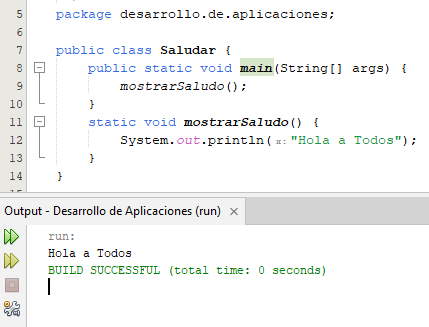
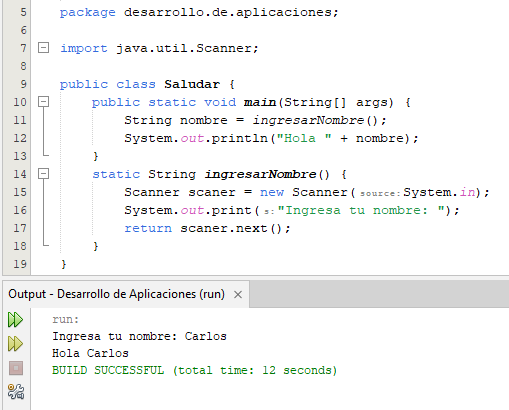
**Desarrollo Avanzado de Aplicaciones I**

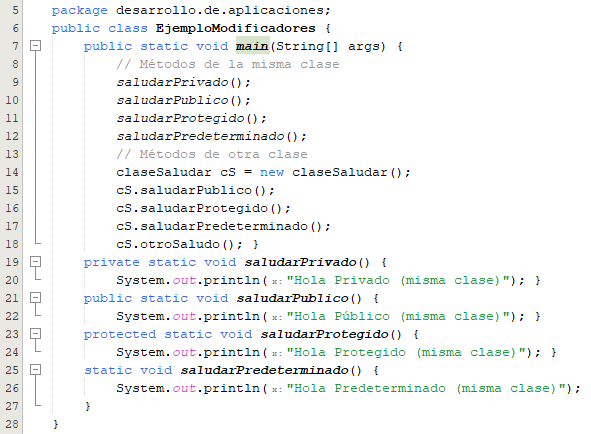
**Ejercicio 1**. Desarrollar una aplicación que invoque un método para mostrar un saludo.

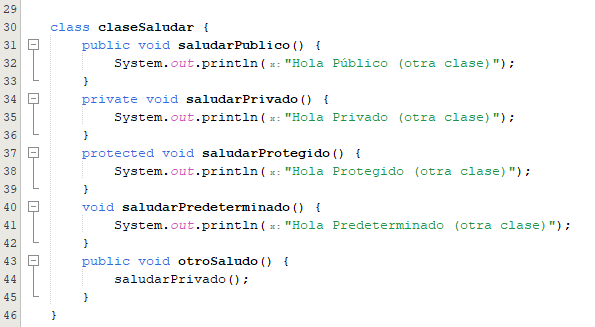


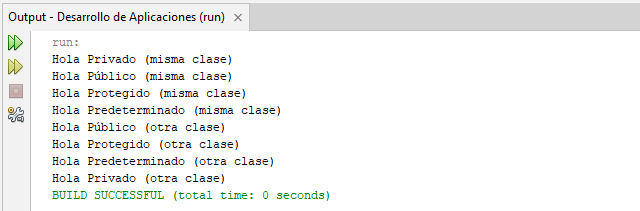
**Ejercicio 2**. Desarrollar una aplicación que invoque un método para solicitar un nombre y lo devuelva para mostrar un saludo.



