

Borospince (40 pont)

Egy borospincében szekrények vannak, amelyek polcain borokat tárolnak. Készítsd el a BorospinceException, Bor és Szekreny osztályokat a feladatleírás alapján!

A BorospinceException osztály (2 pont)

Hozz létre egy BorospinceException nevű kivételosztályt, amely az Exception beépített kivételosztályból öröklődik! Készíts egy egyparáméteres konstruktort az osztályon belül, amely egy hibaüzenetet vár paraméterben! A konstruktorban hívd meg az ősoosztály konstruktorát és add át neki paraméterben a hibaüzenetet!

A Bor osztály (13 pont)

1. Hozz létre egy Bor osztályt, amely a fajta, evjarat és _alkoholtartalom adattagokkal rendelkezik!
2. Az osztály konstruktora a bor fajtáját, évjáratát és alkoholtartalmát várja paraméterben (ebben a sorrendben)!
3. Az alkoholtartalom paraméter értékét ne legyen kötelező megadni, alapértéke legyen 12.5!
4. Inicializáld a fajta, evjarat és _alkoholtartalom adattagokat a paraméterek alapján! (3 pont)
5. Írj gettert és settert a fajta adattaghoz, fajta néven! A getter adja vissza az adattag értékét, a setter pedig állítsa be azt a paraméterben kapott értékre! (2 pont)
6. Írj gettert és settert az evjarat adattaghoz, evjarat néven! (1 pont)
7. Írj gettert és settert az _alkoholtartalom adattaghoz is, alkoholtartalom néven!
 - A setterben kezeld le, hogy csak 0 és 100 közötti valós szám lehessen az adattag értéke!
 - Nem megfelelő típusú vagy értékű paraméter esetén dobj egy BorospinceException típusú kivételt, amit a Nem megfelelo alkoholtartalom! szöveggel inicializálj! (2 pont)
8. Írd át az osztály konstruktorát úgy, hogy az alkoholtartalom értéke itt is ellenőrizve legyen! (1 pont)
9. Definiáld felül az osztályban az objektum szöveggé alakításáért felelő metódust úgy, hogy az a következő szöveggel térjen vissza: {fajta} (evjarat: {evjarat}), melynek alkoholtartalma: {_alkoholtartalom}% (a kapcsos zárójelek helyére értelemszerűen a megfelelő értékek legyenek behelyettesítve)! (2 pont)
10. Definiáld felül az osztályban az == operátor működését!

- Ha az adott objektumot egy másik Bor objektummal szeretnénk összehasonlítani, akkor az operátor adja vissza, hogy a két bor fajtája, évjáráta és alkoholtartalma rendre megegyezik-e! A fajták egyezésének vizsgálatakor a kis- és nagybetűket ne különböztess meg (tehát pl. aszu és Aszu ugyanaz a fajtanév)!
- Amennyiben a függvény nem Bor típusú paramétert kap, akkor logikai hamis értékkel térjen vissza! (2 pont)

A Szekreny osztály (25 pont)

1. Hozz létre egy Szekreny osztályt, amelynek egyetlen adattagja egy borok nevű lista! Ebben a listában tároljuk a szekrény polcain lévő borokat.
2. Az osztály konstruktora nem vár paramétert, és a borok adattagot egy üres listával inicializálja! (2 pont)
3. Írj egy get_bor metódust, amely paraméterül egy n egész számot kap! A metódus térjen vissza a borok lista n-edik indexű elemével!
 - Az egyszerűség kedvéért csak a nemnegatív indexeket kezeljük, így ha a paraméterben érkező n-érték negatív vagy nagyobb, mint a lista utolsó elemének indexe, akkor dobj egy *BorSpinceException* típusú kivételt, amit a Nem letezo index! szöveggel inicializálj! (3 pont)
4. Definiáld felül a += operátort az osztályban úgy, hogy ha az egy Bor objektumot kap paraméterül, akkor szűrje be azt a borok lista végére! Egyéb típusú paraméter esetén dobj egy *TypeError* típusú kivételt, amit a Nem bor! szöveggel inicializálj! (3 pont)
5. Definiáld felül a + operátort az osztályban, amely egy másik Szekreny objektumot várjon paraméterben! Az operátor térjen vissza egy új Szekreny objektummal, amelynek a borok listája az aktuális és a paraméterben érkező objektum borok listáinak összefűzése legyen! Nem megfelelő típusú paraméter esetén dobj egy *TypeError* típusú kivételt, amit a Nem szekreny! szöveggel inicializálj! (3 pont)
6. Írj egy atlag_alkoholtartalom metódust, amely térjen vissza a szekrényen lévő borok (borok adattag) alkoholtartalmának átlagával! Amennyiben nincs egyetlen bor sem a szekrényen, akkor dobj egy *BorSpinceException* típusú kivételt, amit az Ures a szekreny! szöveggel inicializálj! (3 pont)
7. Készíts egy paraméter nélküli statisztika metódust! A metódus számolja össze, hogy a különféle borfajtákból mennyi található a szekrényen, és az eredményt adja vissza egy dictionary-ben! (5 pont)
 - Ha a szekrényen nincs egyetlen bor sem, akkor a visszatérési érték egy üres dictionary.
 - Ha a szekrényen van bor, akkor számold össze az egyes borfajták darabszámát! A borfajták nevében ne különböztess

