Inherits from
deck

Player
└ reserved card - Int - att
└ Card Array
└ ~~give get card~~ - use reserved card - Pop a card
       give card   from array and return
└ receive card - parameter of card   it
└ get contents - return all card values

Card Game

Ask for num players

Ask for pack location (same directory as Card Game)

Load Pack
Verify Pack
Store Pack

Make players and decks

top   Distribute cards from pack to players
to bottom
pops   Distribute cards from pack to decks — (log initial
pull it?   player hand)
                                        win
Check player hands — If for break loop

~~Pick card~~ players pick card from top of deck (log)
       players discard card to bottom of other deck (log)
       log current hand
~~Log deck contents~~ (log player and deck contents
              log a win + log other that a win occurred
              log exit
              log final hand
              log deck contents
              End

Card Game   ,num - Game loop

                          Card Deck
                             ↓
                          Player.

Card
~~Card number~~

Pack Handler
  └ pack — att — array os cards
  └ pack reader — met — loads file and creates pack
  └ validity check — type checking aswell — met
  └ get pacz — returns pack

Card   Pack reader
  └ card number — att   contains
  └ get num — getter sor   validity check
  └ constructor   card number
          └ parameters
                  └ card number

                              Queue — use library
                              ~~Stack~~ ↳

Card Deck ~~Stack~~ Queue
  └ card ~~Array~~ — att
  └ Num Cards — att
  └ ~~pop~~ dequeue — remove from queue and return that thg
  └ enqueue — add to queue — Parameter sorcard
  └ get contents — return all card values
WRITE FILE STUFF

Card Game Plan

1. Ask for number of players
2. Ask for pack file location
3. Load pack
4. Verify pack
5. Store pack
6. Instantiate players and decks
7. Distribute 1st half of pack to players (top to bottom)
8. Distribute 2nd half of pack to players (top to bottom)
9. Log initial player hands
10. ~ Log initial deck states
11. Start game loop
12. Check player hands
    - If a winning hand exists then break loop (step 19)
13. Players pick card from top of deck
14. Log Player pick ups
15. Players discard card to to bottom of other deck
16. Log Player discards
17. Log Player current hands
18. Loop back to step 11
19. Log player win to every player (Note: Winning player log ≠ losing player (log))

20. Log Player exits
21. Log Player final hands
22. Log final deck states