Project 2

Tittle
Three Card Game

Course
CSC-17A

Section
48096

Due Date
December 9, 2016

Author
Sili Guo

Table of Contents

I. Introduction

Three Card Game is a traditional poker gabbling game that is popular in China. Its rules is similar to many of western card games, but it is more flexible about the number of people playing this game. And the game between two people and several people can experience different feeling, and use different strategies during the game.

I like poker games because I enjoy the feeling of guessing others' cards by the action they do during the game; the process of holding small cards to try to bluff makes me feel excited. However, after playing many western poker games, such as Texas hold'em, and Black jack, I miss the day I played Three Card Game with my Chinese friends, for it makes me feel more about relax, rather than nervous.

The most interesting part for this game is there's always a way to win. The rules of the smallest cards set of 2, 3, and 5 can beat the largest cards set of full house! You will never know if you can win others or not before open both cards sets.

Note: In this game, 0 represent to 10.

II. Game Rules

A. Decide players number

The Three Card Game Program only limit 2~3 players to play this game. At the beginning, the program will ask player if he wants to join a 2 people game, or 3 people game

```
Would you like to join in a 2 people game or 3 people game?
Enter 2 or 3 for your choice:
```

B. Discard

After choose the number of players, the game started. Player can see the cards in his hand, and the bet of the table currently.

```
The cards in your hand are:
H9 SJ SQ

The current bet is: $0
```

Game starts by randomly pick the dealer. At the beginning of each round, the system will ask player if he would like to discard his cards. If the player was luckily picked as the dealer, then he will has one chance to leave the table without paying anything; otherwise, player can not take the bet on the table back.

C. Player's Action

If the player decide to play the game with his cards, the system will automatically display the choice menu of player's action: to see, raise bet, or challenge previous player.

```
Do you want to discard? (Y or N)
n

Choice Menu:
1) to see
2) raise bet
3) challenge
Please make your choice (1, 2, or 3)
```

Note: During the first round of the game, if the dealer did not discard, he can only choose to raise bet.

```
Do you want to discard? (Y or N)
N

How many bets do you want to raise? (1 bet = $10)
```

D. To See

To see is a simply strategy in card games to see how other people react, which may implicit if he has a good cards set or not.

By choosing to see, the player's bet will be equal to the current bet on the table, and the current bet will not change.

```
You choose to see.

The current bet is: $30
```

E. Raise Bet

If a player choose to raise bet, that means he wants to increase the amount of the bet on the table. This can be regarding as a signal of good cards set; but also can be a good strategy of bluff during the game.

To raise bet, there will always be a minimum and maximum amount for the bet to raise. For this program, player can raise at least 1 bet($10), but at most 10 bets($100) for each round; and the table can hold totally 120 bets ($1200);

```
You choose to raise bet.
How many bets do you want to raise? (1 bet = $10)
3

The current bet is: $70
```

F. Challenge Previous Player

The most interesting part of this game is to enjoy the process of challenging others and waiting to be challenged.

Challenge in Three Card Game only allow two players to compare(the compare rules will be discussed under this section); and there will be a judge to look at both players' card and report the winner, which means a player will not know others card even his challenge success.

When a player decide to challenge the previous player, he must has at least twice as much money as the bet of the table currently, because challenging others will cost twice as much as bet for their private bet. The winner takes all money, and loser exit game.

When a player is challenged, he can choose to discard and give the bet on the table to the challenger, or he can choose to bet the same amount of challenger to compare their cards.

```
You choose to challenge the previous player with double as bet on the table.
Player 2 discard. You win the game.
```

G. Compare Rules

There are seven ranks of cards set in Three Card Game:

- Full House (three same cards)

- Straight Flush
- Flush
- Straight
- One pair
- Single
- Special (2, 3, 5 in dif suit)

The rules are simple: full house > straight flush > flush > straight > one pair > single > special > full house.

Remind that special is the smallest card set in the game, but it can beat full house.

If two players have cards in same rank, then compare from the largest card to smallest one.

Since Three Card Game do not count suit, if two players have exactly the same cards, the challenger is the loser.

H. Result

If the player lose during the challenge, the game is over; otherwise, output how much the player won or lost.

```
You win $30.
```

III. Summary

Total Line of Code	1000+
Comment Line	
Variable	
Function	

Note: Some of the comment line are used for testing.

The total function based on the class Player. I stored a single card's information in Card class, and derive another class called Player, which stores a Card pointer, the rank of card set, the money and bet of player. To make sure the data is correct, I write it into a binary file, and read it when necessary. The program is constructed with decision makings. I used lots of loop and switch case to help program decide the branches that game will go. The AI makes decisions based on the random number creation. To pass variables between main and function, I use pointer point to function, pass by address, and pass pointer as variables. The variables are called by class in function that make sure the number wont't be mistakenly changed easily.

