

## Unidad 5. Modularidad

### Métodos en clases externas.

Un método siempre está asociado o pertenece a un objeto o clase; un método no puede existir por si solo, siempre necesitará de un objeto. Por tanto, en Java siempre estaremos obligados a crear un objeto para poder crear y usar los métodos.

Java es un lenguaje de programación que permite utilizar los métodos que han sido creados en otras clases. Esto facilita la reutilización de código.

#### Ejemplo 5.4

Realiza el programa en JAVA tal que, dado como datos la base y la altura de un rectángulo, calcule e imprima el perímetro y la superficie del mismo.

Perímetro =  $2 * (base + altura)$

Superficie =  $base * altura$

Datos: base, altura

Donde: **base** y **altura** son variables de tipo entero que representan las dimensiones de un rectángulo.

**perimetro** y **superficie** son variables de tipo entero que representan los cálculos por realizar.

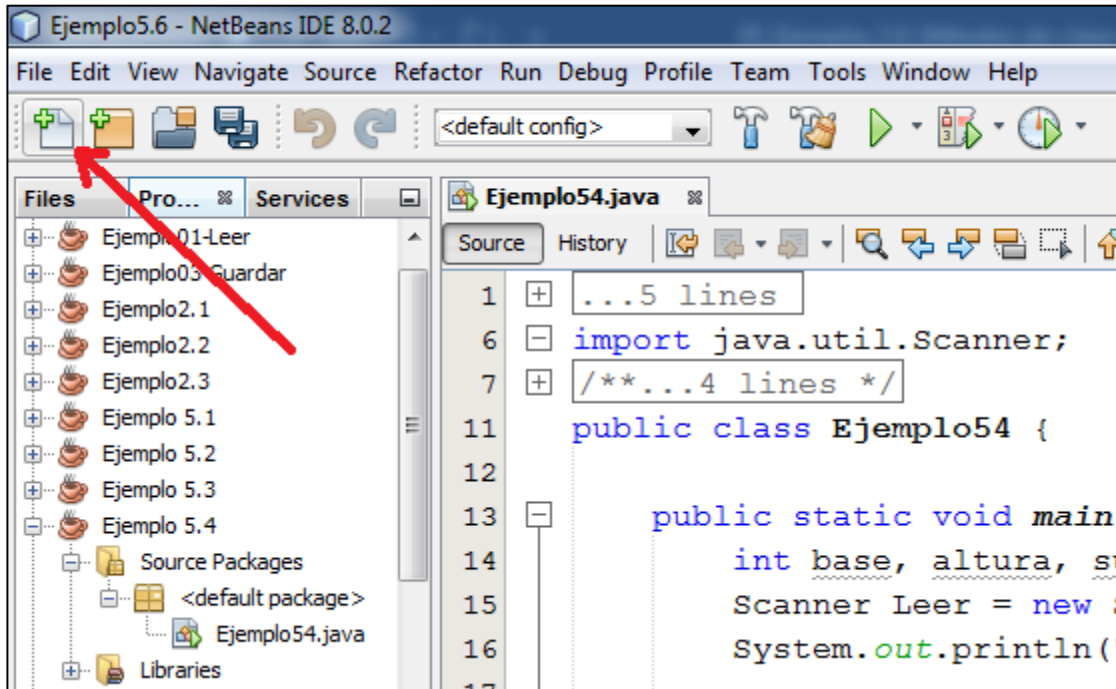
1. Crear el proyecto Ejemplo 5.4 y en el método principal (**main**) se debe declarar las variables, que permitan leer la base y la altura del rectángulo, así como la variable área que recibirá el valor de uno de los métodos.

```
1  ...5 lines
6  import java.util.Scanner;
7  /**...4 lines */
11 public class Ejemplo54 {
12
13     public static void main(String[] args) {
14         int base, altura, area;
15         Scanner Leer = new Scanner(System.in);
16         System.out.println("Perímetro y superficie de un rectángulo");
17         System.out.print("Introduce la base: ");
18         base = Leer.nextInt();
19         System.out.print("Introduce la altura: ");
20         altura = Leer.nextInt();
21
22     }
23
24 }
```

