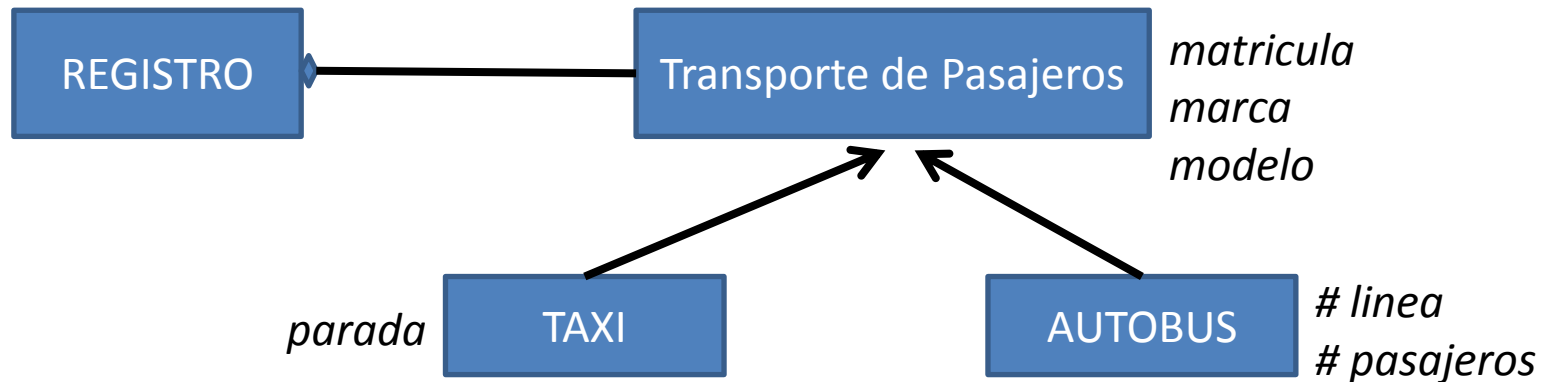




```
class Taxi extends TransporteDePasajeros {
    String parada;
    Taxi (...) {
        super(...);
        ...
    }
    void setParada(..)
    {...}
    String getParada()
    {...}
}
```


```
class Autobus extends TransporteDePasajeros{
    int linea;
    int pasajeros;
    Autobus (...) {
        super(...);
        ...
    }
    void setLinea(..)
    {...}
    int getLinea()
    {...}
    void setPasajeros(..)
    {...}
    int getPasajeros()
    {...}
}
```



El municipio lleva un registro de transportes para calcular los impuestos. Algunas de las operaciones habituales son:

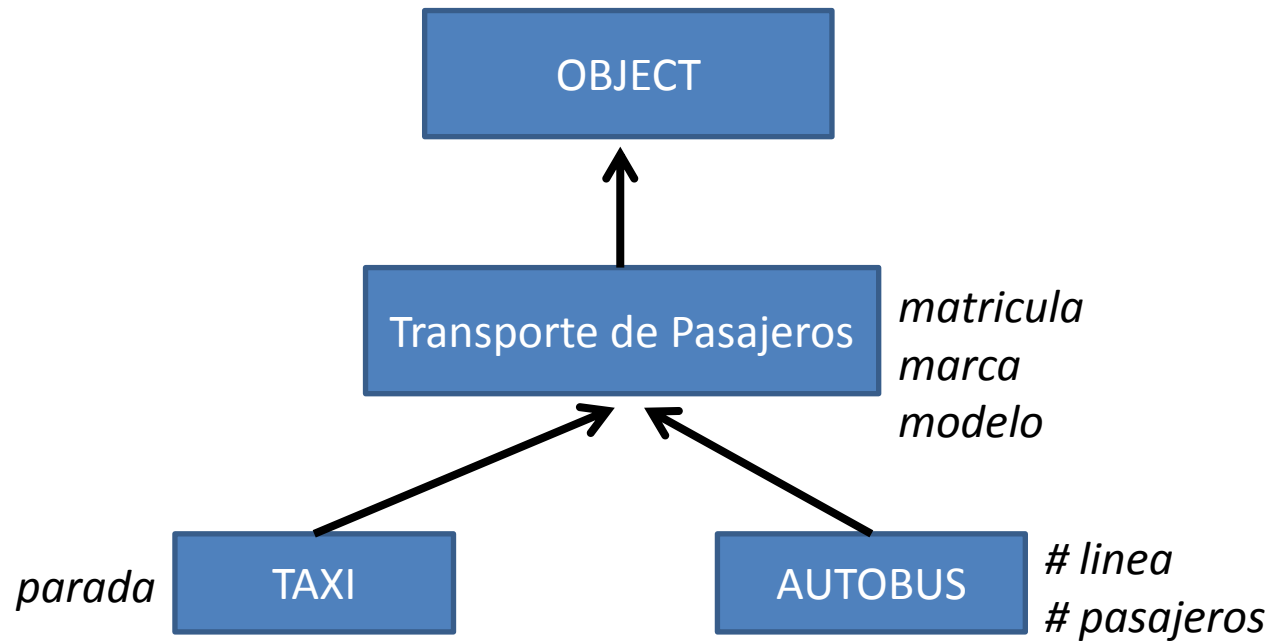
- Recuperar los datos del transporte a partir de la matricula
- Listar todos los taxis de una parada
- Calcular la cantidad de taxis registrados.



```
class Registro {  
    TransporteDePasajeros [] trans;  
    int s;  
  
    Registro()  
    { trans = new TransporteDePasajeros [10];  
      s=0;}  
  
    void agregar(TransporteDePasajeros t)   
    { trans[s] = t;  
      s++; }  
  
    void buscarPorMatricula (String m)  
    for(int i; i<s; i++)  
        if (m.equals(trans[i].getMatricula())   
            SOP(aux.get....);  
    }  
}
```

```
int cantTaxi() {  
    int aux=0;  
    for(int i; i<s; i++)  
        if (trans[i] instanceof Taxi)   
            aux++;  
    return aux;  
}
```

```
void buscarTaxiParada (String p) {  
    Taxi aux;  
    for(int i; i<s; i++)  
        if (trans[i] instanceof Taxi)   
        { aux = (Taxi) trans[i];   
          if (p.equals(aux.getParada()))  
              SOP(aux.get....);  
        }  
}
```



```
import java.util.Vector;
```

```
class Registro {  
    Vector trans;
```

```
    Registro()  
    { trans = new Vector(); }
```

```
    void agregar(TransporteDePasajeros t)  
    { trans.addElement(t); }
```

```
    void buscarPorMatricula (String m)  
    {  
        TransporteDePasajeros t;  
        for(int i; i<trans.size(); i++)  
        { t = (TransporteDePasajeros) trans.elementAt[i];  
          if (m.equals(t.getMatricula()))  
              SOP(aux.get....);  
        }  
    }
```

```
    int cantTaxi() {  
        int aux=0;  
        for(int i; i<trans.size(); i++)  
            if (trans.elementAt(i) instanceof Taxi)  
                aux++;  
        return aux;  
    }
```

```
    void buscarTaxiParada (String p) {  
        Taxi aux;  
        for(int i; i<trans.size(); i++)  
            if (trans.elementAt(i) instanceof Taxi)  
            { aux = (Taxi) trans.elementAt(i);  
              if (p.equals(aux.getParada()))  
                  SOP(aux.get....);  
            }  
    }
```