Prooktatás Python modul vizsga

Tari Enikő

2025.09.20.

I. OOP, SOLID, KISS, DRY, YAGNI Kérdések

1. Mi a DRY (Don't Repeat Yourself) elv lényege? Adj példát!

   Ne írj le többször olyan kódot, ami ugyanazt a logikát követni, vagy ugyanazt csinálja, inkább hozz létre egy olyan metódust vagy függvényt, amit megtudsz hívni. Későbbi karbantartásoknál sokkal egyszerűbb. Például, ha sok helyen kell összeadni érdemes egy összeadás függvényt létrehozni, nem mindig leírni.

2. Melyik KISS (Keep It Simple, Stupid) elvet sérti az alábbi kód?

```
def calculate(a, b, operation):
    if operation == "add":
        return a + b
    elif operation == "subtract":
        return a - b
    elif operation == "multiply":
        return a * b
    elif operation == "divide":
        return a / b
    else:
        return "Invalid operation"
```

   Az elv lényege, hogy a kód egyszerű és könnyen átlátható maradjon, de ebben a kódban nagyon sok az if-elif-else ami bonyolulttá teszi a szerkezetet, nehezen karbantarthatóvá és nehezen bővíthetővé.

3. Hogyan kapcsolódik a YAGNI elv az alábbi megoldáshoz?

```
def save_to_database(data):
    # Implementálás később
    pass

def save_to_cloud(data):
    # Ez jelenleg nem szükséges, de megírtuk
    pass
```

   Az elv lényeg, hogy ne írj olyan funkciókat amire most nincs szükséged, hogy ne bonyolítsd és hosszabbítsd feleslegesen a kódot. Itt a save_to_cloud függvényre jelenleg nincs szükség és már megvan írva így ez YAGNI elvet sért.

4. Mi a különbség az osztályattribútumok és példányattribútumok között?

   Az osztályattribútomokhoz minden példány hozzáfér, de a példányattribútomhoz csak az adott példány.

5. Melyik OOP fogalom használata jelenik meg itt?

```
class Vehicle:
    def start_engine(self):
        raise NotImplementedError("Subclass must implement
abstract method")

class Car(Vehicle):
    def start_engine(self):
        print("Engine started!")
```

   Öröklődés. A szülő osztály előír egy metódust, de a megvalósítást a gyerek osztályra bízza.

II. Elméleti kérdések

1. Hogyan deklarálsz és inicializálsz egy változót Pythonban?

   változónév = érték

2. Milyen adattípus jön létre {'key': 'value'} használatával?

   dictionary

3. Milyen adattípusú a True és False érték?

   booleen

4. Hogyan jelölöd a szeletelést (slice) egy stringben?

   :, [kezdő index : utolsó index : lépésköz]

5. Milyen kulcsszóval definiálható egy függvény Pythonban?

   def

6. Milyen kulcsszót használsz egy modul importálására?

   import

7. Melyik beépített függvényt használod egy fájl megnyitására?

open(), with open() formátumban

8. Hogyan hozol létre egy osztályt Pythonban?

   class

9. Melyik OOP elvet mutatja a super() függvény használata?

   öröklődés

10. Mi az eredménye a 10 >> 1 bitwise műveletnek?

    5

III. Gyakorlati kérdések

11. Egészítsd ki a kódot, hogy a függvény visszatérjen egy lista hosszával.

```
def list_length(lst):
    # Kiegészítés szükséges
```

length = len(lst)

return length

12. Egészítsd ki a kódot úgy, hogy az eredmény True legyen!

```
a = [1, 2, 3]
b = [1, 2, 3]
print(a == b and a is ____ )
```

b

13. Mi lesz a kimenet?

```
numbers = [1, 2, 3, 4]
squared = [x ** 2 for x in numbers]
print(squared)
```

[1, 4, 9, 16]

14. Hogyan tudsz egy új kulcs-érték párt hozzáadni a szótárhoz?

```
my_dict = {'a': 1, 'b': 2}
# Kiegészítés szükséges
```

my_dict[’kulcs’] = ’c’

my_dict[’érték’] = ’3’

15. Mi lesz a result változó értéke?

```
result = True and False or True
```

True, mert and erősebb, mint az or, és az and csak akkor igaz ha mindkettő igaz, or akkor igaz ha az egyik igaz

16. Milyen kivételt dob a kód?

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

IndexError-t

17. Mi a hiba a kódban, és hogyan javítod?

```
def add_numbers(a, b):
    return a + b
print(add_numbers(5))
```

Két paramétert vár a függvény és csak egy van megadva.

18. Mi lesz a kimenet?

```
x = 7
while x > 3:
    print(x)
    x -= 1
    if x == 5:
        break
```

Kimenet:

7

6

19. Mi lesz a my_set értéke a kód futtatása után?

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

my_set = {1, 2, 3, 4}

20. Mi a hiányzó elem a kódban, hogy kivételkezelést használjunk?

```
try:
    x = 1 / 0
except ZeroDivisionError:
    print("Nullával való osztás!")
finally:
    print("Kész!")
```

Nincs olyan eset amikor nem dob kivételt.

21. Hogyan egészíted ki, hogy a reguláris kifejezés illeszkedjen a számokra?

```
import re
pattern = r"___"
result = re.search(pattern, "123abc")
print(result.group())
```

\d+

IV. Kötelezően választható feladatok:

Választott feladatom: A