

Laboratorio 1.

Laboratorio Práctico

Objetivos.

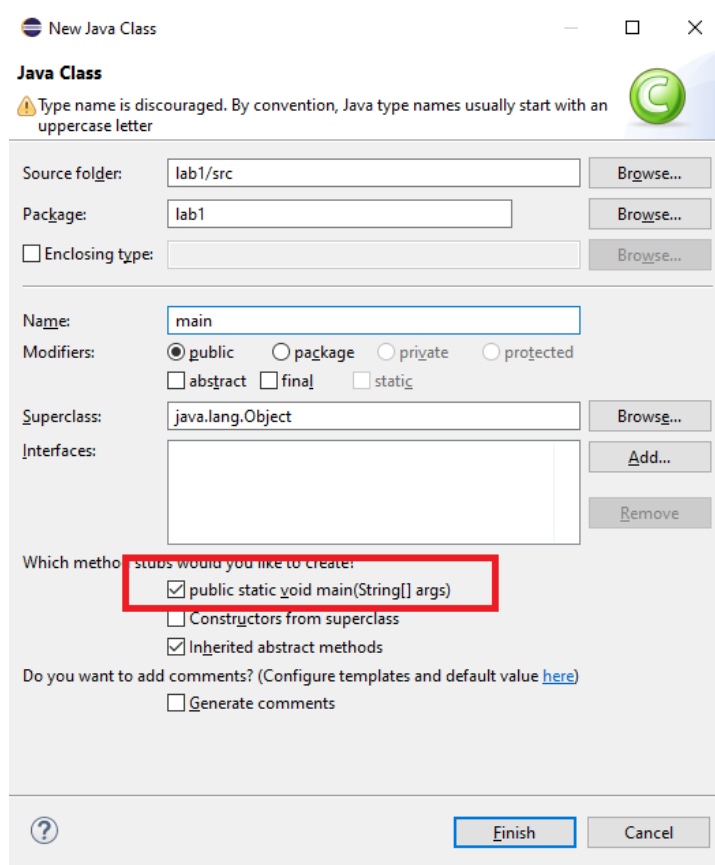
- Entender puntualmente los conceptos de clase e instancia
- Entender como funcionan los modificadores de acceso
- Entender como se comportan los paquetes
- Entender las clases estándares en java

Desarrollo

Paso 1.

Se va entender como se define una clase y como esta clase es instanciada.

Cree un nuevo proyecto de nombre laboratorio 1 en el workspace laboratorySis211. Y dentro de la carpeta src un nuevo archivo main.



Asegúrese de tener marcada la opción public static void main .. esta clase actuara como clase cliente de nuestra primera aplicación.

Hagamos un primer ejemplo.

Dentro de la clase main creada pruebe definiendo un mensaje de bienvenida.

```

package lab1;

public class main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.print("Hola Mundo");
    }

}

```

Deberá tener el mensaje

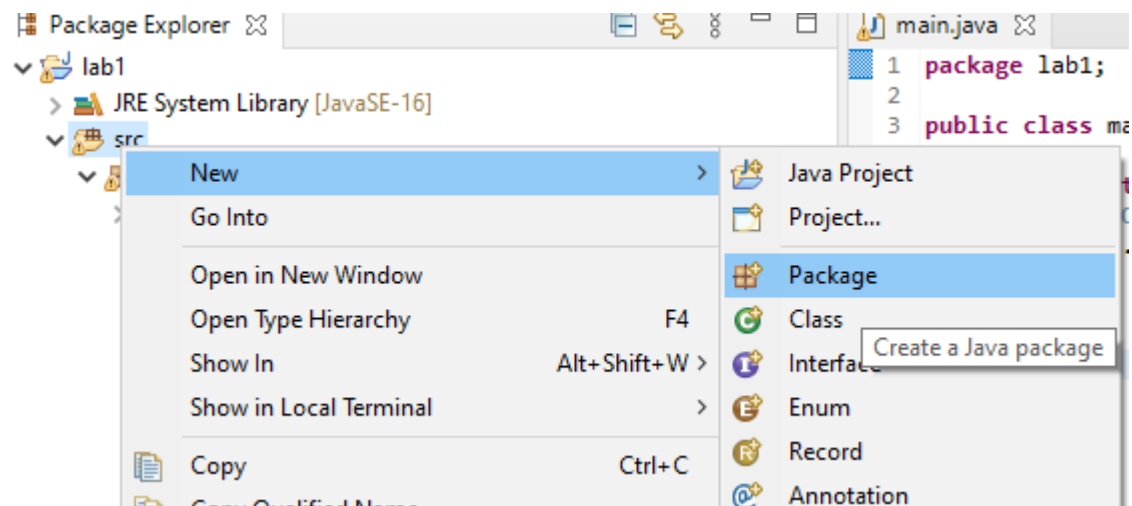
Hola Mundo

Paso 2. Clases he instancias.

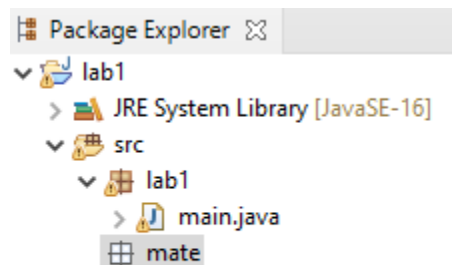
Vamos a crear una nueva clase.

Esta se llamará Mate, su trabajo será la de sumar, restar, multiplicar y dividir números reales.

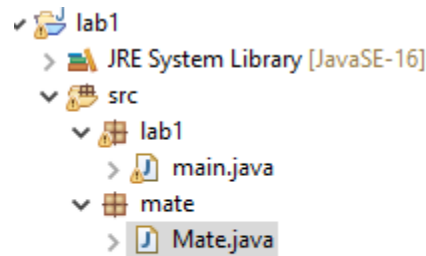
Para ello cree un nuevo paquete llamado, mate, y dentro del paquete la clase Mate.



Como se ve en la imagen.

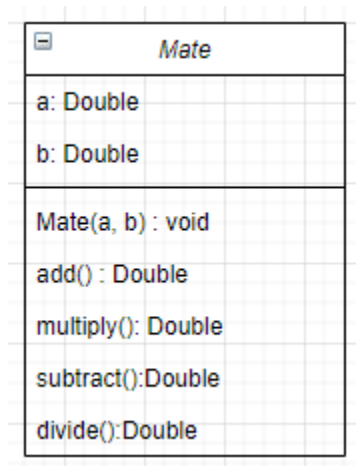


El resultado debería quedar de esa forma, nótese que el paquete mate está dentro de src.



Bien ahora dentro de clase mate vamos a crear nuestra primera clase, la idea es hacer una clase que pueda operar binarios, es decir dos números, en programación a estos operadores se les conocer como binario en este caso usaremos el + - / *.

Definiendo las variables necesarias, como se ve en siguiente diagrama de clases UML.



Esta es la estructura que necesitamos para nuestro objetivo, el atributo a, y b, deberán tener los modificadores de acceso en private. Y los métodos add, multiply subtract, divide, deberán ser públicos.

```
package mate;

public class Mate {
    private Double a;
    private Double b;
    public Mate (Double a, Double b) {
        this.a = a;
        this.b = b;
    }
    public Double add() {
        return a + b;
    }
    public Double subtract() {
        return a - b;
    }
}
```

```

    public Double multiply () {
        return a * b;
    }
    public Double divide() {
        return a / b;
    }
}

```

Vea la analogía de los diagramas UML, con el resultado esperado a nivel de programación, podríamos decir que los diagramas de clase definen un mapa de como se debe implementar una clase.

Ahora en la clase cliente main.

```

package lab1;

import mate.Mate;

public class main {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Test de instancias");
        Mate mat = new Mate(3.0, 5.0);
        Double r1 = mat.add();
        Double r2 = mat.subtract();
        Double r3 = mat.multiply();
        Double r4 = mat.divide();
        System.out.println("Suma " + r1 );
        System.out.println("Resta " + r2 );
        System.out.println("Multiplicación" + r3 );
        System.out.println("División " + r4 );

    }

}


```

Desafío 1.

Sobre el código agregue la función de leer los datos desde la consola. Para ellos use la clase Scanner, tenga en cuenta que los datos deberán ser casteado a Double.

Desafío 2.

Bien bajo esta misma idea agregue otra clase en el paquete mat, siguiendo el siguiente diagrama e clases UML.

 <i>BaseLog</i>
base: Integer
number: Integer
BaseLog(base, number) : vc
logBase() : Double

Java solo puede calcular logaritmos en base e, para una correcta solución deberá usar la siguiente formula.

$$\log_b(x) = \frac{\log_c(x)}{\log_c(b)}$$

En base a lo visto resuelva los siguientes problemas.

Para cada problema se deberá identificar, una clase, y los o métodos que sean necesarios par resolver el mismo.

Ejemplo 1 Aplicado.

Un pediatra requiere un programa para poder calcular prescripciones médicas pediátricas, el tiene una base de datos de diferentes medicamentos, donde tiene la dosis del fármaco (dosis ó día), y la concentración de la presentación, haciendo el proceso de ocultación, determina el peso del infante.

El doctor nos suministra la información que requerimos, la cual es (Dosis * Peso * ml) / (concentración de la presentación).

Ejemplo 2 Aplicado.

Nos encontramos trabajando en la fábrica de autos Toyota en la sección de pruebas de choques, se nos encomienda realizar una función que permita calcular la fuerza de impacto de un vehículo. Para ellos nos dicen que nos pasarán como parámetros la masa del vehículo, la aceleración constante, y la distancia donde se encuentra el muro de choques.

Uso matemático para resolver el problema.

Problema 3 Aplicado.

Estamos trabajando para el departamento de defensa sección inteligencia, en el área de simulaciones, nos piden construir un programa que sea capaz de detectar si un intruso es alcanzado por un radar, para ellos nos fan la potencia del radar, que es equivalente al radio el punto de origen, y las coordenadas del intruso.