

# Proiect BLACKJACK

---

## Scurta descriere

Implementare simpla a unui joc de **blackjack** conform cerintelor proiectului.

Jocul este exclusiv in consola.

## Clase

### IPlayer

Clasa ce contine metode comune jucatorilor si dealer-ului

- adauga o carte in mana curenta
- returneaza suma cartilor din mana curenta, tinand cont de as care poate avea 2 valori: 1 si, respectiv, 11

### Player

Jucator de blackjack. Mosteneste clasa **IPlayer**.

### Dealer

Mosteneste clasa **IPlayer** si reprezinta dealerul jocului de blackjack.

### Players

Lista de jucatori de blackjack care se afla la masa si pot incepe un joc.

- incarca datele jucatorilor din fisier
- creeaza o lista de jucatori (lista de obiecte **Player**)
- metode de a returna \ filtra jucatorii curenti
- jucatorii pot fi filtrati dupa stadiul in runda curenta: castigatori sau invinsi

### Card

Reprezinta o singura carte de joc

### Deck

Reprezinta un pachet standard de 52 de carti de joc.

- creeaza un pachet de carti cu toate cartile in ordine
- metoda de amestecare a pachetului
- metoda pentru extragerea unei carti

Amestecarea pachetului se face de un numar aleator de ori pentru a simula o amestecare cat mai realista; de asemenea timpul dintre amestecari este aleator.

modulul **random** este folosit pentru generarea de numere aleatoare

```
number_of_shuffles = random.randrange(10, 50)
log.debug("Shuffling the deck %d times" % number_of_shuffles)
for i in range(0, number_of_shuffles):
    delay(random.randrange(30, 70, 5))
    random.shuffle(self.card_deck)
```

## Game

Reprezinta jocul propriu-zis. Jocul se desfasoara in una sau mai multe runde.

Contine logica jocului si impreuna cu `game_outcome` reprezinta "grosul" aplicatiei.

Toata "actiunea" se intampla in metoda `run` care contine pasii rularii unei runde:

```
def run(self):
    """
    Game main
    """
    # set the bets
    self.__set_bets()
    # deal cards
    self.__deal_first_hand()
    # log current game state
    self.log()
    # ask each player - (h)it or (s)tand?
    self.__players()
    # dealer's turn
    self.__dealer()
    # game's outcome
    self.__outcome.get()
    # see who's broke
    self.__check_players_for_money()
    # show the round's results
    self.__show_outcome()
    # reset for a new game
    self.__reset()
```

Fiecare functie apelata din `run` are logica necesara implementarii cerintelor problemei:

- se accepta numai input numeric pentru pariuri
- suma pariata trebuie sa fie strict pozitiva si mai mica decat suma curenta din "buzunarul" fiecarui jucator
- jucatorii care nu mai au bani, sunt scosi din joc

## game\_outcome

Reprezinta rezultatul unui joc.

Deasemenea se fac ajustarile necesare in functie de castigul\pierderea fiecarui jucator si a dealer-ului

## Descriere Joc

La inceputul jocului se verifica jucatorii "inscrisi". Se intreaba daca se doreste inceperea unei runde, iar daca raspunsul este afirmativ, runda va incepe.

### Desfasurare runda

Runda se desfasoara intr-o maniera liniara:

1. Jucatorii de la masa pun pariurile

2. Se impart cartile

- fiecare jucator in ordine primeste prima carte
- dealer-ul primeste o carte
- fiecare jucator in ordine primeste a doua carte

3. Randul jucatorilor:

- fiecare jucator este intrebat pe rand daca mai doreste o carte sau se opreste
- daca jucatorul se opreste se trece la urmatorul jucator
- daca jucatorul trece de 21 se trece automat la urmatorul jucator
- daca nu mai sunt jucatori se trece la dealer

4. Randul dealer-ului:

- daca toti jucatorii au trecut de 21 dealerul castiga automat si nu mai cere o carte
- dealer-ul extrage cate o carte din pachet pana depaseste 17 (dar nu 21)
- daca dealerul a depasit 21 runda se opreste

5. Se verifica jucatorii care au ramas fara bani

- acestia sunt scosi automat din joc

6. Se afiseaza rezultatul rundeii curente

7. Se intreaba daca se mai doreste inceperea unei runde noi

- raspuns afirmativ

7.1. Se verifica daca mai sunt "Destule" carti in joc (destule insemnand complet aleator ca avem cel putin 5 carti pentru fiecare persoana de la masa - jucatorii si dealer-ul).

7.1.2. Daca nu sunt destule carti se va folosi un pachet nou si amestecat

- raspuns negativ -> jocul se termina si aplicatia se inchide

### Reguli Joc

1. As-ul valoreaza 1 sau 11

2. Valoarea As-ului poate fi schimbata oricand din 1 in 11 sau invers (varianta cea mai buna va fi aleasa)

3. Daca toti jucatorii au peste 21 la final, dealer-ul castiga automat

4. Daca dealerul pierde:
  - 4.1. Jucatorii care au peste 21 vor pierde
  - 4.2. Jucatorii care au sub 21 vor castiga
5. Fiecare jucator castigator va castiga doar valoarea pariata de el (isi va dubla suma pariata)
6. Jocul se joaca jucator vs. dealer a.i. un jucator poate castiga iar unul poate pierde, rezultand ca dealerul a castigat intr-un "meci" si a pierdut in celalalt.