VI. Challenge During Program

A. Logic Errors

The two classes and pointers make me a little confused about the logic, and I made some mistakes about my pointer in class, and about the way I called get function, which cost me really long time to work on.

B. Transfer Program From Structure

Although structure is so similar as class, the way we use the variables declared in structure and class are so different, that one we can called just by pointer, one ask to call function to get the value.

I declared a pointer inside Player class, and return the point er to my main, however, it is hard to get the value inside the point er, I finally has to create 3 separate pointers to contain the pointer which makes me change a lot in my program.

V. Description

A. Sample Input/Output

First decide the players number

```
Would you like to join in a 2 people game or 3 people game?
Enter 2 or 3 for your choice: 2
```

The player is luckily chose as dealer. Ask if player if he would like to discard

```
Write to the file...
Read from the file...

The cards in your hand are:
D6 DQ CK

The current bet is: $0

Do you want to discard? (Y or N)
n
```

For first round, only ask to raise bet

```
How many bets do you want to raise? (1 bet = $10)
3
```

Then round for AI

```
The current bet is: $30

Player 2 choose to raise bet.
Player 2 raise two bets.
```

Player's 2nd round, ask if discard
If no, output choice menu

```
The current bet is: $50

Do you want to discard? (Y or N)
n

Choice Menu:
1) to see
2) raise bet
3) challenge
Please make your choice (1, 2, or 3)
1
```

Choose 1 means to see

```
You choose to see.
```

AI choose to challenge player

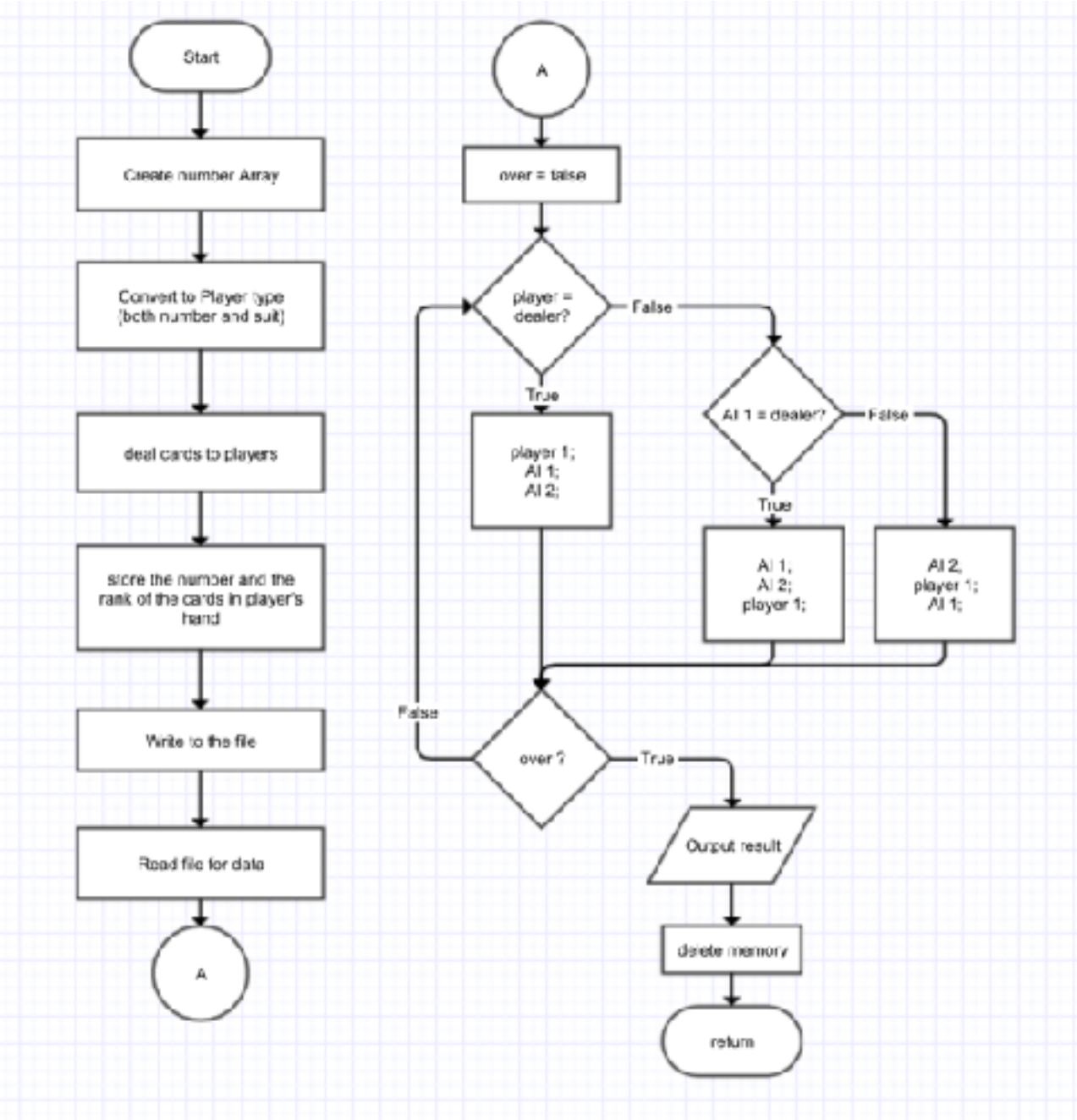
```
The current bet is: $30

Player 2 choose to challenge you with double bet of the table.

Would you like to accept challenge with double bet? (Y or N)
Remind if you refuse, you will lose game.
y
```

