

Vizsga kérdések

**Beadási határidő:
2030.20.30. 20:30:00**

I. OOP Kérdések

1. Mi a célja a SOLID elvből az I? Fogalmazd meg 1-2 mondatban!

2. Melyik SOLID elv sérül az alábbi példában?

```
class Animal:
    def make_sound(self):
        print("Some sound")

class Dog(Animal):
    def make_sound(self):
        print("Bark")

class RobotDog(Dog):
    def make_sound(self):
        print("Beep")
```

3. Mi a különbség az osztályattribútumok és példányattribútumok között?

II. Elméleti kérdések

- Melyik adattípus a (1, 2, 3) kifejezés?
- Melyik függvénnyel tudsz egy listát hossz szerint lekérdezni?
- Mi a különbség a == és = operátor között?
- Mi történik a break utasítás használatakor egy ciklusban?
- Mi a neve a kivételkezelés fő kulcsszavának?
- Hogyan írnál egy reguláris kifejezést, ami számot keres?
- Milyen kulcsszóval hozol létre globális változót egy függvényben?
- Melyik comprehension típus jön létre {x: x**2 for x in range(3)} használatával?
- Hogyan lehet szöveges formázást végrehajtani f-string használatával?
- Milyen típusú objektum a Python set adattípusa?

III. Gyakorlati kérdések

- **Mi lesz a kimenet?**

```
x = 5
y = 3
print(x ** y)
```

- **Mi történik a kódban, és mi a kimenet?**

```
a = 10
b = 3
print(a // b)
```

- **Milyen kivételt dob a kód?**

```
my_list = [1, 2, 3]
print(my_list[5])
```

- **Mi lesz a `my_set` értéke a kód futtatása után?**

```
my_set = {1, 2, 3}
my_set.add(4)
my_set.add(2)
```

- **Egészítsd ki a kódot úgy, hogy a fájlt tartalommal nyissa meg és írjon bele!**

```
with open("example.txt", "___") as file:
    file.write("Hello, World!")
```

- **Mi a kimenet?**

```
import math
print(math.sqrt(16))
```

IV. Kötelezően választható feladatok: Válaszd ki tetszőlegesen: A és B feladat közül az egyiket és oldd is meg!

A. JSON fájl feldolgozás

Írj egy Python függvényt, amely a `json` modul segítségével beolvas egy JSON fájlt, amelyben felhasználók adatai találhatók (név, életkor, email).

A függvény szűrje ki azokat a felhasználókat, akik 18 évnél idősebbek és érvényes e-mail címmel rendelkeznek (regex-szel ellenőrizve).

A függvény térjen vissza a szűrt felhasználók listájával.

Példa bemenet:

```
[
  { "name": "Anna", "age": 25, "email": "anna@example.com" },
  { "name": "Béla", "age": 16, "email": "bela@example.com" },
  { "name": "Cecil", "age": 30, "email": "cecilatexample.com" }
]
```

Példa futás:

```
users = filter_valid_users("users.json")
print(users)
# [{'name': 'Anna', 'age': 25, 'email': 'anna@example.com'}]
```

B. API hívás tesztelése (mockolással)

Írj egy Python unittest osztályt, amely teszteli a következő függvényt, ami egy API-ról lekérdezi az időjárást.

Mivel valós API-hívást nem akarunk futtatni a tesztekben, **mockolással** (pl. `unittest.mock`) szimuláld az API válaszát.

A függvény neve `get_weather(city: str)`, és egy szótárat ad vissza a hőmérséklettel.

Példa futás:

```
print(get_weather("Budapest"))
# {'city': 'Budapest', 'temp': 22}
```

Elvárt tesztek:

- Ellenőrizze, hogy a visszaadott városnév helyesen szerepel.
- Ellenőrizze, hogy a hőmérséklet szám típusú.
- Tesztelje, hogy hibás város esetén kivételt dob.