

```

package clase18;

import java.util.ArrayList;
import java.util.List;

public class Contenedor<T> {
    private List<T> lista = new ArrayList<>();

    public void guardar(T elemento) {
        lista.add(elemento);
    }

    public void mostrarPorClase(T clase) {
        lista.forEach(e -> {
            if (e.getClass() == clase) {

                if (clase == Estudiante.class) {
                    System.out.println("Nombre: " + ((Estudiante) e).getNombre());
                    System.out.println("Calificacion: " + ((Estudiante)
e).getCalificacion());
                }

                if (clase == Pareja.class) {
                    System.out.println("Elemento 1: " + ((Pareja<?, ?>)
e).getElemento1());
                    System.out.println("Elemento 2: " + ((Pareja<?, ?>)
e).getElemento2());
                }
                System.out.println("-----");
            }
        });
    }

    public double sumarNumeros() {
        final double[] total = {0};

        lista.forEach(e -> {
            if (e.getClass() == Byte.class) total[0] += (Byte) e;
            if (e.getClass() == Short.class) total[0] += (Short) e;
            if (e.getClass() == Integer.class) total[0] += (Integer) e;
            if (e.getClass() == Long.class) total[0] += (Long) e;
            if (e.getClass() == Float.class) total[0] += (Float) e;
            if (e.getClass() == Double.class) total[0] += (Double) e;
        });
        System.out.println(total[0]);
        return total[0];
    }

    public ArrayList<Empleado> filtrarEmpleados(ArrayList<Empleado> lista, double
salarioMinimo){
        ArrayList<Empleado> empleados = new ArrayList<Empleado>();

```

```

        lista.forEach(empleado ->{
            if(empleado.getSalario() > salarioMinimo) empleados.add(empleado);
        });

        return empleados;
    }
}

```

```
package clase18;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
public class ContenedorString {
    private List<String> lista = new ArrayList<String>();

    public void guardar(String elemento) {
        lista.add(elemento);
    }

    public void mostrarPorLongitud(int limite){
        lista.forEach(elemento ->{
            if(elemento.length() > limite) System.out.println(elemento);
        });
    }
}

```

```
import clase18.Operador;
```

```
import java.util.LinkedList;
```

```
public class Main {
    public static void main(String[] args) {
        Operador operador = new Operador();
        LinkedList<Integer> lista1 = new LinkedList<>();
        LinkedList<Integer> lista2 = new LinkedList<>();

        lista1.add(1);
        lista1.add(2);
        lista1.add(3);

        lista2.add(2);

        System.out.println(operador.compararListas(lista1, lista2));
    }
}

```

```

package clase18;

public class Estudiante {
    private String nombre;
    private float calificacion;

    public Estudiante(String nombre, float calificacion) {
        this.nombre = nombre;
        this.calificacion = calificacion;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public float getCalificacion() {
        return calificacion;
    }

    public void setCalificacion(float calificacion) {
        this.calificacion = calificacion;
    }
}

```

```

package clase18;

import java.util.Collections;
import java.util.Iterator;
import java.util.LinkedList;

public class Operador<T> {
    public LinkedList<T> invertir(LinkedList<T> lista) {
        Collections.reverse(lista);
        return lista;
    }

    public LinkedList<Integer> eliminarPares(LinkedList<Integer> lista) {
        LinkedList<Integer> aux = new LinkedList<Integer>();
        Iterator it = lista.iterator();
        while (it.hasNext()) {
            int val = (int) it.next();
            if (val % 2 != 0) aux.add(val);
        }
        return aux;
    }
}

```

```

    }

    public boolean compararListas(LinkedList<T> lista1, LinkedList<T> lista2) {
        Iterator it;
        boolean igual = true;
        LinkedList<T> objetivo;
        int contador = 0;
        if (lista1.size() > lista2.size()) {
            it = lista2.iterator();
            objetivo = lista1;
        } else {
            it = lista1.iterator();
            objetivo = lista2;
        }

        while (it.hasNext()) {
            Object e = it.next();
            if (objetivo.get(contador) != e) igual = false;
        }

        return igual;
    }
}

```

```
package clase18;
```

```

public class Pareja<T,U> {
    private T elemento1;
    private U elemento2;

    public Pareja(T elemento1, U elemento2) {
        this.elemento1 = elemento1;
        this.elemento2 = elemento2;
    }

    public T getElemento1() {
        return elemento1;
    }

    public void setElemento1(T elemento1) {
        this.elemento1 = elemento1;
    }

    public U getElemento2() {
        return elemento2;
    }

    public void setElemento2(U elemento2) {
        this.elemento2 = elemento2;
    }
}

```

