



## Práctica 2

### Enunciado

Escriba la clase **PilaMin** como subclase de la clase **PilaDeEnteros**. Las operaciones deben tener complejidad **O(1)**.

### Especificación de los métodos de PilaMin:

- **Integer min()**: retorna, sin eliminar, el elemento menor de la pila.
- **void poner(elem)**: agrega elem a la pila (sobre-escribe el de **PilaDeEnteros**).
- **Integer sacar()**: elimina y devuelve el elemento en el tope de la pila (sobreescribe el de **PilaDeEnteros**).



## Práctica 2

```
public class PilaMin extends PilaDeEnteros {
    private PilaDeEnteros minima = new PilaDeEnteros();

    private Integer min() {
        return minima.tope();
    }

    public void poner(Integer e) {
        super.poner(e);
        if (minima.esVacia() || e <= min())
            minima.poner(e);
    }

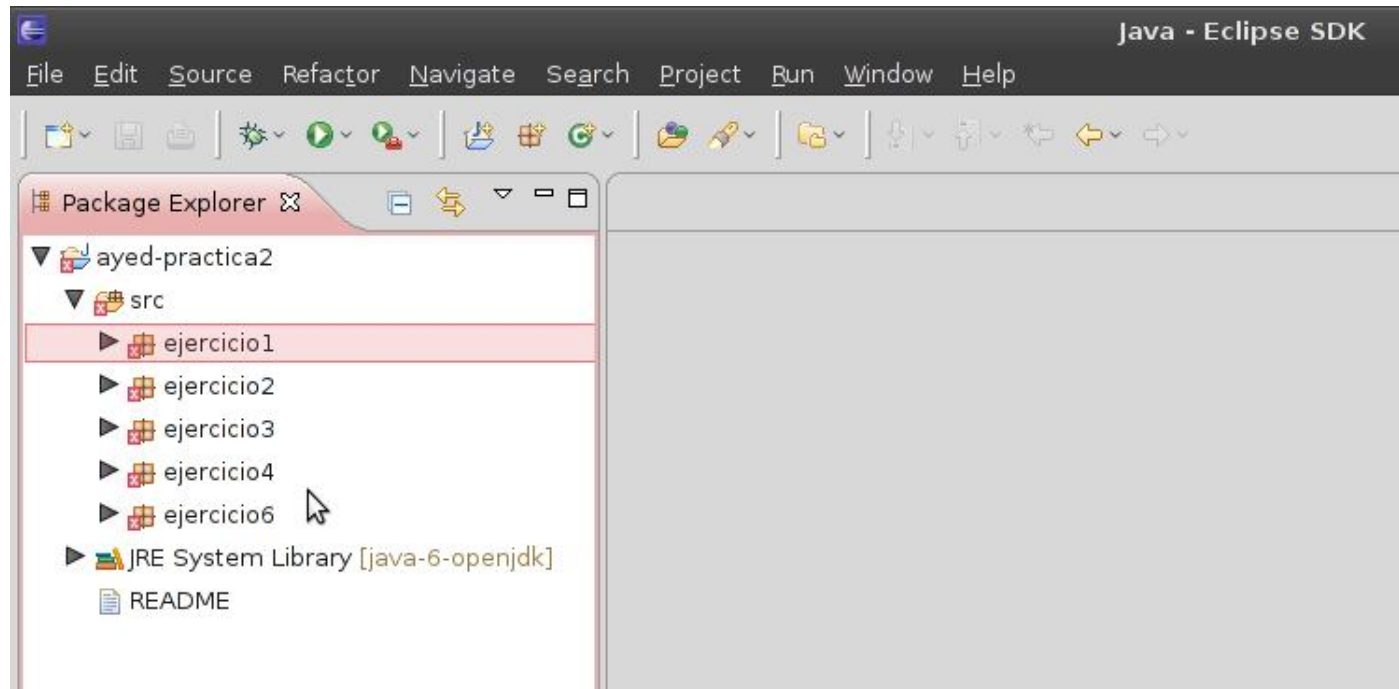
    public Integer sacar() {
        /** Se saca el tope de la pila original. Luego se
        •verifica si el valor del tope es igual al tope de la * pila de mínimos y si lo es, también se saca.
        •*/
        Integer e = null;
        e = super.sacar();
        if (e.equals(min()))
            minima.sacar();
        return e;
    }

    public String toString() {
        return super.toString() + "\n" + "Mínimos: "
            + minima.toString();
    }
}
```



# JUnit

Una vez creado el proyecto e importado el código:

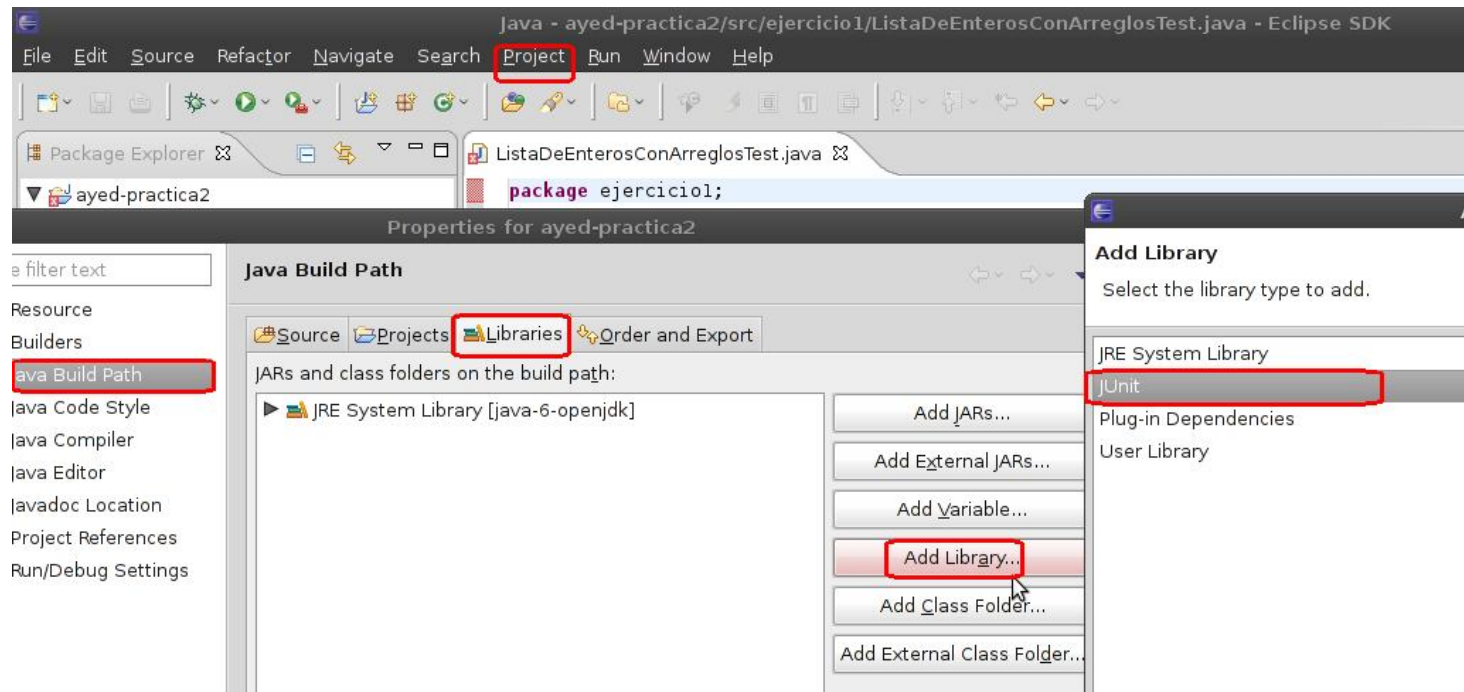




# JUnit

Debemos agregar las librerías de JUnit para que funcionen los Tests Cases:

**En Project → Properties → Java Build Path → Libraries → Add Library ...**

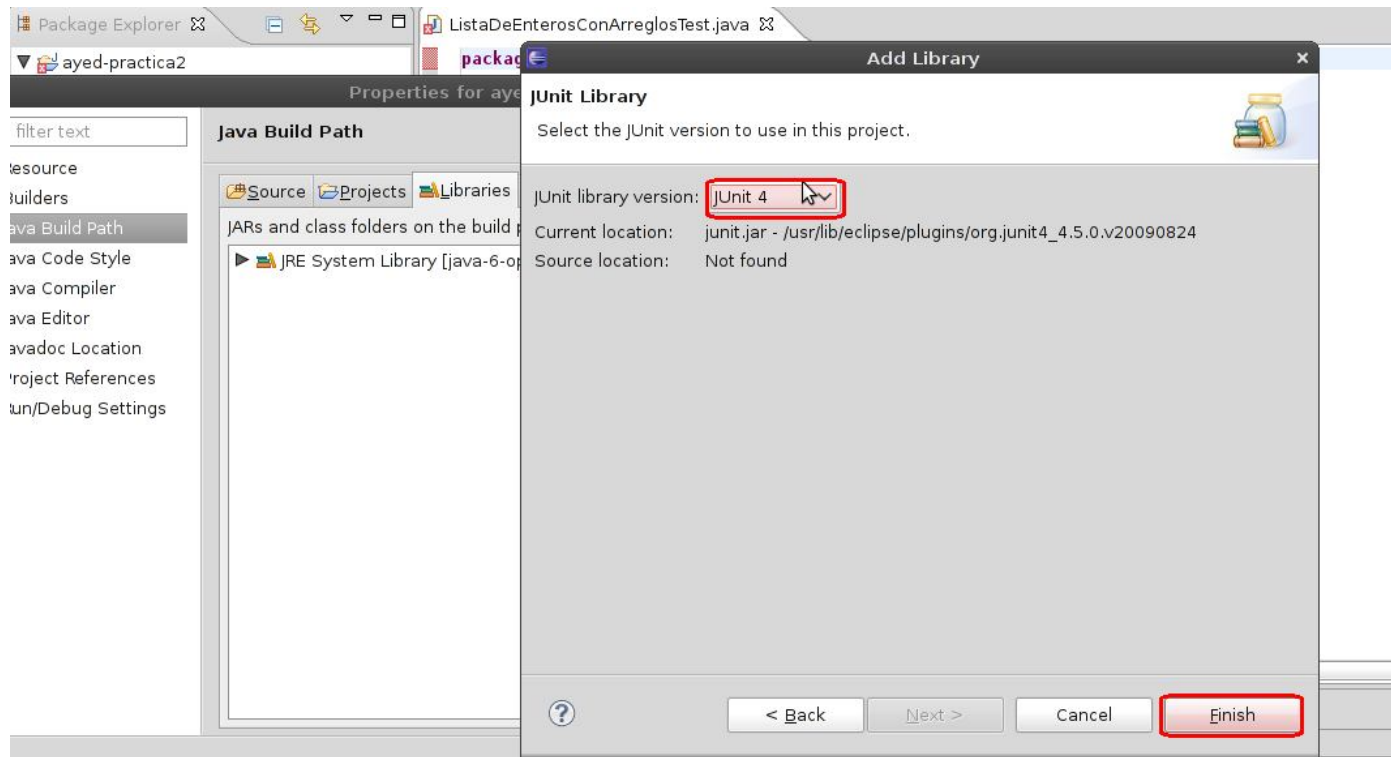




# JUnit

Debemos agregar las librerías de JUnit para que funcionen los Tests Cases:

**Seleccionamos JUnit 4 y terminamos.**

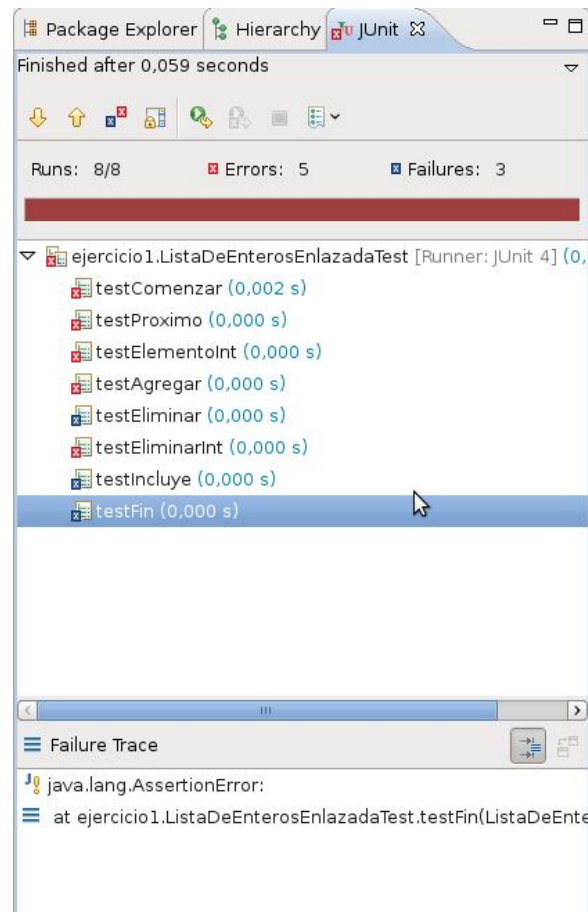




# JUnit

## Ejecutando los Tests Cases:

Un test fallido se vería así:





# JUnit

## Ejecutando los Tests Cases:

Seleccionando la clase del Test: **Run** → **Run as..** → **JUnit Test**

Un test exitoso se vería así:

