

Beste Benjamin,

Je scoorde 7/10 voor Labo 5. Graag geven we nog volgende feedback mee:

- Weet dat interfaces ook kunnen overerven van andere interfaces. Je kon dus Content laten extenden van NoteElement (wat je deed) en dan CheckList en Message Content laten implementeren, zonder opnieuw NoteElement te implementeren (dit gebeurde al impliciet door de relatie tussen Content en NoteElement).
- Bij het aanmaken van je FileStorageMap lees je alle files in als string en gebruik je gans die string als key in je map. Het is eerder de bedoeling dat je het *id* uitleest van de gedeserialiseerde Note en die als key gebruikt. (uniek, en een stuk kleiner en minder gevoelig aan eventuele mismatches door andere spaties of zo)
- De bedoeling van de invoke() methodes was om enkel de stukken die betrekking hebben tot de filestorage effectief te implementeren. Op het einde roep je immers de standaard Map methode op met method.invoke(proxiedMap, args), dus het verwijderen of vervangen in de proxyMap gebeurt daar al.
- Verder kan je bij een map gewoon rechtstreeks de key gebruiken om de juist entry te manipuleren (verwijderen/vervangen). Nu itereer je over alle files, op zoek naar de juiste, terwijl je eigenlijk al weet welke (key) te gebruiken. Bijvoorbeeld:

```
File file = new File(directory, args[0]+".txt");  
file.delete();
```