2021-09-09

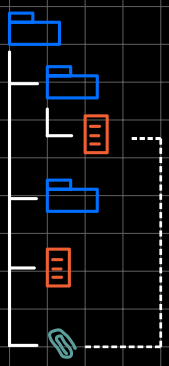
Esercizio 2 (9 punti)

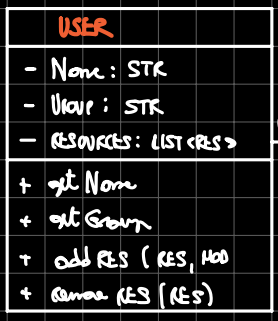
Si consideri la struttura tipica di un *file system*. Le directory sono organizzate gerarchicamente: ogni directory può contenere altre directory, file, oppure link. Un link è un riferimento a un file fisicamente memorizzato in un'altra directory; in questo modo il file riferito diventa virtualmente parte anche della directory che contiene il link. Una directory ha un nome; ogni file è caratterizzato da un nome, una dimensione e un tipo. Un link ha un nome e un tipo.

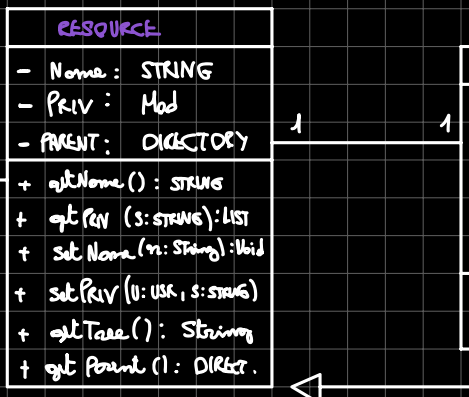
Ogni elemento (directory, file o link) è associato con un insieme di diritti d'accesso: lettura, scrittura e esecuzione. Questi sono concessi al proprietario di una risorsa (un singolo utente), a gruppi di utenti o a tutti gli utenti.

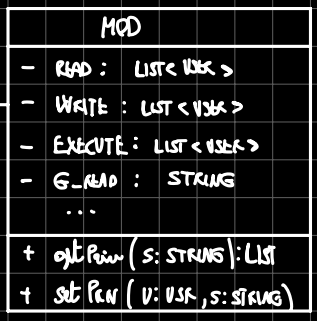
- Si modelli il problema descritto con un diagramma delle classi UML, evidenziando i metodi e gli attributi principali delle diverse classi.
- In funzione delle classi identificate nel diagramma precedente, si scriva il corpo del metodo `stampaNomeCompleto` per scrivere a video il nome completo di un file partendo dalla radice del file system.
- Si scriva l'invariante privato della classe `Directory` per dire che un suo oggetto non può mai contenere più di 200 file e deve esistere almeno un utente in grado di leggere il contenuto della directory.

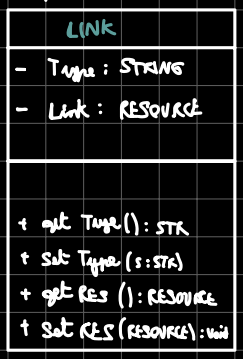
Se si ritiene che la descrizione informale contenga ambiguità, si descriva a parole come il diagramma UML proposto risolve tale ambiguità.












```
String getPath() {
    String str = this.getName();

    while (this.getParent() != null)
        str = getParent().getName() + "/" + str

    return str
}
```

```
/* @ Public Invariant ( \forall DIRECTORY Dir; dir != null; this.getRes().size <= 200
                        && (\exists USR USR; USR != null; USR.getRES().contains(this)
                            && this.getPriv("read").contains(USR)
                           )
                      );
   @
   @
   @
   @
   @
*/
```