INTRODUCTION TO PROGRAMMING
DM550, DM857, DS801 (Fall 2019)
Exam project: Introduction

This year's project is centered on a card game. In the first phase, you will program an interface to play against the computer using classes that are provided in compiled form; in the later phases, you will design the lower-level classes and program an automatic player.

Before you start, you need to be well acquainted with the game. This document explains the rules and the scoring system. Be sure to understand them (and play a few games!) before starting to program.

Rules of the game

This game is played between two players. The goal is to score as many points as possible; ties are possible.

Preparation

The game is played with a deck of forty cards, divided in four suits (Spades, Hearts, Clubs and Diamonds) with ten possible face values (2, 3, 4, 5, 6, Queen, Jack, King, 7 and Ace, in ascending order). At the start of the game, each player is dealt three cards, which the other player cannot see. The remaining cards are put face-down in a pile; the lowest card of the pile is turned over and returned to the bottom of the pile: its suit is the *trump suit*.

Game play

In each round, each player places one card from their hand face-up on the table, starting with the player who won the previous round. (The players agree on who starts the first round.) The round's winner is decided as follows:

- if the second player played a card of higher face value in the same suit as the first player, then the second player wins the round;
- if the second player played a card in the trump suit and the first player did not, then the second player wins the round;
- otherwise, the first player wins the round.

The winning player collects both cards and adds them to his/her pile of collected cards. The players then take one card from the deck into their hand, in the same order as they played, so that they again have three cards in hand.

End game

When there are no cards left on the draw pile, the players continue playing as before, except that they no longer draw a new card at the end of the round. The game ends when there are no cards left in the players' hands.

Scoring

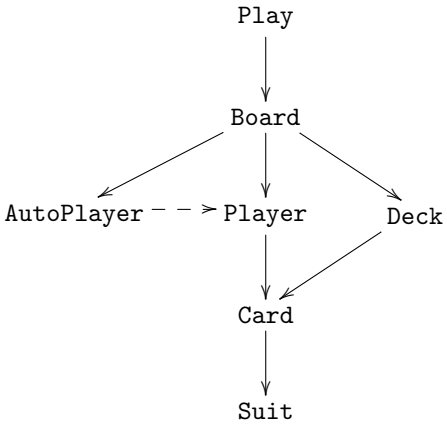
After the game ends, each player's score is determined by adding the total points in the cards they collected. Each Ace is worth 11 points, each 7 is worth 10 points, each King is worth 4 points, each Jack is worth 3 points, and each Queen is worth 2 points (see table). The remaining cards are worth no points, so there are 120 points in total. The player with the most points wins; if both players score 60 points, the game is a tie.

Card	Ace	Seven	King	Jack	Queen
Value	11	10	4	3	2

Table 1: Card values

The project

The class structure for the implementation is given below. The solid arrows represent client/provider relationships (with the arrow pointing from the client to the provider). The dashed arrow is a different kind of relationship that will be explained later in the course.



In the first phase of the project, you will develop the top-level class `Play`. All the remaining classes will be provided in compiled form, together with their documentation. In the second phase, you will develop the lower layer (classes `Suit`, `Card` and `Player`), following the contract previously defined. Finally, in the third phase you will develop the intermediate layer (classes `Deck`, `Board` and `AutoPlayer`), and try to make your automatic player as good as possible.

In each phase you will receive more detailed information about the methods that you need to implement, as well as some implementation tips and examples.