Armin Rezaiean-Asel – MP1 TASK 8

At times, different classes are so alike that they have similar properties and relatively commonly used pieces of code. A simple example of this is the Dude and Wizard case from class. Because the Wizard class was so similar to Dude, there was no point to re-write a majority of the code. Therefore, Wizard “extends” Dude and inherits its fields and methods, but there is still the possibility to add code to highlight its exclusive and unique new features.

As for implementation, when a class is implementing another class, the implementation refers to an interface. The interface contains only methods that the class implementing it could find useful. With respect to Basic2DMatrix extending the class AbstractBasicMatrix, it is useful because the Basic2DMatrix class has the purpose of creating, as it says, a basic Matrix in 2 dimensions. By extending AbstractBasicMatrix, the Basic2DMatrix class inherits the fields and methods from the former. After that, we're able to add specific code on top of what was already inherited so that Basic2DMatrix can do what AbstractBasicMatrix does (create a basic matrix) and then the additional functionality as well (of creating a basic matrix that is specifically 2D). This saves the programmer time and helps them avoid redundancy and unnecessary amounts of repeated code.

Other reasons of usefulness to use inheritance include the saving of memory space. As Basic2DMatrix extends the AbstractBasicMatrix class, the code that we would normally have to write in Basic2DMatrix is being inherited from AbstractBasicMatrix; therefore, we are saving memory. Basic2DMatrix being a separate class also allows for the possibility to extend other classes from it, thus permitting the possibility to save even more space, have less code, and do more in the program (through additional subclasses).

As for implementing the interface DenseMatrix, as I mentioned above, this implies that the implementer (Basic2DMatrix) inherits only the fields and methods from DenseMatrix, meaning Basic2DMatrix can be run with some of the c ode found in DenseMatrix. In this particular case, this implementation isn’t hugely useful. This is because in DenseMatrix, the only method is toArray. Furthermore, since implementation implies that Basic2DMatrix inherits the methods from DenseMatrix (which is only toArray in this case), there isn’t much use because Basic2DMatrix overrides the very definition of the toArray method. For this reason, toArray could simply be defined and declared in the Basic2DMatrix class itself.