Accept challenge, and see the result

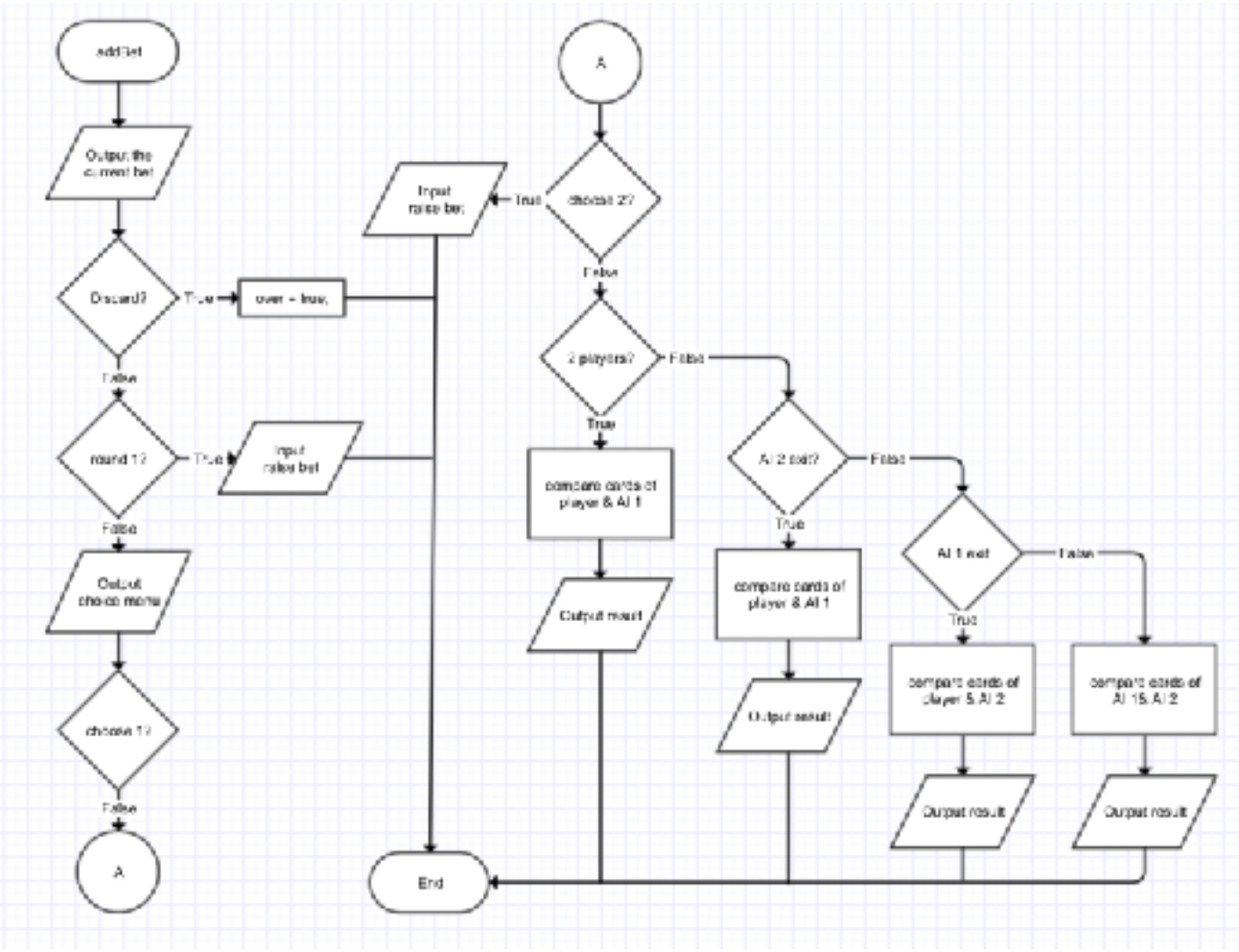
```
Sorry, you lose the game.
The cards in player 2's hand are:
H5 D8 DJ

You lose $60.
```