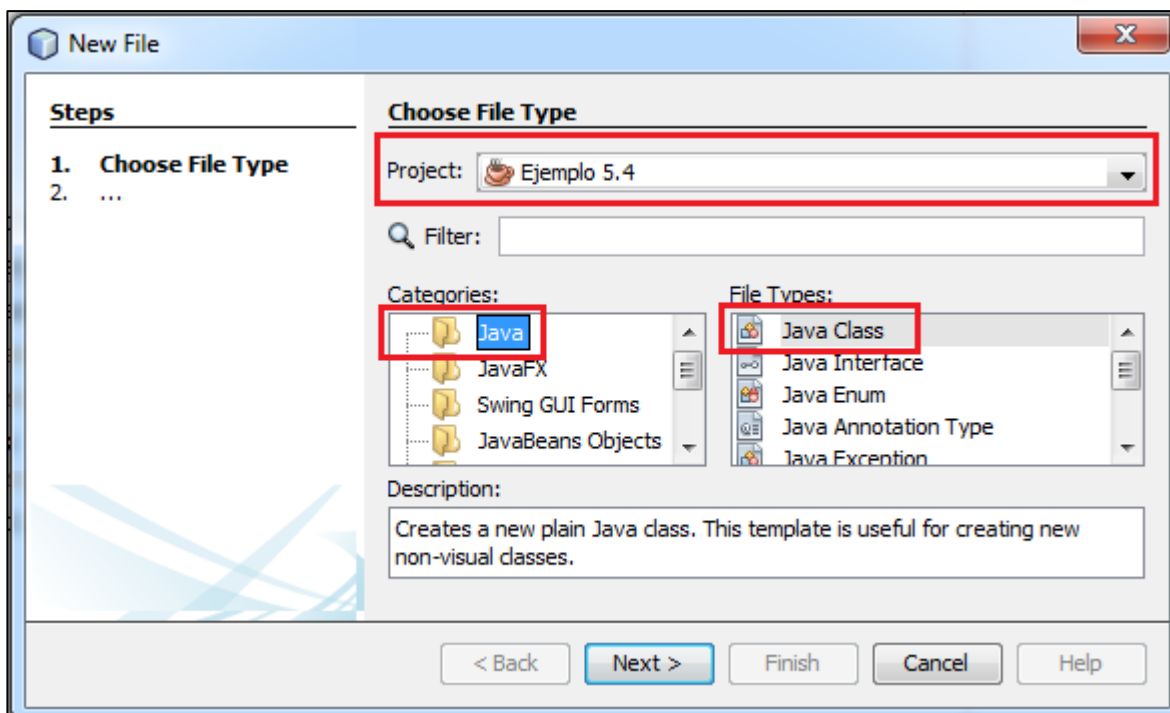
## Crear la clase externa.

La clase externa es un archivo **.java** que normalmente se guarda en el mismo lugar que el programa principal. Para crear la clase externa se deben hacer las siguientes acciones:

2. Crear un nuevo archivo en el menú **File/ New File** o haciendo clic en la hoja blanca.



3. En la ventana de **New File**, elegir el proyecto **Ejemplo 5.4**, la categoría **Java** y el tipo de archivo **Java Class**. Entonces oprimir botón Next.



4. En la ventana de **New Java Class**, en el campo **Class Name** escribir **Calculos** (que será el nombre de la clase). Entonces oprimir botón Finish.

**New Java Class**

**Steps**

1. Choose File Type
2. **Name and Location**

**Name and Location**

Class Name: **Calculos**

Project: Ejemplo 5.4

Location: Source Packages

Package:

Created File: D:\Documentos\Adm\Documents\FP-Academicos\6\Ejemplo 5.4\src\Calculos.java

Warning: It is highly recommended that you do not place Java classes in the default package.

< Back Next > Finish Cancel Help

Observa que se ha creado un nuevo archivo (**Calculos.java**), el cual contiene la clase que también tiene el nombre **Calculos**.