meg a kis- és nagybetűket (tehát pl. aszu és Aszu ugyanaz a fajtanév)!

Példa a metódus működésére:

A szekrény tartalma:
Bor('tokaji aszu', 2017, 13.5),
Bor('egri bikaver', 2013, 12.0),
Bor('TOKAJI ASZU', 2015, 13.8)

A statisztika metódus visszatérési értéke:
{'tokaji aszu': 2, 'egri bikaver': 1}

8. Írj egy megisszak metódust, amely egy Bor típusú objektumot vár paraméterben! A metódus törölje az adott bort a borok listából, amennyiben az szerepel a listában! Feltehetjük, hogy minden bor objektum egyszer fordul elő a szekrényen.
 - Ha a bor nem szerepel a listában, akkor dobj egy *BorospinceException* típusú kivételt, amit a Bor nem található! szöveggel inicializálj!
 - Ha a metódus nem Bor típusú paramétert kap, akkor dobj egy *TypeError* típusú kivételt, amit a Nem bor! szöveggel inicializálj! (3 pont)
9. Definiáld felül az objektum szöveggé alakításáért felelő metódust az osztályban!
 - Amennyiben a borok lista üres, akkor a metódus térjen vissza az Ez egy üres szekrény. szöveggel!
 - Ellenkező esetben a metódus térjen vissza egy olyan szöveggel, amely 1 vesszővel és 1 szóközzel elválasztva tartalmazza a szekrényen lévő borfajták nevét és az azokból meglévő mennyiséget, {darab} {fajta} formátumban! A szöveg végén ne legyen se vessző, se szóköz! (3 pont)

Tip

Használd fel a statisztika metódus visszatérési értékét!

Példa a metódus működésére:

A szekrény tartalma:
Bor('tokaji aszu', 2017, 13.5),
Bor('egri bikaver', 2013, 12.0),
Bor('TOKAJI ASZU', 2015, 13.8),
Bor('chardonnay', 2019, 13.0)

Az `__str__` metódus visszatérési értéke:
'2 tokaji aszu, 1 egri bikaver, 1 chardonnay'

Példa a kód működésére

```
bor1 = Bor('Tokaji aszu', 2017, 13.5)
bor2 = Bor('Gyanus kinezetu kannasbor', 2010)
bor3 = Bor('ToKaji AsZu', 2015, 13.8)
bor4 = Bor('Chardonnay', 2019, 13.0)

bor2.fajta = 'Egri bikaver'
bor2.evjarat = 2013
bor2.alkoholtartalom = 12.0
print(f'{bor2.fajta}, {bor2.evjarat}, {bor2.alkoholtartalom}') # 'Egri bikaver, 2013, 12.0'

print(bor1)          # 'Tokaji aszu (evjarat: 2017), melynek alkoholtartalma: 13.5%'
print(bor1 == Bor('TOKAJI ASZU', 2017, 13.5)) # True
print(bor1 == bor2)   # False
print(bor1 == 'Hibas tipusu parameter!')     # False

szekreny1 = Szekreny()
szekreny2 = Szekreny()

szekreny1 += bor1
szekreny1 += bor2
szekreny1 += bor3
szekreny2 += bor4

szekreny3 = szekreny1 + szekreny2

print(szekreny3.get_bor(0))      # 'Tokaji aszu (evjarat: 2017), melynek alkoholtartalma: 13.5%'
print(szekreny3.get_bor(3))      # 'Chardonnay (evjarat: 2019), melynek alkoholtartalma: 13.0%'
print(szekreny3.atlag_alkoholtartalom()) # 13.075

szekreny2.megisszak(bor4)
print(szekreny2.statisztika())    # {}
print(szekreny3.statisztika())    # {'tokaji aszu': 2, 'egri bikaver': 1, 'chardonnay': 1}

print(szekreny2)                  # 'Ez egy ures szekreny.'
print(szekreny3)                  # '2 tokaji aszu, 1 egri bikaver, 1 chardonnay'
```