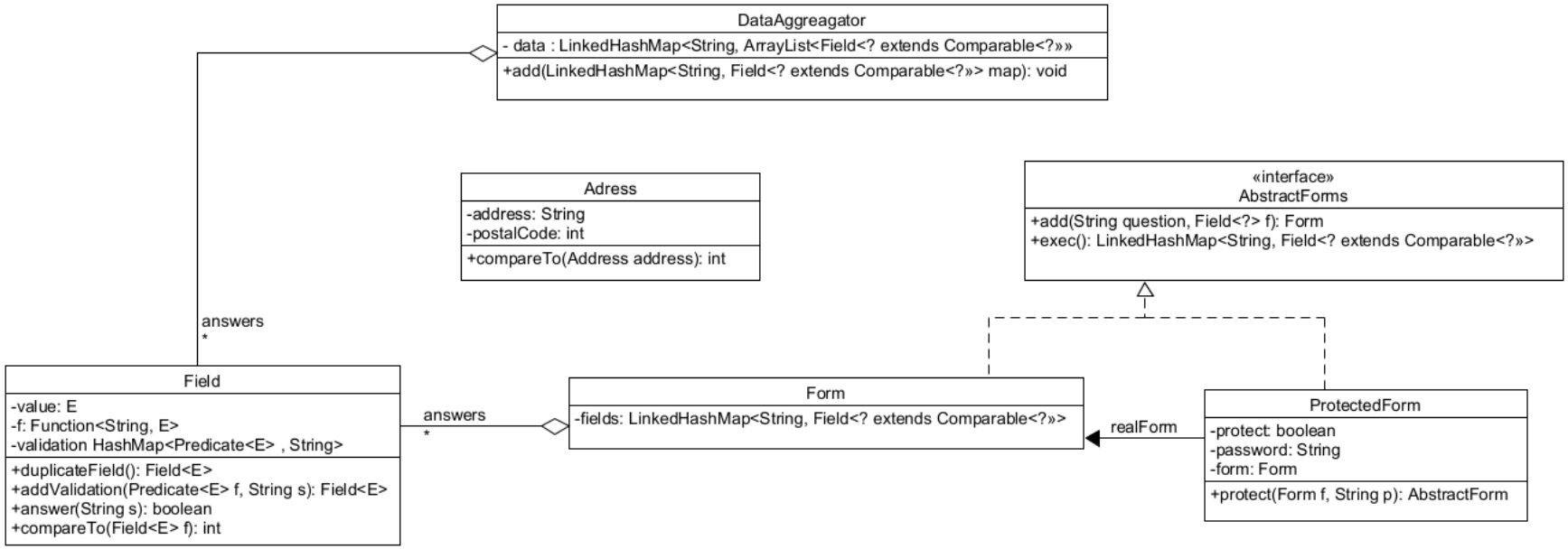
Assignment 5

Inheritance, interfaces and exceptions

28/04/21
ADSOF 2292
Daniel Varela
Guillermo Martín-Coello

Class diagram:



Decisions and problems through the development:

In the first section we first thought about the decision of creating different interfaces to create a more modular program. This decision was scrapped because we realise that the classes we needed to implement had completely different functionalities so it wasn't necessary to establish a common interface between them. We also designed the exec function of the form class to return the output it creates as a String, but we later realised this was not a good decision and changed it. A big problem we had was about the Scanner class we decided to use to read the terminal input from the user, because we didn't know we couldn't close it and open it again, but we could fix it.

In the second section we needed to make a few changes to have an easier implementation for the new functionalities. We also realised we couldn't keep the Scanner class to read the user's so we changed all the existing code of the reader to use instead a method with the buffer. To be able to save the different fields inside the new data aggregator we were implementing, we decided that the best solution was to duplicate the fields and save the copy.

We first created all the code for the last section of the assignment, and then we realised that using the proxy design pattern would help us create a more extensible and efficient implementation, so we changed it to do so.

Additional Question of part 3:

To avoid the need to provide in the constructor of field a converter for standard data types, we could create new subclasses for each of the different data types we are expecting to use. This would create a more modular and extensible design where each subclass could have its own conditions and treats, also making much simpler the process of ordering or managing the different fields depending on their data type.