

## Assignment 07: UML, Design

Deadline: 7.5.2009, 8:15

Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

**Informatik:**  G1 (Prähofer)     G2 (Prähofer)     G3 (Würthinger)     G4 (Prähofer)

**WIN:**         G1 (Khalil)         G2 (Khalil)         G3 (Schwinger)

Task	Points	Paper submission	Electronic submission	corr.	Points
Assignment	24	Java source code, testing output	Java source code	<input type="checkbox"/>	
Bonus	4	<b>Please check the box, when you have implemented the Bonus</b>		<input type="checkbox"/>	

### Texas hold 'em Implementation (24 Points)

Implement the Texas hold 'em ([http://en.wikipedia.org/wiki/Texas\\_hold\\_'em](http://en.wikipedia.org/wiki/Texas_hold_'em)) Poker game based on the design worked out for assignment 6. You should start using the skeleton classes and methods of task 4 of the previous assignment. However, you need not strictly adhere to them and may change them as needed. You should use the classes given in assignment 6 (**Suit**, **Kind**, **Card**, and **Pokerhand**). The method **Pokerhand.compareHands** will help you determining the winning hand(s).

#### Extensibility

The implementation should be extensible (by adding classes that implement certain interfaces and not changing any of the existing game classes) in the following ways:

- New players (either human or computer players) can be added.
- New user interfaces can be used.

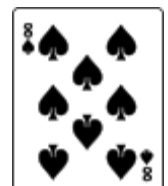
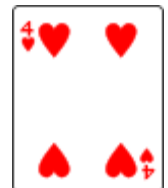
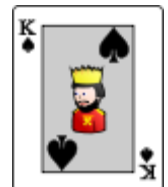
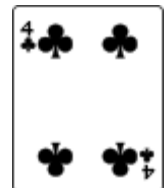
You should write at least one textual user interface as well as one implementation for a human player and a (simple) computer player.

#### Simplifications

In your implementation you may use the following simplifications:

- The maximum bet cannot be more than the money left of the poorest player (i.e. each player must be able to reach the maximum bet with an all-in move). This means that there cannot be a "side pot" ([http://de.wikipedia.org/wiki/Side\\_Pot](http://de.wikipedia.org/wiki/Side_Pot)).
- When programming the game logic you may assume that the players are cooperative and adhere to the rules (e.g. bet no more than they own and keep correctly track of their money).

There might be corner cases where you feel that other simplifications are appropriate. You may use those simplifications if you *justify and document* them.



### Bonus (4 Points)

Write several different computer player implementations with different (simple) strategies. Add another user interface implementation that does not display anything except a winner statistics. Test the computer player implementations by letting them play a large number of games and print the statistics.