

# **BBM203 Software Laboratory I – Assignment I Report**

**Deniz Gönenç**

**Engineering Faculty**

**Department of Computer Engineering**

**21946146**

**Subject : Arrays**

**Deadline : 19.11.2020**

**Programming Language : C++**

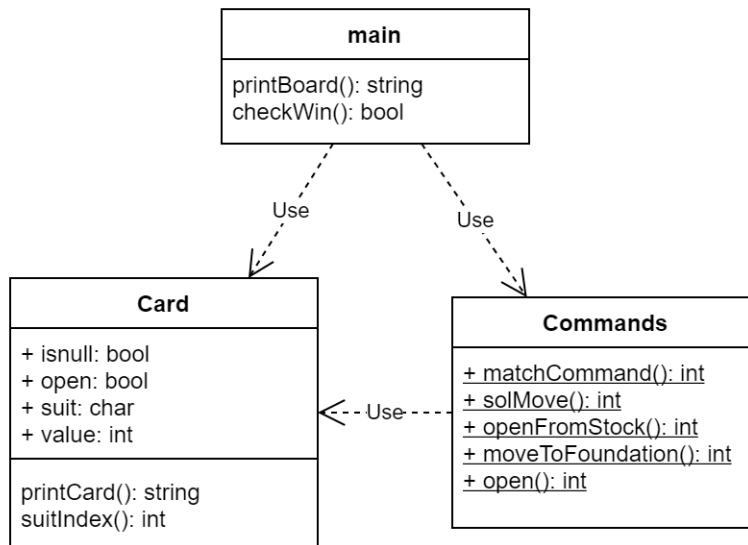
## Software Design Notes:

1. **Problem:** I needed to implement a klondike solitaire game in C++ with only OOP and arrays. Using vectors and the like was forbidden.

2. **Solution:**

2.1 – From reading the pdf I thought that creating a Card class was the most important part so I began with that. It held the cards value, its suit, if it is empty or not, if it is open or closed. That made it a lot easier to implement the required commands and writing to the file. I created my arrays around that class as well. It became the fundamental part of the program.

2.2 –



**main:** Has a function to print the board and a function to check the win conditions. File I/O happens here.

**Card:** Holds all the info about a card, has functions to print the card and return the index of a suit.

**Commands:** Holds all the commands to manipulate the arrays. All of the functions are static.

- 2.3 - I used five different arrays that contained Card objects, four of which the Commands manipulated. At the beginning of the program I used a Deck array to read Deck.txt. Then I created Piles, Stock, Waste and Foundation arrays in that order. Piles took the last 28 of the Deck array, and initialized them in an upper-triangular-matrix-like way. The stock took the remaining 24 cards in Deck. Waste and Foundation was empty. Also Piles and Foundation arrays were two-dimensional arrays. These four arrays represented the game board. I passed them as arguments to all the commands to execute them.

**NOT:** dos2unix \* ile çalıştırılması gerekli (ayrıca 4. sample I/O'da commands dosyasının adında ekstra bir boşluk var ve hataya sebep oluyor)