Pseudocode: CardGame
Elnaz_Dehkharghani_11015404

```
CLASS player that represents player methods and it's attributes
   INIT attrs for each player:
      name as string
      player hand list that hold cards
      is player currently in god mode
      is player currently in resurrect mode
      player has god spell?
      player has resurrect spell?
      points
      is player computer

   METHOD choose_characteristics:
      gets chosen card and PRINT its choose_characteristics
      IF player is computer RETURN random choose_characteristics
                        WHILE chosen characteristic is wrong ask user to chose the right one
      RETURN INPUT choose_characteristics

   METHOD give_card
      through this method we append cards to player hand

   METHOD won
      add player point by one

   METHOD dice_throw
      RETURN a random int (0,6)

   METHOD draw_card
      RETURN one card from card list at desire index

   METHOD ask_play_god
      IF player not used it's god spell:
         IF player is computer:
            randomly choose Whether to use spell or not if yes call use_spell method and add it's state instance
         IF player is not computer:
            PRINT would you like to play god spell?
            IF yes call use_spell method and save it's state

   METHOD use_spell
        gets spell we want to use and change instance state for using that spell

   METHOD ask_play_resurrect
      IF player is computer:
         randomly choose Whether to use spell or not if yes call use_spell method and add it's state instance
      IF player is not computer:
            PRINT would you like to play resurrect spell?
            IF yes call use_spell method and save it's state

END CLASS
```

```
CLASS deck
   INIT:
   deck name
   deck cards

   METHOD add_cards
   add cards to deck

   INNER METHOD __repr__ to show deck name

   METHOD shuffle
      to shuffle cards using random numbers

   METHOD distribute_cards(player1,player2):
      WHILE we have cards:
         call player1 Give Card method
         call player2 Give Card method

END CLASS

CLASS outdated_deck INHERITS FROM deck
   INIT:
   inherits all methods from deck CLASS

   METHOD add and shuffle
      add cards and shuffle it

   METHOD draw random card
      RETURN random card from this deck

END CLASS

CLASS character:
   INIT:
   name
   characteristics dictionary

END CLASS

CLASS play_card INHERITS FROM character
   INIT:
   inherits all from character

   METHOD repr:
      return card name

   METHOD compare
      gets opponent card and characteristic for compare



define cards as play_card instance
define oudated deck instance
define main deck
add cards to main deck
shuffle main deck
```

```
define instance for player1
INPUT would you like to play with computer?
IF yes
   define player 2 instance as computer with it's attributes
IF no:
   define player 2 as ordinary player

distribute cards from main deck method to player 1,player2

WHILE dice results are eual
   throw dice for each player

IF dice1_result > dice2_result:
   consider player 1 as starter
   consider player 2 as opponent
ELSE
   consider player 2 as starter
   consider player 2 as opponent

define counter for counting rounds

WHILE both players have at least 1 card in hand
   rounds+=1
   PRINT rounds
   starter.ask_play_resurrect()
   starter draw card and chose it's characteristics with it's corresponding method
   starter ask play god

   opponent ask play resurrect

   IF starter is currently in god spell mode AND opponent is in resurrect spell mode:
      check if starter is computer or not
      IF starter is computer randomly chose if opponent should play resurrect card or randomly chosen card
      IF starter is not computer INPUT would you like let opponent use resurrected card?
      opponent chosen card will draw at index or top of the hand based on conditions

   IF starter points more than other:
      starter won method will be called to add point
   ELSE:
      opponent won method will be called to add points
      and starter,opponent replace their place

   starter will reset current spell if it's using any
   opponent will reset current spell if it's using any

END WHILE

IF starter.points > opponent.points:
   PRINT starter.name wins
ELSE:
   PRINT opponent.name wins
```