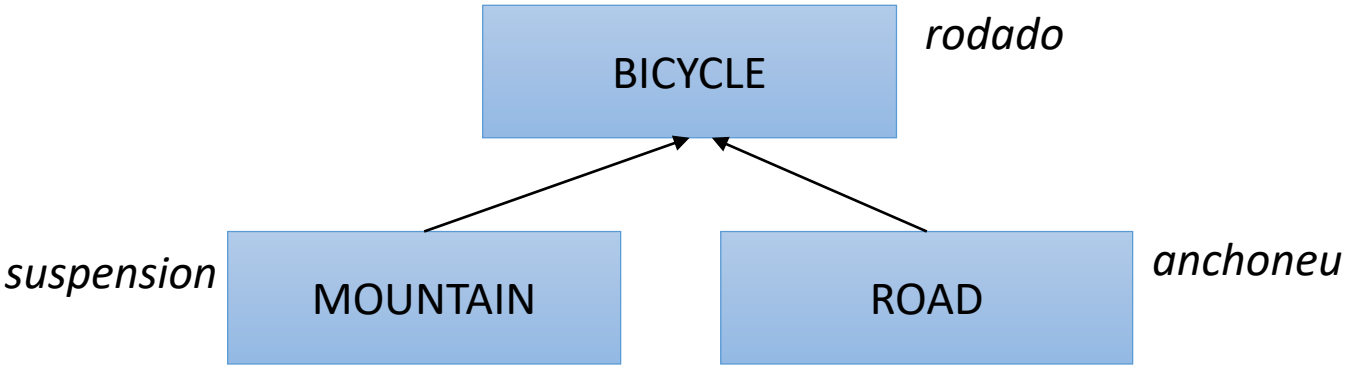
Una bicicleta es un vehículo de transporte personal de propulsión humana. Las bicicletas Mountain Bike tienen suspensión y en las Road Bike lo importante es el ancho de los neumáticos. Pero todas tienen un nº de rodado.

Mountain Bike ***es una*** bicicleta
Road Bike **es una** bicicleta




```
class Bicycle {
    int rodado;

  Bicycle (int rodado) {
      this.rodado = rodado;
   }

  int getRodado()   {
      return this.rodado;
   }

void setRodado (int rodado) {
      this.rodado = rodado;
   }
```

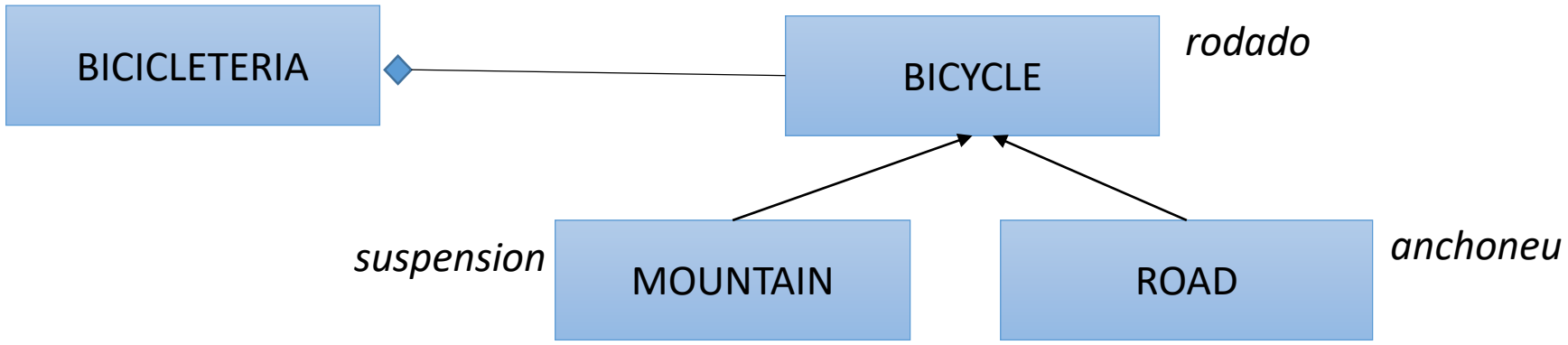
```
class Mountain extends Bicycle {
    String suspension;

  Bicycle (int rodado, String sus) {
      super(rodado);
      this.suspensión = sus;
   }

  String getSuspension()   {
      return this.suspension;
   }

void setSuspension (String sus)  {
      this.suspension = sus;
   }
```

```
class Road extends Bicycle {
    float anchoneu;

  Bicycle (int rodado, float ancho){
      super(rodado);
      this.anchoneu = ancho;
   }

  float getAnchoneu()   {
      return this.anchoneu;
   }

void setAnchoneu(float ancho)  {
      this.anchoneu = ancho;
   }
```

Una bicicleterìa vende todo tipo de bicicletas. Operaciones necesarias para satisfacer las consultas de los clientes sin necesidad de revisar el deposito son:

- Listar las bicicletas de un determinado rodado e indicar su tipo
- Listar todas las bicicletas Mountain de un determinada suspensión y rodado
- Total de bicicletas por tipo

Bicicleteria **tiene** bicicletas



```
class Bicicleteria {
    Vector bicis;

  Bicicleteria () {
      bicis = new Vector ();
  }

  void addBike(Bicycle b)   {
      bicis.addElement(b)
   }
```

```
// Listar las bicicletas de un determinado rodado e indicar su tipo

 void listarRodado(int rodado) {
      Bicycle b;

      for (int i=0; bicis.size(); i++)
      { b = (Bicycle) bicis.elementAt(i);
        if (b.getRodado() == rodado)
            if (b instanceof Road)
                S.O.P (“road”);
            else
                S.O.P. (“Mountain”);
       }
}
```

```
class Bicicleteria {
    Vector bicis;

  Bicicleteria () {
      bicis = new Vector ();
  }

  void addBike(Bicycle b)   {
      bicis.addElement(b)
   }
```

```
// Listar todas las bicicletas Mountain de un
determinada suspensión y rodado

 void listarMountain(int rodado, String sus) {
    Mountain m;
    for (int i=0; bicis.size(); i++)
       if (bicis.elementAt(i) instanceof Mountain)
          { m = (Mountain) bicis.elementAt(i);
            if (m.getRodado()==rodado &&
                   m.getSuspension()== sus)
                S.O.P (“ -----”);
          }
}
```

```
class Bicicleteria {
    Vector bicis;

  Bicicleteria () {
      bicis = new Vector ();
  }

  void addBike(Bicycle b)   {
      bicis.addElement(b)
   }
```

```
//   Total de bicicletas por tipo

 void totalPorTipo() {
      Bicycle b;
      int tr=0; tm=0;
      for (int i=0; bicis.size(); i++)
      { b = (Bicycle) bicis.elementAt(i);
        if (b instanceof Road)
                tr++;
            else
                tm++;
         }
        S.O.P (…..);
       }
}
```