



UNIVERSIDAD AUTONOMA DE NUEVO LEÓN
FACULTAD DE CIENCIAS FISICO MATEMATICAS

“Practica 7”

UNIDAD DE APRENDIZAJE: Programación Orientada a
Objetos

MAESTRO: JORGE ALBERTO ISLAS PINEDA

CARRERA: LICENCIATUA EN CIENCIAS
COMPUTACIONALES

Output:

```
run:
Navegando por mar... ??
Transportando 60 pasajeros en Marítimo [BAR-01]
Mantenimiento marino: revisión de casco y motores.
---
Rodando en carretera... ??
ERROR: Pasajeros (60) exceden capacidad (50) en Terrestre
Mantenimiento terrestre: llantas y frenos.
---
Volando en el aire... ??
Transportando 60 pasajeros en Aéreo [AV-777]
Mantenimiento aéreo: inspección de turbinas y sistemas.
---
Circulando por vías férreas... ??
Transportando 60 pasajeros en Ferroviario [TRN-21]
Mantenimiento ferroviario: revisión de rieles y vagones.
---
BUILD SUCCESSFUL (total time: 0 seconds)
```

• Practica7

```
package Runpractica7;
```

```
/***
*
* @author garza
*/
import java.util.ArrayList;

public class Practica7 {
    public static void main(String[] args) {
        ArrayList<Transporte> lista = new ArrayList<>();
        lista.add(new TransporteMaritimo("BAR-01", 80));
        lista.add(new TransporteTerrestre("BUS-33", 50));
        lista.add(new TransporteAereo("AV-777", 120));
        lista.add(new TransporteFerroviario("TRN-21", 200)); // Nuevo tipo
    }
}
```

```

int pasajeros = 60;

for (Transporte t : lista) {
    t.mover();
    try {
        t.transportar(pasajeros);
    } catch (CapacidadExcedidaException e) {
        System.out.println("ERROR: " + e.getMessage());
    }
    if (t instanceof Operable) {
        ((Operable) t).realizarMantenimiento();
    }
    System.out.println("---");
}
}
}
}

```

- **CapacidadExcedidaException**

```

package Runpractica7;

/**
 *
 * @author garza
 */
public class CapacidadExcedidaException extends Exception {
    public CapacidadExcedidaException(String mensaje) {
        super(mensaje);
    }
}

```

- Operable

```
package Runpractica7;
```

```
/**  
 *  
 * @author garza  
 */  
  
public interface Operable {  
    void realizarMantenimiento();  
}
```

- Transporte

```
package Runpractica7;
```

```
/**  
 *  
 * @author garza  
 */  
  
public abstract class Transporte {  
    private String id;  
    private int capacidad;  
  
    public Transporte(String id, int capacidad) {  
        setId(id);  
        setCapacidad(capacidad);  
    }  
  
    public String getId() {  
        return id;  
    }
```

```
}
```

```
public void setId(String id) {
```

```
    this.id = id;
```

```
}
```

```
public int getCapacidad() {
```

```
    return capacidad;
```

```
}
```

```
public void setCapacidad(int capacidad) {
```

```
    if (capacidad <= 0) {
```

```
        throw new IllegalArgumentException("La capacidad debe ser mayor a 0");
```

```
}
```

```
    if (capacidad > 500) {
```

```
        throw new IllegalArgumentException("La capacidad no puede superar 500");
```

```
}
```

```
    this.capacidad = capacidad;
```

```
}
```

```
public abstract String tipo();
```

```
public abstract void mover();
```

```
public void transportar(int pasajeros) throws CapacidadExcedidaException {
```

```
    if (pasajeros > capacidad) {
```

```
        throw new CapacidadExcedidaException("Pasajeros (" + pasajeros +
") exceden capacidad (" + capacidad + ") en " + tipo());
```

```
}
```

```
        System.out.println("Transportando " + pasajeros + " pasajeros en " + tipo() + " [" + id +
    "]");
    }
}
```

- **TransporteAereo**

```
package Runpractica7;
```

```
/**
```

```
*
```

```
* @author garza
```

```
*/
```

```
public class TransporteAereo extends Transporte implements Operable {
```

```
    public TransporteAereo(String id, int capacidad) {
```

```
        super(id, capacidad);
```

```
}
```

```
@Override
```

```
    public String tipo() {
```

```
        return "Aéreo";
```

```
}
```

```
@Override
```

```
    public void mover() {
```

```
        System.out.println("Volando en el aire... ");
```

```
}
```

```
@Override
```

```
public void realizarMantenimiento() {  
    System.out.println("Mantenimiento aéreo: inspección de turbinas y sistemas.");  
}  
}  
}
```

- TransporteFerroviario

```
package Runpractica7;
```

```
/**  
 *  
 * @author garza  
 */
```

```
public class TransporteFerroviario extends Transporte implements Operable {
```

```
    public TransporteFerroviario(String id, int capacidad) {  
        super(id, capacidad);  
    }
```

```
    @Override  
    public String tipo() {  
        return "Ferroviario";  
    }
```

```
    @Override  
    public void mover() {  
        System.out.println("Circulando por vías férreas... 🚂");  
    }
```

```
    @Override
```

```
public void realizarMantenimiento() {  
    System.out.println("Mantenimiento ferroviario: revisión de rieles y vagones.");  
}  
}
```

- Transporte Marítimo

```
package Runpractica7;
```

```
/**  
 *  
 * @author garza  
 */
```

```
public class TransporteMaritimo extends Transporte implements Operable {
```

```
    public TransporteMaritimo(String id, int capacidad) {  
        super(id, capacidad);  
    }
```

```
    @Override  
    public String tipo() {  
        return "Marítimo";  
    }
```

```
    @Override  
    public void mover() {  
        System.out.println("Navegando por mar... ");  
    }
```

```
    @Override
```

```
public void realizarMantenimiento() {  
    System.out.println("Mantenimiento marino: revisión de casco y motores.");  
}  
}
```

- Transporte Terrestre

```
package Runpractica7;
```

```
/**  
 *  
 * @author garza  
 */
```

```
public class TransporteTerrestre extends Transporte implements Operable {
```

```
    public TransporteTerrestre(String id, int capacidad) {  
        super(id, capacidad);  
    }
```

```
    @Override  
    public String tipo() {  
        return "Terrestre";  
    }
```

```
    @Override  
    public void mover() {  
        System.out.println("Rodando en carretera... 🚕");  
    }
```

```
    @Override
```

```
public void realizarMantenimiento() {  
    System.out.println("Mantenimiento terrestre: llantas y frenos.");  
}  
}
```