***Potential improvements/ extensions***

# **1. Making the conversion process less specific to the given files**

The code that I have presented is quite specific to these files: after inspecting them, I came to notice that they all seem to follow a very similar, if not identical, pattern in how the data is laid out. I made use of that pattern when coding and making sure each relevant part of the file is checked, by looking for specific types of data, and handling them in a way that would be best suitable for these files alone. An extension to the code I have provided would be to generalise this process: have a way of converting several, completely different FHIR files using the same code.

# **2. Looking into how Docker could be used to improve the code**

For writing up the code, I have made use of standard development tools, such as PyCharm, XCode and IDLE. I am not entirely familiar with Docker, how it is used and the ways in which it could help improve the code and would need some time to look further into that. Hence, it can be noted as an extension to do more research on Docker, to better understand what it is used for and how and use Docker to create a container for my program.

# **3. Offer support for Cloud technologies**

Another extension would be the adaptability of the code to Cloud technologies, particularly to Cloud storage. This could also tie into the previous extension regarding Docker, and how this may be used for supporting cloud technologies. However, implementing such a feature would require access to a Cloud of some sort, where one can store the files there, and try accessing them through a Python script.

# **4. Improve the efficiency of the file conversion algorithm**

Although fully functional, the algorithm is quite inefficient due to the large number of for loops, which make it dependent on the size of the file, number of files, number of elements in a file or subpart of a file, etc. Looking into means of improving efficiency, by reducing the number of for loops, or better designing some of the helper functions, would help speed up the conversion process, and improve the algorithm’s runtime. An extension of the provided code would be to use different programming techniques to reduce the runtime of the algorithm (even by a constant factor, although it may not be significant, would still be considered an improvement).