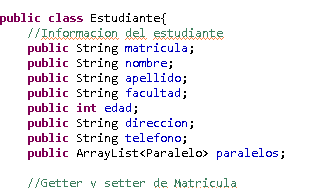
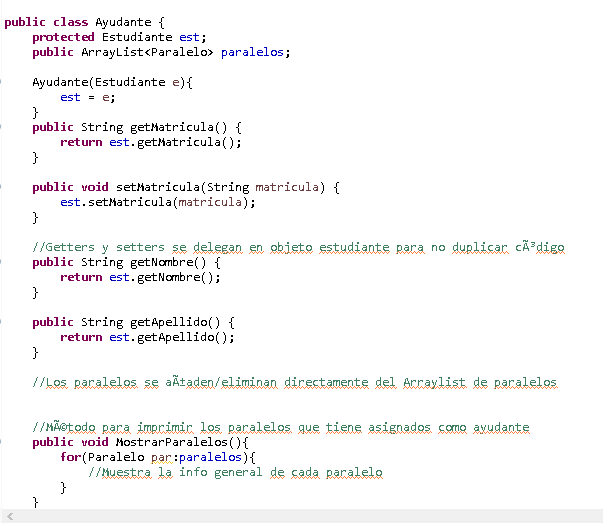
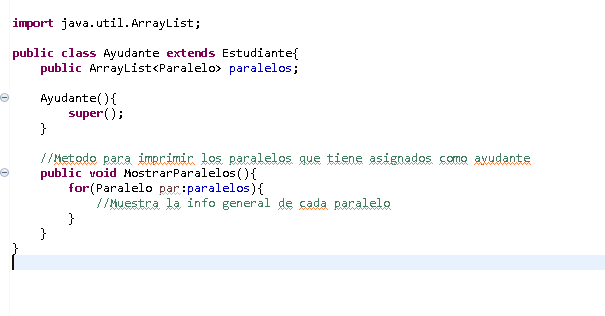
**Code Smell: Inappropiate Intimacy**  
En este caso la clase Ayudante necesita acceder a los parámetros de la clase Estudiante para su funcionamiento generando un acoplamiento entre las clases.

**Consecuencias:** Esto genera que el desarrollador no pueda entender de una manera correcta ninguna de las dos clases. En caso de que se desee realizar un cambio en la clase Estudiante afectaría directamente a la clase Ayudante.



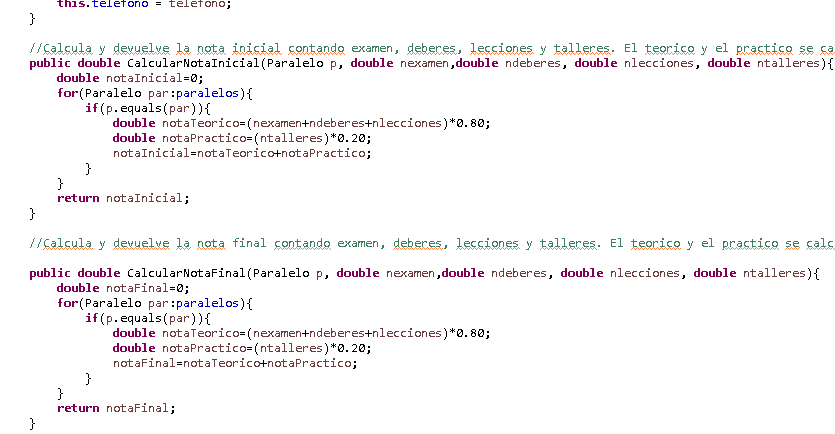
**Refactor.**

En este caso lo más optimo seria extraer la clase y hacer uso de la herencia, se toma en cuenta al ayudante como un caso “especial” de estudiante haciendo que extienda directamente de dicha clase.



**Code Smell: Long Parameter List**

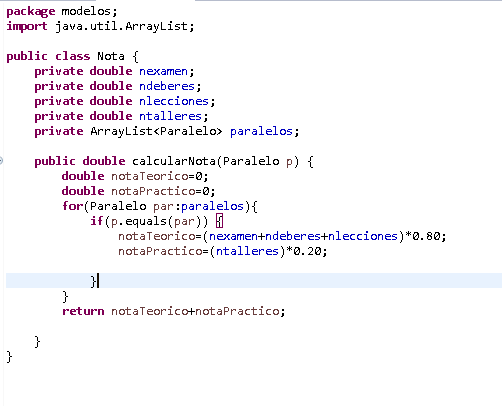
**Consecuencia:** Los métodos para calcular notas tienen demasiados parámetros lo que hace inentendible el método.



**Refactor.**

Lo ideal es crear una clase que se encargue de maneja cada parámetro del método calcular notas, para que sea más sencillo acceder a ellas y evitar la sobrecarga en el método.

**Nota.java**



**Estudiante.java**

