

Projekt zaliczeniowy z języka Python

Autor: Piotr Piechowicz

Tytuł projektu: Gra w Blackjacka

Spis Treści

1. Opis zasad gry w Blackjacka
2. Zasady korzystania z programu
3. Opis implementacji

1. Zasady gry w Blackjacka

1.1 Cel Gry

Celem gracza w Blackjacku jest zebranie łącznej sumy punktów w swoich kartach, która jest **równa** lub jak **najbliższa od dołu wartości 21**.

Zebranie powyżej 21 punktów oznacza przegrane rozdanie.

Zebranie dokładnie 21 punktów (tzw. „oczko”), czyli dobranie **Blackjacka** jest od razu zwycięstwem gracza, chyba że krupier też dobierze 21 (remis).

1.2 Wartości kart

Punkty w Blackjacku są sumowane na podstawie wysokości otrzymanych przez gracza kart.

- Figury (walety, damy i króle) są warte **10 punktów**.
- Asy są warte **1** lub **11 punktów**, w zależności od decyzji gracza.
- Pozostałe karty (2-9) mają wartość odpowiadającą ich wysokości.

Do gry w Blackjacka używa się pełnej talii kart, jednocześnie korzystając z 6-8 talii lub tylko 1 talii (taki rodzaj Blackjacka jest zaimplementowany w tym projekcie)

1.3 Przebieg gry

Na wstępie krupier oraz gracz otrzymują po dwie karty. Gracz odkrywa obydwie karty, natomiast u krupiera widoczna jest tylko jedna karta. W tym momencie jeśli gracz nie uzyskał 21 punktów dokonuje wyboru:

- Gracz dobiera kolejną kartę (może to robić do czasu aż nie osiągnie lub przekroczy wyniku 21)
- Gracz pasuje

Po przekroczeniu 21 punktów gracz automatycznie przegrywa.

Po spasowaniu lub osiągnięciu 21 punktów krupier odkrywa drugą kartę i dopóki nie osiągnie wyniku większego lub równego 17 to dobiera kolejne karty

Następnie odbywa się porównanie wyników, wygrywa ten zawodnik który osiągnął 21 punkty lub był najbliżej tej liczby od dołu.

2. Zasady korzystania z programu

Po uruchomieniu programu użytkownik otrzymuje widok swoich kart oraz wyniku, a także podgląd jednej karty krupiera.

```
--- Witaj w kasynie! ---  
Twoja ręka:  
| Dwójka Kier |  
| Ósemka Pik |  
Wynik: 10  
  
Ręka krupiera:  
| Ukryta Karta |  
| Jopek Trefl |  
  
Wybierz co robisz [Dobieram / Pasuje]
```

Następnie może dokonać wyboru wpisując „dobieram” albo „pasuje” (lub w skrócie „d” albo „p”), dopóki nie osiągnie lub przekroczy 21 punktów lub

dopóki nie pasuje.

```
Wybierz co robisz [Dobieram / Pasuje] d
Twoja ręka:
| Dwójka Kier |
| Ósemka Pik |
| Dziewiątka Kier |
Wynik: 19
```

Następnie w terminalu przedstawiony jest proces myślowy krupiera, który dobiera lub pasuje, po czym następuje porównanie wyników.

```
--- Krupier odkrywa kartę ---
```

```
Ręka krupiera:
| As Pik |
| Jopek Trefl |
Wynik: 21
```

```
--- Końcowe wyniki ---
```

```
Twoja ręka:
| Dwójka Kier |
| Ósemka Pik |
| Dziewiątka Kier |
Wynik: 19
```

```
Ręka krupiera:
| As Pik |
| Jopek Trefl |
Wynik: 21
```

```
Krupier Wygrywa!
```

Po rozgrywce program pyta użytkownika czy chce zagrać kolejny raz, wpisanie „tak” lub „t” oznacza zagranie kolejnej partii, wpisanie „n” lub „n”

oznacza zakończenie programu.

```
Krupier Wygrywa!
-----
Czy chcesz zagrać ponownie? [T/N] t
Twoja ręka:
| Ósemka Kier      |
| Szóstka Kier     |
Wynik: 14

Ręka krupiera:
| Ukryta Karta     |
| Trójka Karo      |

Wybierz co robisz [Dobieram / Pasuje] |
```

```
Wygrywasz!
-----
Czy chcesz zagrać ponownie? [T/N] n
--- Kasyno dziękuje za wizytę! ---
```

3. Opis implementacji

Program składa się z 4 klas :

- Card
- Deck
- Competitor
- Game

Dodatkowo program korzysta z funkcji shuffle z modułu random

Klasa Card składa się z krotek zawierających nazwy i kolory kart, oraz słownika zawierającego nazwy oraz ich wartości, a także wartości oraz koloru konkretnej karty, zaimplementowana jest tam także metoda `__str__` , dzięki której wyświetlana karta jest wycentrowana.

Klasa Deck składa się z listy kart oraz metod shuffle oraz deal_card, które odpowiednio losowo tasują oraz zwracają kartę z góry talii (lub jeśli nie ma już kart to wywołują wyjątek NoCardsException)

Klasa Competitor składa się z listy kart, które posiada dany gracz, flagi określającej czy gracz jest krupierem oraz wartości punktowej jaką posiada gracz. Ponadto, klasa ta zawiera metody do dodania karty do ręki, odrzucenia kart z ręki, obliczenia wyniku, oraz wyświetlenia ręki i wyniku gracza.

Klasa Game składa się z talii kart utworzonej z klasy Deck oraz dwóch zawodników (gracza oraz krupiera) utworzonych z klasy Competitor(). Ponadto, klasa Game zawiera metody służące określeniu czy wynik gracza kończy grę, dodaniu karty graczowi (jeśli skończą się karty to tworzona jest nowa talia) lub krupierowi, oraz metody play(), w której zawarta jest główna logika gry.