

**author** = '8175858, Braun'

## Analysis:

The program is developed according to the EVA design pattern.  
Input and output are done in the console.

Output: Output is done in the console. Output is a float.

## Description of the program:

The program does not require any further libraries other than random, it needs the ue05\_modul.py and spielkarten.py module.

The program is written and tested in Python 3.9.6. A Python interpreter must be installed.

The program can be started with the command "python3 ue05.py" in the terminal.

The Output is done in the console.

No bugs are known.

The functions from ue05\_modul.py are used to play the game.

First, the user is asked to enter the players via the load\_players function. This function returns the player names in a list. Player count needs to be between 3 and 4.

Next the program initializes the game with the initialize\_variables function. This function creates a score dictionary and gives each player (key) a score of 0 (value). Then it makes a card list via the create\_card\_list function from spielkarten.py and shuffles it with the shuffle\_card\_list function from ue04.py and hands out equal amounts of cards to each player with the hand\_out\_cards function from ue04.py. It then returns the score and card list of each player. I have rewritten the hand\_out\_cards function, so it now returns a dict and have also deleted 'Pik' from the create\_card\_list function.

Then the game is played via the play\_game function. In it check\_card() checks if the card is playable bases on rules and returns a bool. play\_card() plays a viable card and removes it from the held\_cards list of that player. The player can go through all of his cards and select the one he wants to play. If he goes through all of his cards None is returned which later is catche via a condition in a loop, so he can select from the beginning again. get\_winner() determines the winner by counting the card values that are trump and returns the index of the player that won the current trick. update\_score() gives each winner of a trick a minus point for every 'Kreuz' he has won.

The game is played until one of the players held card list is empty. The playing order is randomized for everz trick. First a player gets to select the trump card for this trick. The current\_trick list keeps track of all of the cards played by

each player in a list of lists. The index of the current playing order is maintained in the `current_trick` list. In every trick the function goes through a players cards and the player chooses which one to play. Informations like trump card or currently held\_cards are also given. The loop breaks if a played card is not trump, because that's when a trick ends and scores get updated. If a player has no cards, the game is immediately over even without calculating the current trick. After every trick the `current_trick` list is cleared and the next trick starts. Also, the winner of the trick is determined and the score is updated. After the game is over the score is returned.

To determine the winner of the game, the `determine_winner` function is used. It returns the index of the player with the highest score, meaning the least amount of penalty points. The winner or winners are then printed out line by line in the console.