1.목표

1) card 클래스의 __init__

인자로 들어온 suit, rank를 멤버 변수 suit, rank에 저장

2) card 클래스의 get_value()

멤버 변수 suit, rank를 반환

3) get_card()

멤버변수 deck에서 하나를 뽑아서 반환

4)get_values()

card의 리스트를 인자로 받고 인자를 읽어가며 게임 점수를 계산

2.과제를 해결하는 방법

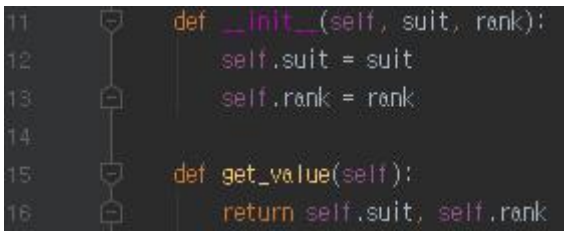
1) list에서 pop에 대한 이해

2) for 반복문, 튜플형 자료 구조, Mapping type 자료 구조 이해를 통한 get_values()문제 해결

3) 현재 인스턴스를 나타내는 self의 이해

3.과제를 해결한 방법

1) card 클래스의 __init__와 get_value



(1) 생성자 역할을 하는 __init__를 통해 클래스 자체 인스턴스에 suit, rank를 저장한다.

(2) get_value()를 통해 클래스 자체 인스턴스에 저장된 suit과 rank를 불러온다.

2) dealer 클래스의 get_card와 get_values

```
27      def get_card(self):
28          return self.deck.pop()
29
30      def get_values(self, cards):
31          sum = 0
32          ace = False
33          for index in cards:
34              rank = index.get_value()[1]
35              value = VALUE[rank]
36              if rank == 'Ace':
37                  ace = True
38              sum = sum + value
39          if ace and sum > 21:
40              sum = sum -10
41          return sum
```

(1) pop을 이용하여 리스트에 저장된 마지막 원소를 반환한다

(2) 점수를 반환할 변수 sum과 list내 ace카드 존재 유무를 판단할 변수 ace를 선언하고, 초기값을 각각 0과 False로 설정한다. for문을 통해 cards의 리스트를 순서대로 탐색하면서 점수를 저장한다. rank에 현재 index에 저장된 카드 패를 저장한다. get_value는 {카드 종류, 카드 패}와 같은 형태로 반환하기 때문에, ace 존재 유무와 점수 존재를 위해서는 두 번째에 저장된 자료가 필요하다. 따라서 get_value()[1]과 같은 형태로 두 번째 저장된 자료를 rank에 저장한다. 이후 Mapping type 자료 구조인 VALUE에서 해당 패에 해당하는 점수를 불러와 value에 저장한다. 만약 리스트에 Ace 카드가 있다면 ace에 True를 저장해서 존재 유무를 알 수 있게 한다. sum에는 기존 sum 값에 value를 더한 값을 저장한다. 결과를 반환하기 전 ace가 있는 상태에서 sum이 21를 넘으면 sum에서 10을 빼서 Ace가 1이 된 효과를 나게 한다. 최종적으로 점수에 따라 ace의 점수가 달라지는 get_values가 완성된다.

4.결과화면

1) card_tests 결과

```
C:\Users\HyunTaek\PycharmProjects\Blackjack\venv\Scripts\python.exe C:/Users/HyunTaek/PycharmProjects/Blackjack/card_tests.py
test___eq__ (__main__.CardTest) ... ok
test_create (__main__.CardTest) ... ok

----------------------------------------------------------------------
Ran 2 tests in 0.001s

OK

Process finished with exit code 0
```

2) dealer_tests 결과

```
C:₩Users₩HyunTaek₩PycharmProjects₩Blackjack₩venv₩Scripts₩python.exe C:/Users/HyunTaek/PycharmProjects/Blackjack/dealer_tests.py
test__create_deck (__main__.DealerTest) ... ok
test_get_card (__main__.DealerTest) ... ok
test_get_values_aa09 (__main__.DealerTest) ... ok
test_get_values_aj10 (__main__.DealerTest) ... ok
test_get_values_akqj (__main__.DealerTest) ... ok
test_get_values_k10 (__main__.DealerTest) ... ok
test_get_values_q08 (__main__.DealerTest) ... ok

----------------------------------------------------------------------
Ran 7 tests in 0.003s

OK

Process finished with exit code 0
```

3) ace가 11로 처리 됐을 경우의 main()

```
----+ 당신의 패 ----+
Clubs Ace
Spades 6
카드를 더 받을까요? (y/n): n
----+ 당신의 점수 ----+
총점: 17

Process finished with exit code 0
```

4) ace가 1로 처리 됐을 경우의 main(ace가 1로 처리)

```
----+ 당신의 패 ----+
Hearts 7
Hearts 4
Diamonds Ace
Clubs 2
카드를 더 받을까요? (y/n): n
----+ 당신의 점수 ----+
총점: 14

Process finished with exit code 0
```