## Зертханалық жұмыс - №3

# Объектіге бағытталған программалаудың элементтері

Жұмыстың мақсаты: Python программалау тілінде объектіге бағытталған программалау элементімен танысу

```
In [ ]: from IPython import display
```

#### Класс құруға мысал

```
In [ ]:
         display.Image('oop-1.png')
Out[]:
                           Circle
         -radius:double = 1.0
         +Circle()
         +Circle(radius:double)
         +getRadius():double
                                                        Use JDK constant Math.PI for \pi
         +setRadius(radius:double):void
         +getArea():double •---
         +getCircumference():double ●
                                                        "Circle[radius=?]"
         +toString():String •
In [ ]:
In [ ]:
         class Circle:
             def __init__(self, *args):
                if len(args) > 0:
                    if isinstance(args[0], int) or isinstance(args[0], float):
                        self.__radius = float(args[0])
                elif len(args) == 0:
                    self. radius = 1.0
             def getRadius(self)->float:
                return self.__radius
             def setRadius(self, radius:float):
                if radius > 0 and (isinstance(radius, float) or isinstance(radius, int)):
                    self.__radius = float(radius)
             def getArea(self)->float:
                return 3.1415*self. radius**2
             def getCircumference(self)->float:
                return 2*3.1415*self.__radius
             def str (self):
                return "Circle[radius = " + str(self.__radius) + "]"
```

```
In [ ]:
        c1 = Circle(1.1)
        print(c1)
       Circle[radius = 1.1]
In [ ]:
        c2 = Circle()
        print(c2)
       Circle[radius = 1.0]
In [ ]:
        c1.setRadius(2.2)
        print(c1)
        print(f"radius is: {c1.getRadius()}")
       Circle[radius = 2.2]
       radius is: 2.2
In [ ]:
        print(f"area is: {c1.getArea():.2f}")
        print(f'circumference is: {c1.getCircumference():.2f}')
       area is: 15.20
       circumference is: 13.82
      ООП-ның мұрагерлік қасиетіне мысал
In [ ]:
        display.Image('oop-8.png')
Out[]:
                         Circle
        -radius:double = 1.0
        -color:String = "red"
        +Circle()
        +Circle(radius:double)
        +Circle(radius:double,color:String)
        +getRadius():double
        +setRadius(radius:double):void
        +getColor():String
        +setColor(color:String):void
        +getArea():double
        +toString():String.
                                                     "Circle[radius=r,color=c]"
                             \ superclass
                  extends
                              subclass
                        Cylinder
        -height:double = 1.0
        +Cylinder()
        +Cylinder(radius:double)
        +Cylinder(radius:double,height:double)
        +Cylinder(radius:double,height:double,
           color:String)
        +getHeight():double
        +setHeight(height:double):void
        +getVolume():double
```

```
lab3
In [ ]: | class Circle2:
             def __init__(self, *args):
                 if len(args) == 0:
                     self.__radius = 1.0
                      self. color = "red"
                 elif len(args) == 1:
                      if isinstance(args[0], int) or isinstance(args[0], float):
                          self.__radius = float(args[0])
                          self.__color = "red"
                 elif len(args) == 2:
                     if (isinstance(args[0], int) or isinstance(args[0], float)) \
                          and isinstance(args[1], str):
                              self.__radius = float(args[0])
                              self.__color = args[1]
             def getRadius(self)->float:
                 return self. radius
             def setRadius(self, radius:float):
                 if radius > 0 and (isinstance(radius, float) or isinstance(radius, int)):
                      self.__radius = float(radius)
             def getArea(self)->float:
                 return 3.1415*self.__radius**2
             def getColor(self)->str:
                 return self.__color
             def setColor(self, color:str):
                 if isinstance(color, str):
                     self.__color = color
             def __str__(self):
                 return "Circle[radius = " + str(self.__radius) + ", color = " + self.__color
In [ ]:
         class Cylinder(Circle2):
```

```
def __init__(self, *args):
    if len(args) == 0:
        super().__init__()
        self.__height = 1.0
    elif len(args) == 1:
        super().__init__(args[0])
        self.\_height = 1.0
    elif len(args) == 2 and isinstance(args[1], float):
        super().__init__(args[0])
        self. height = args[1]
    elif len(args) == 3 and isinstance(args[1], float) \
        and isinstance(args[2], str):
        super().__init__(args[0], args[2])
        self.__height = args[1]
    else:
        raise ValueError
def getHeight(self)->float:
    return self.__height
def setHeight(self, height:float):
    if isinstance(height, float):
        self.__height = height
def getVolume(self)->float:
    return super().getArea() * self. height
def __str__(self):
```

```
return "Cylinder: sunclass of " + super().__str__() + \
                     " height = " + str(self.__height)
In [ ]:
         c1 = Cylinder()
         print(f"radius = {c1.getRadius()}")
         print(f"height = {c1.getHeight()}")
         print(f"area = {c1.getArea()}")
         print(f"volume = {c1.getVolume()}")
         print(c1)
        radius = 1.0
        height = 1.0
        area = 3.1415
        volume = 3.1415
        Cylinder: sunclass of Circle[radius = 1.0, color = red] height = 1.0
In [ ]:
        c2 = Cylinder(10.0)
         print(f"radius = {c2.getRadius()}")
         print(f"height = {c2.getHeight()}")
         print(f"area = {c2.getArea()}")
         print(f"volume = {c2.getVolume()}")
         print(c2)
        radius = 10.0
        height = 1.0
        area = 314.150000000000003
        volume = 314.15000000000003
        Cylinder: sunclass of Circle[radius = 10.0, color = red] height = 1.0
In [ ]:
         c3 = Cylinder(2.0, 10.0)
         print(f"radius = {c3.getRadius()}")
         print(f"height = {c3.getHeight()}")
         print(f"area = {c3.getArea()}")
         print(f"volume = {c3.getVolume()}")
         print(c3)
        radius = 2.0
        height = 10.0
        area = 12.566
        volume = 125.66000000000001
        Cylinder: sunclass of Circle[radius = 2.0, color = red] height = 10.0
In [ ]:
         c4 = Cylinder(2.0, 10.0, "blue")
         print(f"radius = {c4.getRadius()}")
         print(f"height = {c4.getHeight()}")
         print(f"area = {c4.getArea()}")
         print(f"volume = {c4.getVolume()}")
         print(f"color = {c4.getColor()}")
         print(c4)
        radius = 2.0
        height = 10.0
        area = 12.566
        volume = 125.66000000000001
        color = blue
        Cylinder: sunclass of Circle[radius = 2.0, color = blue] height = 10.0
```

