Tartalomjegyzék

Feladatok

Overload

1. Írjunk egy függvényt, aminek az els? argumentuma **n**, egy **int** típusú változó. A függvény térjen vissza **True**-val, ha annyi extra paraméterrel hívták meg, mint az els? bemeneti paraméter értéke, egyébként térjen vissza **False**-szal.
2. Definiáljunk egy **szumma** függvényt, ami tetsz?legesen sok bemeneti paraméterének összegével tér vissza!
    1. Kezeljük le a kivételt, ha a paraméterek típusa nem azonos!
3. Definiáljunk egy **print_words** függvényt, úgy, hogy a megadott (akármennyi) szavakat annyiszor írja ki, amennyit megadunk bemenetnként (szavanként)!
    1. Kezejük le kivételként, ha a bemeneten nem egész számot adtak meg a szó gyakoriságára!

Sakk

Definiáljuk a **Piece** osztályt. Ez reprezentál egy sakkbábut, tároljuk a pozícióját a táblán két koodinátával, a színét (black/white), illetve a **__repr__** írja ki, hogy hol áll (A2, G3 etc.)!

- Definiáljuk a bábu leszármazottjaként a **King** és a **Pawn** osztályokat!
- Legyen a leszármazottaknak (King, Pawn) is **__repr__** függvénye úgy, hogy az már a figura típusát is kiírja. (Nem muszáj a leszármazott osztályban megvalósítani, lehet az ?sosztályban is)
    - Érdemes megnézni a sakk figurák unicode kódját: [1]
- Minden leszármazottnak legyen egy **move(pos)** metódusa, ahol a **pos** egy sztring (A3, G2 etc.)! Mozgassuk el a bábut, ha szabályos a lépés! Ha a lépés szabályos volt, térjünk vissza **True**-val, egyébként **False**-szal!
- Definiáljuk a **PieceMoveError** osztályt. Ha szabálytalan a lépés, dobjunk egy ilyen exceptiont és kezeljük le!

Legyen most egy **Board** osztályunk! Két listát tároljunk: **white** és **black**, a játékosok bábuival!

- Legyen a **Board** osztálynak egy **move(player, pos1, pos2)** metódusa, ami a **pos1** pozícióban álló bábut a **pos2** helyre mozgatja, ha a lépés szabályos! Ehhez definiáljuk át a **King** és a **Pawn** osztályok **move** metódusát, hogy paraméterül kapjon egy **board** változót, ami a tábla!
    - A normál szabályok mellett vegyük figyelembe, hogy áll-e ott más bábu! Ha az a bábu sajátunk, a lépés szabálytalan, ha az ellenfél bábuja, akkor távolítsuk el a pályáról, hiszen leütöttük!
    - Írjuk meg a **Knight, Rook, Bishop** és **Queen** osztályokat!
        - Kezdjük a **Rook**-kal, ez a legegyszer?bb! Ügyeljünk rá, hogy a ne lépjünk át másik bábun, hiszen ez szabálytalan!
        - Másodikként a **Knight** osztályt írjuk meg! A ló ugorhat, tehát a szabályok írásánál ezt nem kell figyelembe venni.
        - A **Bishop** után a **Queen** lépését könny? összerakni.

        - ♦ Oldjuk meg, hogy a **Pawn** ne állhasson az ellenfél alapvonalán!
    - A **Board** osztályon belül definiáljunk egy **check** metódust. Térjen vissza azzal a színnel, amelyik király sakkban áll!
        - ♦ Definiáljuk át a lépést, hogy az csak akkor legyen szabályos, ha a lép? játékos királya nem áll sakkban a lépés után!
    - Legyen **__repr__** függvénye a **Board**-nak!

Miután mindez megvan, közel állunk ahhoz, hogy tudjunk sakkozni. Definiáljuk a **start** metódust, ami kezd?állípotba teszi a táblát.

Ínyenceknek

- Módosítsuk a gyalogot az *en passant* szabály értelmében.
- Értelmezzük az algebrai notációt (Bf5, Qc3, Ne2, Kcd4, Kxd5 stb.) és lépjünk eszerint!
- Legyen lehet?ségünk sáncolni, ha még nem tettük meg (0-0 és 0-0-0 az algebrai jel)!
- Soroljuk fel egy játékos összes lehetséges szabályos lépését!
- Definiáljuk a mattot!