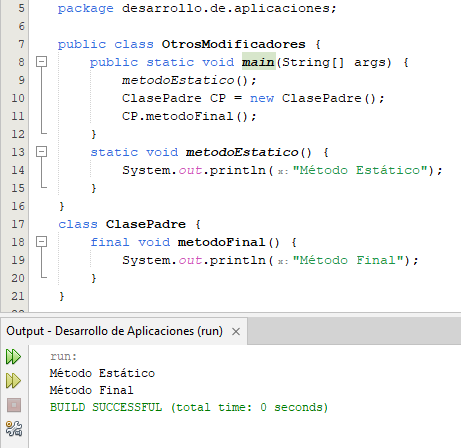
**Ejercicio 3**. Desarrollar una aplicación que invoque a métodos públicos, privados, protegidos y predeterminados de la misma clase y de otra clase.



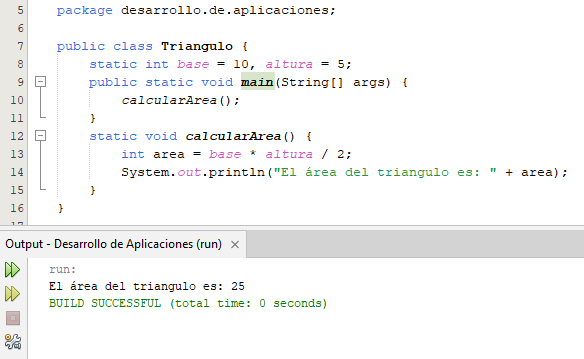




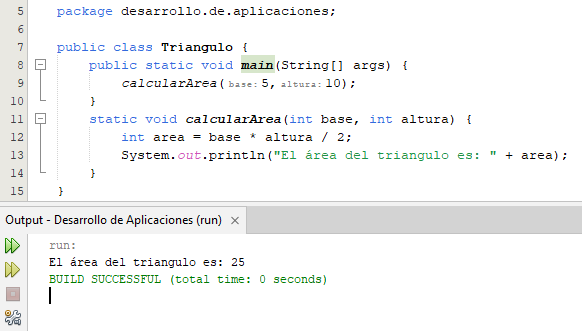
**Ejercicio 4**. Desarrollar una aplicación que utilice metodos estáticos y finales.



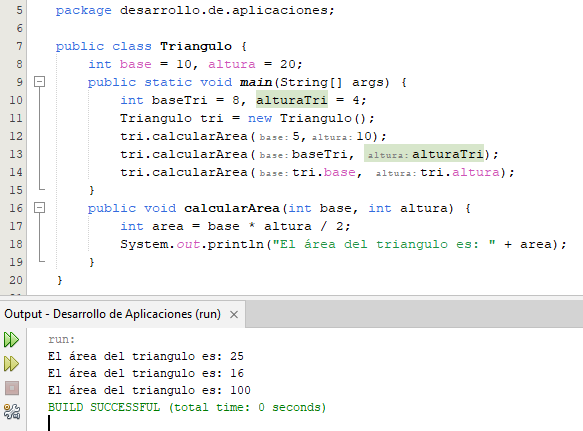
**Ejercicio 5**. Desarrollar una aplicación que utilice variables globales y locales para llamar a un método que calcule el área de un triangulo.



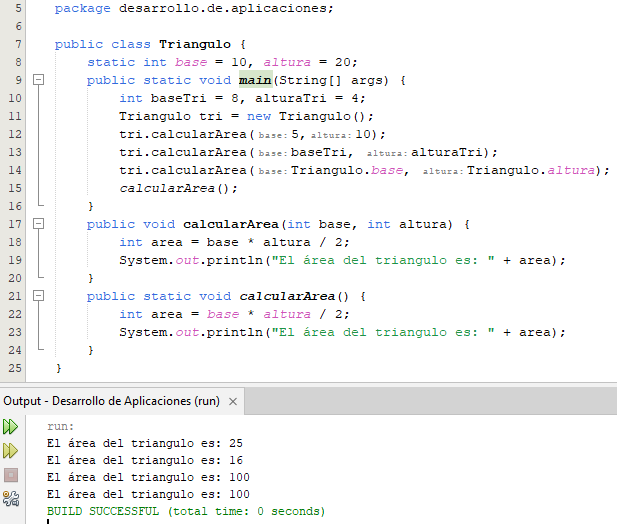
**Ejercicio 6**. Modificar el ejercicio anterior para que el método “calcularArea” use parámetros.



**Ejercicio 7**. Desarrollar una aplicación que llame a un método con parámetros que calcule el área de un triangulo. Mostrar el resultado pasando como argumentos: valores, variables locales y variables globales.

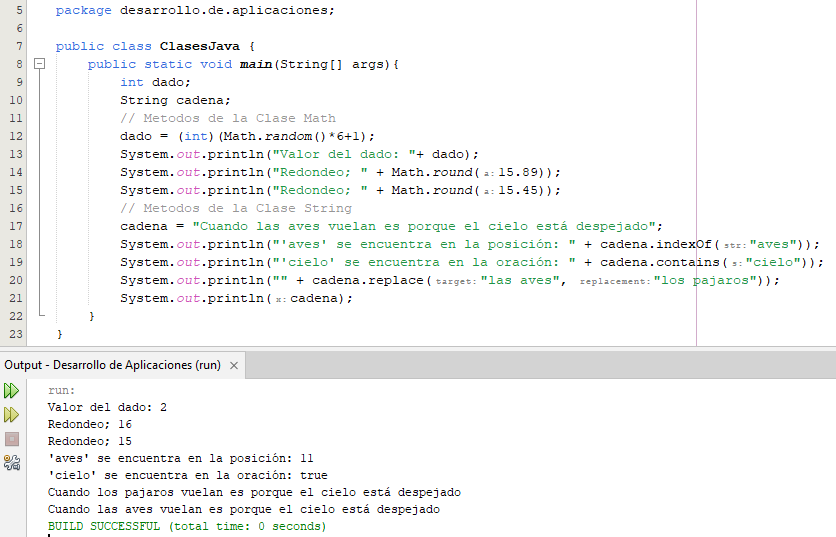


**Ejercicio 8**. Modificar el ejercicio anterior para que incluya un método sobrecargado que calcule el área del triangulo sin parametros utilizando las variables globales.

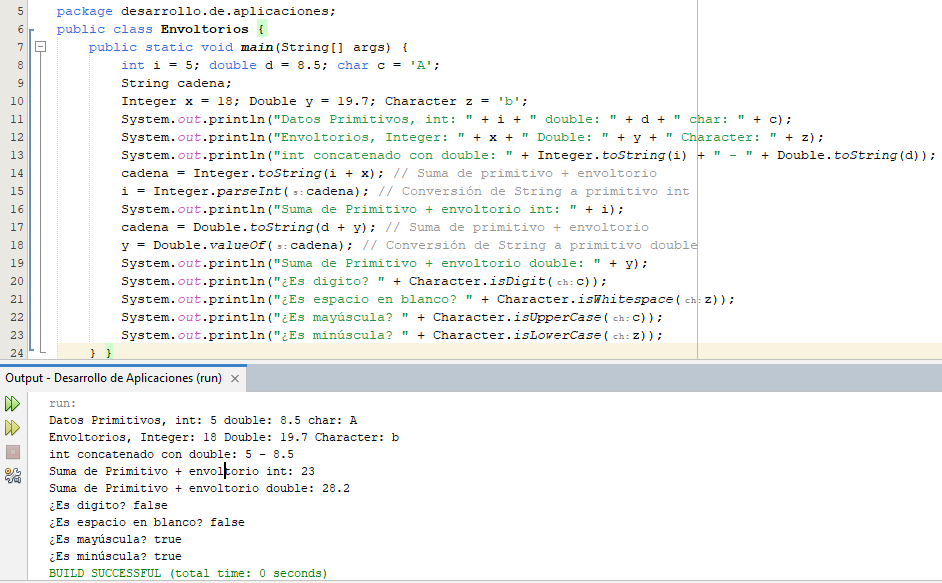


¿Porqué hay que cambiar las variables de la línea 14?