B. Flowcharts

Main:



addBet Function:



C. Pseudocode

- create a number array contains the number of 2~14
- convert the number array to Card array, containing suit and number
- promote the players number from player
- deal 3 cards to each player
- convert the 3 cards to number, and calculate the rank
- store all data about player into Player *players
- write the data into the file
- read the data from file
- randomly choose a player as dealer
- If player:
    - ask if discard
    - during first round, promote for the raised bet
    - not first round, output choice menu
    - if 1: to see —— do nothing, calculate money
    - if 2 raise bet —— promote raised bet, calculate money
    - if 3 challenge ——
        - if 2 players —— compare player & AI 1

- if 3 players ——
    - if AI 2 exit —— compare player & AI 1
    - else —— compare player & AI 2

- if AI:
    - AI I never discard
    - if 3 players —— AI 2 10% discard
    - if AI 2 not exit
    - if first round —— AI randomly raise bet
    - if not first —— randomly choose from 1~3 (1:2:2)
    - if choose 1 —— calculate money
    - if choose 2 —— randomly raise bet
    - if choose 3 ——
        - if AI 1:
            - if 2 players —— compare player & AI 1
            - if 3 players ——
                - if AI 2 exit, game over
                - else continue
        - If AI 2:

    - if AI 1 exit —— compare player & AI 2
    - else —— compare AI 1 & AI 2
- calculate the money of player
- output result (win or lose challenge, win or lose money)

D. Variables

Type	Variable Name	Description	Declare Location(line)
const int	Size	size of the number array	33
int	number[SIZE]	the number array of cards	34
	swap	represent the suit	34
	r	hold a random number	108
	choice	the number of players	126
	sortNum	temp hold number for sort	164
	start	random choose dealer	260
	max_one	max bet in one round	268
	max	max bet in the game	269
	round	the round	270
	bet	the bet on the table	271
	money	the money player holds after game	311
bool	mak	sign to stop sort loop	165
	over	game over	263
	two_exit	AI 1 exit game	264
	thr_exit	AI 2 exit game	265
Card	plyCard1	Contain class pointer for player 1	151
	plyCard2	Contain class pointer for player 2	269

	plyCard3	Contain class pointer for player 3	270
Player *	players	class variable	149

E. Concepts

Concept	Type	Code	Location(line)
Binary file	fstream	ios::out | ios::binary ios::in | ios::binary	894 904
Structure pointer	PlyNum	PlyNum *plyNum;	83
Dynamic structure array	PlyNum*	plyNum = new PlyNum[choice];	84
Class pointer	Card	Card *plyCard1;	151
	Player	Player *players;	149
Dynamic Class array	Player*	players = new Player[3];	
Constructor	Card	Card();	Card.h 24
	Player	Player();	Player.h 35
Destructor	Player	~Player();	Player.h 37
Operator Overloading	bool	operator>(const Player &);	Player.h 71
Static Member	static int	static int Tbet;	Player.h 26
Static Public Function	static void	static int getTbet()	Player.h 68

