

## Python OOP Tests (Uz papīra, ~40 min)

### 1. daļa: Koda izpratne (15 punkti)

1. uzdevums. Ko izvadīs šis kods? Paskaidro katru rindu. (5 punkti)

```
class Rectangle:
    def __init__(self, width, height):
        self.width = width
        self.height = height

    def change_orientation(self):
        self.width, self.height = self.height, self.width

    def is_square(self):
        return self.width == self.height

r = Rectangle(5, 8)
r.change_orientation()
print(r.width, r.height)
print(r.is_square())
```

2. uzdevums. Ko dara zemāk redzamā Point2D metode? Apraksti to saviem vārdiem. Ko tā atgriežs, ja p1 = Point2D(2, 3) un p2 = Point2D(5, 7)? (5 punkti)

```
import math
class Point2D:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def direction_to(self, other):
        dx = other.x - self.x
        dy = other.y - self.y
        magnitude = math.sqrt(dx**2 + dy**2)
        return dx / magnitude, dy / magnitude
```

3. uzdevums. Prognozē izvadi un paskaidro: (5 punkti)

```
class Triangle:
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def is_valid(self):
        return self.a + self.b > self.c and \
               self.a + self.c > self.b and \
               self.b + self.c > self.a

t1 = Triangle(3, 4, 5)
t2 = Triangle(1, 2, 3)
print(t1.is_valid())
print(t2.is_valid())
```

### 2. daļa: Klūdu labojumi un uzlabojumi (15 punkti)

4. uzdevums. Zemāk redzamā klase simulē bankas kontu, bet tajā ir vairākas klūdas. Nosauc vismaz **3 problēmas**, un pārraksti withdraw() un deposit() metodes pareizi. (7 punkti)

```

class BankAccount:
    def __init__(self, owner, balance):
        self.owner = owner
        self.balance = balance

    def deposit(self, amount):
        self.balance = amount

    def withdraw(self, amount):
        if amount > self.balance:
            print("Balance too low")
        self.balance -= amount

```

**5. uzdevums.** Uzraksti jaunu metodi `transfer_to(self, other_account, amount)`, kas:

- No konta `self` noņem norādīto summu
- Pievieno to kontam `other_account`
- Pārbauda, vai pietiek naudas pirms veikšanas Uzraksti šīs metodes kodu. (4 punkti)

**6. uzdevums.** Kas notiku, ja pārskaitījumu veiktu uz to pašu kontu? Kā tu to novērstu koda? (4 punkti)

**3. daļa: Izvades un dizaina izpratne (10 punkti)**

**7. uzdevums.** Uzraksti `__str__` metodi klasē zemāk, lai izvadot objektu tiktu rādīts: `Point(x=3, y=5)` (2 punkti)

```

class Point2D:
    def __init__(self, x, y):
        self.x = x
        self.y = y

```

**8. uzdevums.** Papildini `Triangle` klasi ar metodi `type()`, kas atgriež:

- "equilateral", ja visas malas vienādas
- "isosceles", ja divas vienādas
- "scalene", ja visas malas atšķirīgas (3 punkti)