**Files** | **Pro...** | **Services**

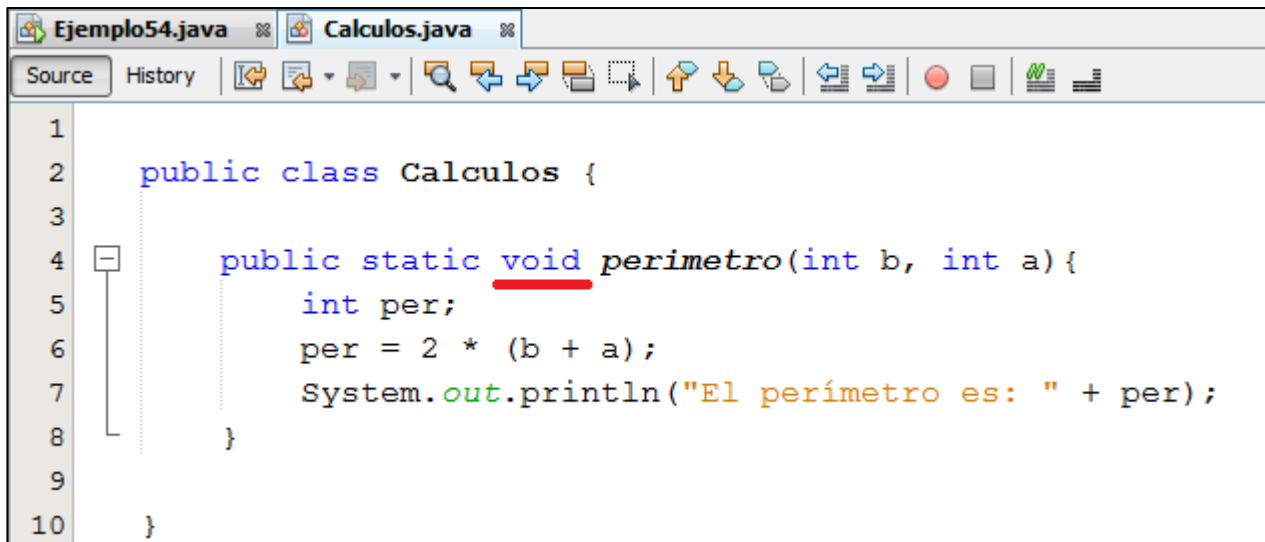
- Ejemplo 5.1
- Ejemplo 5.2
- Ejemplo 5.3
- Ejemplo 5.4**
  - Source Packages
    - <default package>
      - Calculos.java**
      - Ejemplo54.java
  - Libraries

**Ejemplo54.java** | **Calculos.java**

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  /**
8   *
9   * @author Adm
10  */
11  public class Calculos {
12
13  }
```

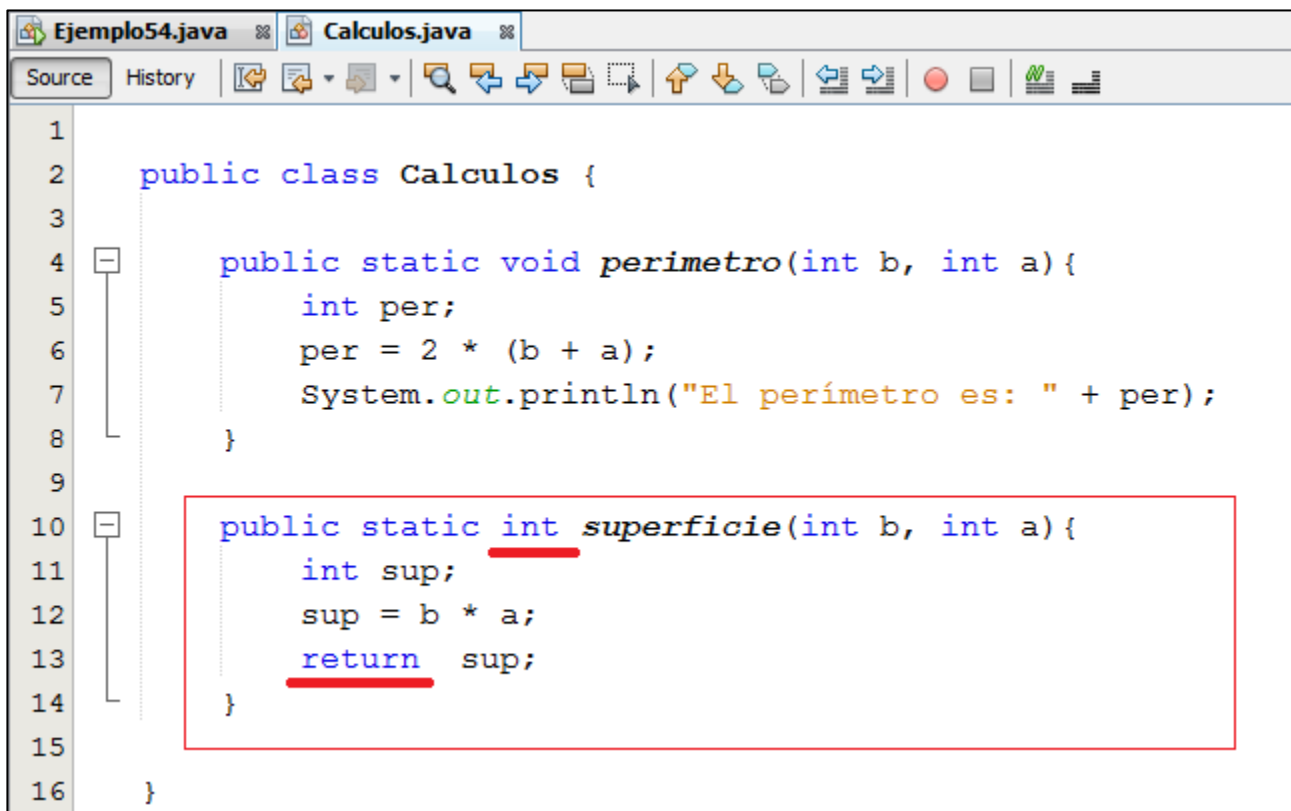
## Crear los métodos en la clase externa.

5. Crear el método **perimetro**, para que ahí mismo se calcule y se muestre el resultado. Este método debe ser de tipo **void** pues solo ejecutará código, sin devolver ningún valor.



