

Tartalomjegyzék

- 1 7. gyakorlat - Osztályok definiálása
 - ◆ 1.1 Feladatok
 - ◆ 1.2 Feladatok megoldása
 - ◆ 1.3 Házi feladatok megoldása
 - ◇ 1.3.1 ember osztaly
 - ◇ 1.3.2 kinyer metodus

7. gyakorlat - Osztályok definiálása

Ezen a gyakorlaton az el?z? heti el?adás anyagát dolgozzuk fel.

Feladatok

Feladatok a CloudCoder-en megtalálhatóak. Ajánlott sorrend foglalas_1-től foglalas_6-ig számsorrendben.

Az összes feladat egymásra épül, ugyanazt a **Foglalas** nevű osztályt fejlesztitek egyre tovább. Ezért amikor kész az egyik feladat, érdemes átmásolni az addigi megoldásokat a következő feladathoz. De vigyázzatok, az ellen?rz? kódot ne másoljátok át, csak a megoldásokat!

A mostani feladatok mind tartalmaznak a kódvázlatban ellen?rz? kódot. Ehhez ne nyúljatok hozzá (a házi feladtnál pl. ha hozzányúltok, az hibásnak számít). Az se fontos, hogy megértsetek ami oda van írva, nem kell elolvasnotok. Ha véletlenül átírnátok, itt az eredeti kódvázlat mindegyik feladathoz:

foglalas_1:

```
class Foglalas(object):
    pass

#####
# Ellenorzo kod, ne modositsuk!

def foglalas_1(a, b):
    o = Foglalas(a, b)
    return o.__dict__
```

foglalas_2:

```
class Foglalas(object):
    pass

#####
# Ellenorzo kod, ne modositsuk!

def foglalas_2(a, b):
    return a.nem_foglalt(b)
```

foglalas_3:

```
class Foglalas(object):
    pass

def szabad(foglalasok, osszes_ules):
    pass

#####
# Ellenorzo kod, ne modosituk!

def foglalas_3(a, b):
    return szabad(a, b)
```

foglalas_4:

```
class Foglalas(object):
    pass

#####
# Ellenorzo kod, ne modosituk!

def foglalas_4(a, b):
    assert a.valtoztatas(b) == None
    return a.__dict__
```

foglalas_5:

```
class Foglalas(object):
    pass

def foglal(foglalasok, osszes_ules, nev, darab):
    pass

#####
# Ellenorzo kod, ne modosituk!

def foglalas_5(a, b, c, d):
    ret = foglal(a, b, c, d)
    if ret:
        return ret.__dict__
    return ret
```

foglalas_6:

```
class Foglalas(object):
    pass

def foglal(foglalasok, osszes_ules, nev, darab):
    pass

#####
# Ellenorzo kod, ne modosituk!

def foglalas_6(a, b, c, d):
    try:
        ret = foglal(a, b, c, d)
        return ret.__dict__
    except:
        return "Ellenor: hiba " + sys.exc_info()[0].__name__ + " " + str(sys.exc_info()[1].args)
```

Feladatok megoldása

Itt az összes feladat együttes megoldása:

```
class Foglalas(object):
    def __init__(self, nev, helyek):
        self.nev = nev
        self.helyek = helyek

    def nem_foglalt(self, lehetosegek):
        return [lehetoseg for lehetoseg in lehetosegek if lehetoseg not in self.helyek]

    def valtoztatas(self, uj_helyek):
        self.helyek = uj_helyek

def szabad(foglalasok, osszes_ules):
    ulesek = osszes_ules
    for foglalas in foglalasok:
        ulesek = foglalas.nem_foglalt(ulesek)

    return ulesek

def foglal(foglalasok, osszes_ules, nev, darab):
    szabadok = szabad(foglalasok, osszes_ules)
    helyek = []
    for ules in osszes_ules:
        if ules in szabadok:
            helyek.append(ules)
            if len(helyek) >= darab:
                return Foglalas(nev, helyek)
        else:
            helyek = []
    raise RuntimeError("Nincs eleg ules!")
```

Házi feladatok megoldása

ember_osztaly

```
class Ember(object):
    def __init__(self, nev):
        self.nev = nev
        self.baratok = []
```

kinyer_metodus

Az osztály végső kódja:

```
class Meccs(object):
    def __init__(self, A_nev, B_nev):
        self.A_nev = A_nev
        self.B_nev = B_nev
        self.A_pont = 0
        self.B_pont = 0

    def pont(self, nev, szam):
        if self.A_nev == nev:
            self.A_pont += szam
        elif self.B_nev == nev:
            self.B_pont += szam
        else:
```

```
        raise ValueError("Ismeretlen csapatnev!")

def kinyer(self, ponthatar):
    if self.A_pont >= ponthatar:
        return self.A_nev
    elif self.B_pont >= ponthatar:
        return self.B_nev
    else:
        return "senki"
```