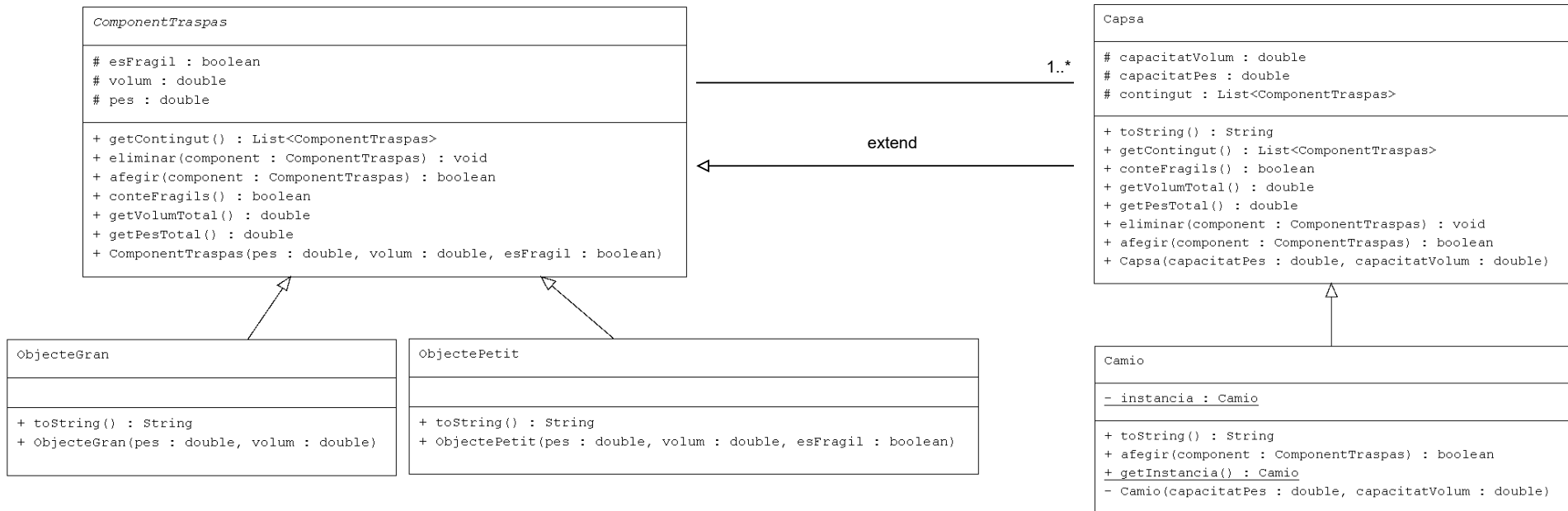


## Exercici 2:

2.a. Utilitzem el patró Composite en aquest exemple perquè tracta d'una jerarquia on posem objectes dins de capsas o del camió, ens permet tractar de manera uniforme tant els objectes individuals (fulles que poden ser objectes petits o grans) com les col·leccions d'objectes (compostos com capsas i el camió), i mantenint el codi modular, escalable i fàcil d'estendre.

2.b. Els membres són:

- ComponentTraspas és una classe abstracta amb els atributs comuns per tots els elements (pes, volum i fràgil).
- ObjecteGran és una fulla, és una classe que estén la classe abstracta i les instàncies es posaran dins del camió.
- ObjectePetit és una fulla, és una classe que estén la classe abstracta i les instàncies es posaran dins de capsas.
- Capsa és un compost, és una classe que estén la classe abstracta i conté un llistat amb objectes petits, es controla el pes, el volum i fragilitat.
- Camió és un compost, és una classe que estén la classe capsa, però que serà singleton (un únic camió), sobreescriu el mètode afegir per només permetre instàncies de capsas i objectes grans.



2.c. Com gestionem els conceptes fràgils, pes i volum:

- El mètode afegir de la classe Capsa comprova que sigui un objecte petit (tampoc pot ser null) i comprova si hi ha espai (la suma del pes i volum afegint el nou objecte petit no sobrepassa la capacitat), també comprova que o bé la capsa és buida (i es marcarà com a fràgil o no) o bé la marca de la capsa coincideix amb el tipus d'objecte.
- El mètode afegir de la classe camió comprova que el nou element sigui un objecte gran o una capsa (tampoc pot ser null ni ell mateix) i comprova si hi ha espai (la suma del pes i volum afegint el nou element no sobrepassa la capacitat).