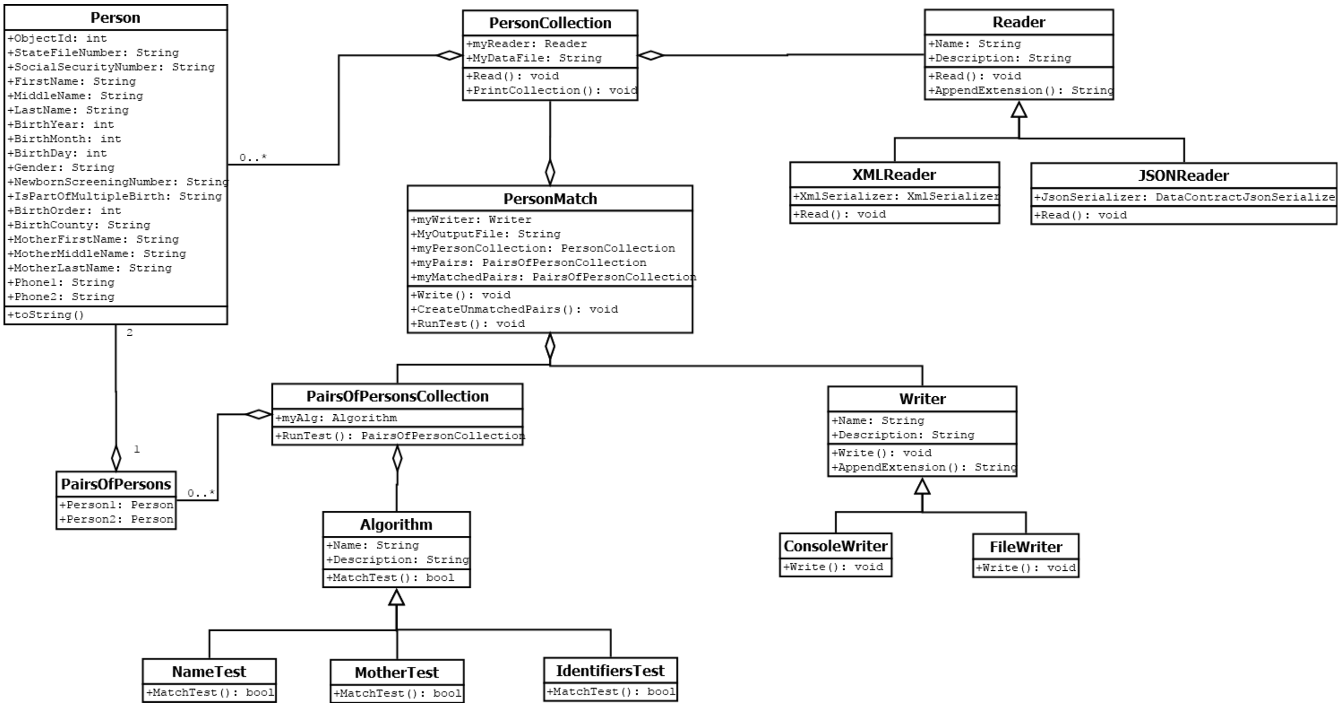
Matthew Morris 9/14/17

HW1-Report



Designing the Strategy Pattern into this homework assignment seemed fairly straightforward to me. Make classes that use certain behaviors like Writing or Reading, and make them extensible to input or output different file types. This was also implemented in the Algorithm class to have new algorithms easily added. Other classes, such as PersonMatch have these extensible classes built into them, while also any other unknown future class could use these Writer and Reader classes. I had an internal debate on whether to put Reader and Writer into the same class, but I figured it helped the modularity of the whole program to do it this way.

Implementing this was more challenging than creating the design. I realized I was putting many aspects of the program into the PersonCollection class, when really a PersonCollection shouldn’t do more than hold a list of Persons and get its information from the Reader. I had to add the PersonMatch class later on after I had already got my whole program working properly, to help make a more logical design.

Testing my algorithms did not give me too much trouble. I created 3 test cases for each algorithm that test simple to extreme values for each algorithm. I did find an error when I was comparing all null values that resulted in a matched person. I also created a menu type system in the console that I felt was a lot more intuitive to use than stating the input file, algorithm, and output file in a single line. Also, much of my code was modified from the example given by Professor Clyde. Let me know if that is a problem for future assignments. My input test files that were given are located in PersonMatcher/bin/Debug, and the output file location is in the root directory where I also have my UML.png diagram.