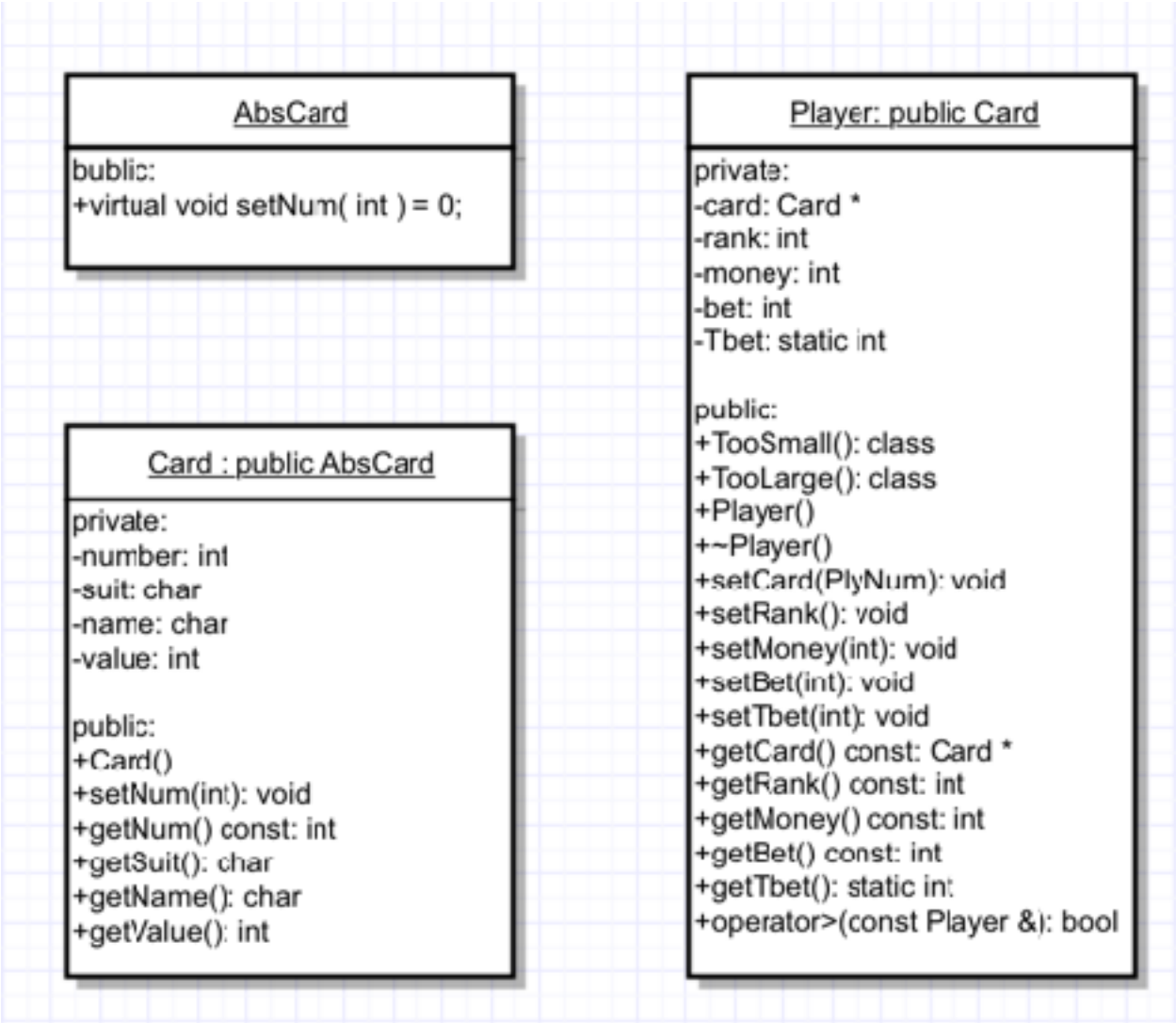
Aggregation	Player	Card *card	Player.h 22
Inheritance	Public	class Player : public Card	Player.h 20
Abstract Base Class	AbsCard	virtual void setNum(int) = 0;	AbsCard.h 13
Exception	class	TooSmall	Player.h 30
	Class	TooLarge	Player.h 33
	Player	Card * hold;	Player.h
Pointer in class	Card	Card *card;	Player.h 22
Return pointer in class	get function	Card *getCard() const { return card; }	Player.h 36
const function in class	Int	int getNum() const	Card.h 25

Bubble sort	int	sort = players[i].hold[j];
		players[i].hold[j] = players[i].hold[j + 1];
		players[i].hold[j + 1] = sort;
			162-183

Switch	switch	switch(start)	275
Loop	for	for (int i = 0; i < SIZE; i++)	79
	do-while	do {} while(mak)	165
Type casting	reinterpret_cast <type>	reinterpret_cast<char *> (p)	896

Random number	rand()	if (rand() % 5 < 3)	621

VI. UML


AbsCard
bublic:
+virtual void setNum(int) = 0;
Player: public Card
private:
-card: Card *
-rank: int
-money: int
-bet: int
-Tbet: static int
public:
+TooSmall(): class
+TooLarge(): class
+Player()
+~Player()
+setCard(PlyNum): void
+setRank(): void
+setMoney(int): void
+setBet(int): void
+setTbet(int): void
+getCard() const: Card *
+getRank() const: int
+getMoney() const: int
+getBet() const: int
+getTbet(): static int
+operator>(const Player &): bool
Card : public AbsCard
private:
-number: int
-suit: char
-name: char
-value: int
public:
+Card()
+setNum(int): void
+getNum() const: int
+getSuit(): char
+getName(): char
+getValue(): int