```
1
2 public class Calculos {
3
4     public static void perimetro(int b, int a){
5         int per;
6         per = 2 * (b + a);
7         System.out.println("El perímetro es: " + per);
8     }
9
10 }
```

6. Crear el método **superficie**, el cual devolverá el área del rectángulo mediante la sentencia **return**, en este caso al método **superficie** se le debe asignar el tipo de dato **int**.



```
1
2 public class Calculos {
3
4     public static void perimetro(int b, int a){
5         int per;
6         per = 2 * (b + a);
7         System.out.println("El perímetro es: " + per);
8     }
9
10     public static int superficie(int b, int a){
11         int sup;
12         sup = b * a;
13         return sup;
14     }
15
16 }
```

## Llamar o invocar a los métodos de una clase externa.

Para llamar o invocar a los métodos externos se usa la notación de **punto**, es decir, se especifica el nombre del objeto [**punto**] nombre del método, y se envían los parámetros en caso de que el método los tenga. La sintaxis es:

```
nombreDeLaClase.nombreDelMétodo([parámetros]);
```

En este ejemplo, el método **perimetro** que es de tipo void (no devuelve valor) se invocaría de la siguiente manera:

```
Calculos.perimetro(base, altura);
```

Si el método devuelve algún valor, es común recibirlo en una variable con el mismo tipo de dato que el método. La sintaxis es:

```
Variable = nombreDeLaClase.nombreDelMétodo([parámetros]);
```





En el ejemplo, el método **superficie** que es de tipo **int** (devuelve un valor entero), se invocaría de la siguiente manera:

```
area = Calculos.superficie(base, altura);
```

7. Hacer la llamada de los métodos **perimetro** y **superficie** en el método principal (**main**).

```
1  ...5 lines
6  import java.util.Scanner;
7  /**...4 lines */
11 public class Ejemplo54 {
12
13     public static void main(String[] args) {
14         int base, altura, area;
15         Scanner Leer = new Scanner(System.in);
16         System.out.println("Perímetro y superficie de un rectángulo");
17         System.out.print("Introduce la base: ");
18         base = Leer.nextInt();
19         System.out.print("Introduce la altura: ");
20         altura = Leer.nextInt();
21
22         Calculos.perimetro(base, altura);
23         // nombre del método void (que no devuelve valor)
24         area = Calculos.superficie(base, altura);
25         // nombre del método que devuelve valor
26         System.out.println("La superficie es: " + area);
27
28     }
29
30 }
```

8. Finalmente, ejecutar el programa e identificar lo que sucede en cada uno de los métodos (main, perimetro, y superficie) si se introduce 10 para la base y 5 para la altura.

```
run:
Perímetro y superficie de un rectángulo
Introduce la base: 10
Introduce la altura: 5
El perímetro es: 30
La superficie es: 50
BUILD SUCCESSFUL (total time: 9 seconds)
```