";**

***boolean* respuesta = JuegoTextos.validarISBN(isbn);**

**if (respuesta == true){**

**System.out.println("ISBN correcto");**

**}else{**

**System.out.println("El numero ISBN es incorrecto");**

**}**

***Scanner* teclado = new Scanner(System.in);**

**System.out.println("Introduzca un texto numérico");**

***String* texto = teclado.nextLine();**

***int* suma = JuegoTextos.getSumarTexto(texto);**

**System.out.println("La suma de " + texto + " es " + suma);**

**}**

**}**

Necesito un método que compruebe si un número es **Narcisista** o no.

Un número narcisista es aquel en el que la suma de sus potencias de longitud de caracteres devuelve el mismo número.

**^**

Ejemplo: 153

Longitud de caracteres: 3.

La potencia a elevar cada carácter es 3.

Potencia: **Math.pow**

1 elevado a 3 -> 1

5 elevado a 3 -> 125

3 elevado a 3 -> 27

**1+125+27=153**