**Ejercicio 9**. Desarrollar una aplicación que utilice los métodos “random” y “round” de la clase “Math” y los métodos “indexOf”, “contains” y “replace” de la clase “String”.



**Ejercicio 10**. Desarrollar una aplicación que muestre la utilización de las Clases Envoltorios.



**Ejercicio 11**. Desarrollar una aplicación que solicite ingresar dos números y que solicite ingresar que operación desea realizar (suma, resta, multiplicación, división y módulo). Las operaciones deben realizarse mediante métodos por cada operación. Realizar el ejercicio para cada uno de los siguientes casos:

* Usando variables globales y sin paso de argumentos. Los métodos deben estar en la misma clase.
* Usando variables locales y pasándolas como argumentos. Los métodos deben estar en la misma clase.
* Usando métodos de otra clase.

**Ejercicio 12**. Desarrollar una aplicación que llame a un método dependiendo de que tipo de carácter es. Si es un número debe mostrar el número, su doble y su triple. Si es una vocal mostrar la palabra “Murcielago” con esa vocal remplazada por un guión “-“. Si es una consonante mostrar si en la cadena “Los anexos y documentos adjuntos faltaban indexar y ver alguna huella o pizca de queja” se encuentra esa consonante. Los métodos de estar sobrecargados (un método para número y otro para carácter).

**Ejercicio 13**. Desarrollar una aplicación que solicite ingresar dos número múltiplos y llame a un método para verificar si efectivamente son múltiplos. Si el método devuelve verdadero mostrar el mensaje “Eres inteligente”, en caso contrario mostrar el mensaje “Debes practicar”.

**Ejercicio 14**. Desarrollar una aplicación que solicite ingresar una figura geometrica y dependiendo de que figura geométrica es, instancie a la clase correspondiente y llame al método para calcular su área donde debe solicitar el ingreso de los datos necesarios:

|  |  |  |
| --- | --- | --- |
| **Figura Geométrica** | **Datos a Solicitar** | **Resultados a Mostrar** |
| Triangulo | base y altura | Área = base \* altura / 2 |
| Cuadrado | lado | Área = lado \* lado |
| Círculo | radio | Área = PI \* radio \* radio |

PI = 3.1416 y no debe poder modificarse.

Cada figura geométrica debe ser una clase diferente y tener su propio método para calcular su área.

**Ejercicio 15**. Modificar el ejercicio anterior para agregar un método a cada clase de las figuras geométricas para calcular su perímetro.

|  |  |  |
| --- | --- | --- |
| **Figura Geométrica** | **Datos a Solicitar** | **Resultados a Mostrar** |
| Triangulo | 3 lados | Perímetro = lado1 + lado2 + lado3 |
| Cuadrado | lado | Perímetro = lado \* 4 |
| Círculo | radio | Perímetro = 2 \* PI \* radio |

**Ejercicio 16**. Desarrollar una aplicación que solicite ingresar un correo electrónico y una contraseña, luego llame a un método en otra clase para validar si el correo electrónico y la contraseña tienen el formato correcto. Debe haber un método para cada uno.

Para que sea correcto el formato del correo electrónico debe cumplir lo siguiente:

* Debe contener un caracter “@”.
* Al lado izquierdo del carácter “@” solo debe permitor letras, números y guiones.
* Al lado derecho del caracter “@” debe tener al menos un punto “.” Y después del último punto debe tener solo 2 o 3 caracteres.

Para que sea correcto el formato de la contraseña debe cumplir lo siguiente:

* Debe contener por lo menos una mayúscula.
* Debe contener por lo menos una minúscula.
* Debe contener por lo menos un número.
* Debe contener por lo menos un carácter especial.
* Debe tener una longitud minimo de 8 caracteres.

Mostrar un mensaje final indicando el resultado de las validaciones.