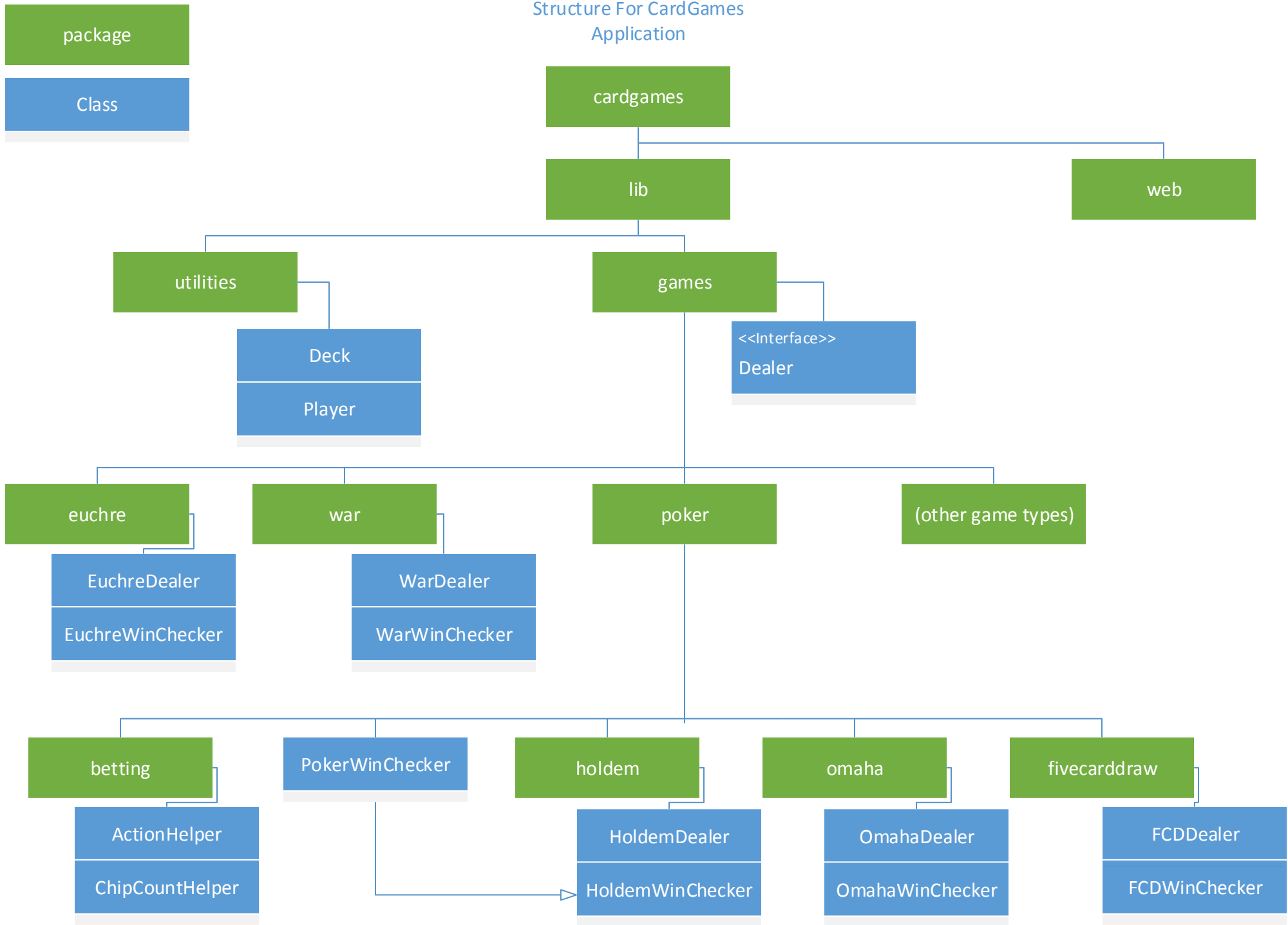


Structure For CardGames
Application



Notes

- The cardgames.lib package is the core Java code that controls the games themselves. We could also implement utility functions for connecting to databases, servers, etc somewhere in here eventually. The cardgames.web package is for the web application integrating with our games library.
- Classes in cardgames.lib.utilities should be used in many (or all) games. This includes the Deck class, which will provide general methods for creating a deck, shuffling the deck, dealing cards, and resetting a deck to full. This will also need to either allow for removal of cards from the standard 52 card deck or special cases for games like euchre that don't use a full deck. It also includes a Player class which could be used for storing player information for use by the Dealer classes.
- Each game has a Dealer and WinChecker class. Dealers control overall flow of the game such as dealing to players, calling methods for the phases of the game, and getting input/returning output. These should be created when a new instance of a game is created. As of now, these would be the objects to call from the web portion, pending changes. WinChecker classes evaluate game rules on a hand to hand basis, such as comparing card values.
- Each Dealer would create a Deck and WinChecker object in their implementations.
- Each Dealer class implements the Dealer interface (Each share common methods such as dealing to players and restarting for a new hand).
- Each poker WinChecker class inherits from PokerWinChecker, which can share common methods such as comparing five card hands and utility functions for values and suits.
- Each poker game can use the betting package to control most aspects of its betting, as a lot of this will remain the same in each type.