ECM2414 – Production Code

Our production code was designed using the UML Class diagram that we created as a pair at the beginning of the design phase. This was built upon once we started coding as there were many finer details that were not thought about in the plan that had to be implemented at a later date.

With respect to the code produced, we created four different classes which were named *Card, CardGame, Player* and *Deck*.

Card

The *Card* class is very simple, however it is a key part of the overall implementation of the game; each player has four *Card* objects and equally so does each of the decks. A *Card* is largely a static object that merely holds an integer face value that can be called upon by the other classes.

Deck

Although the *Deck* class should be simple, this is where the main thread safety code is located as each deck can only be accessed by one *Player* at a time to avoid exceptions. It contains two main methods, these are *addCard* and *removeCard*. These methods are used by the *Player* class to discard and pick up new cards. Another key part of this class is the *ReentrantLock*. This is used to place a lock on the object by an instance of the *Player* class when it is calling either the *addCard* or *removeCard* methods. We decided to use this lock instead of synchronised methods because the implementation is more flexible and it automatically supports a fairness policy. A fairness policy means that it will try and make each object that attempts to access it be allowed access the same amount of times.