### Тапсырмалар:

# Төмендегі класс диаграммаларға қарап python тілінде класстар құрыңыздар

```
In [ ]:
        display.Image('oop-2.png')
Out[]:
                        Circle
        -radius:double = 1.0
        -color:String = "red"
        +Circle()
        +Circle(radius:double)
        +Circle(radius:double,color:String)
        +getRadius():double
        +getColor():String
        +setRadius(radius:double):void
        +setColor(color:String):void
        +toString():String
        +getArea():double •
                                                  "Circle[radius=?,color=?]"
In [ ]:
        display.Image('oop-3.png')
Out[]:
                      Rectangle
        -length:float = 1.0f
        -width:float = 1.0f
        +Rectangle()
        +Rectangle(length:float,width:float)
        +getLength():float
        +setLength(length:float):void
        +getWidth():float
        +setWidth(width:float):void
        +getArea():double
        +getPerimeter():double
        +toString():String ◆
                                                "Rectangle[length=?,width=?]"
In [ ]:
        display.Image('oop-4.png')
Out[ ]:
```

```
Employee
        -id:int
        -firstName:String
        -lastName:String
        -salary:int
        +Employee(id:int,firstName:String,
          lastName:String,salary:int)
        +getID():int
        +getFirstName():String
                                                 "firstName lastname"
        +getLastName():String
        +getName():String '
                                                 salary * 12
        +getSalary():int
        +setSalary(salary:int):void
        +getAnnualSalary():int ◆
                                                 Increase the salary by the percent and
        +raiseSalary(int percent):int
                                                 return the new salary
        +toString():String.
                              "Employee[id=?,name=firstName lastname,salary=?]"
In [ ]:
        display.Image('oop-5.png')
Out[]:
                      Account
                                               Add amount to balance, return balance
        -id:String
        -name:String
        -balance:int = 0
                                               If amount <= balance
        +Account(id:String, name:String),
                                                subtract amount from balance
        +Account(id:String, name:String,
                                               else print "Amount exceeded balance"
          balance:int)
                                               return balance
        +getID():String
        +getName():String
                                               If amount <= balance
        +getBalance():int
                                                transfer amount to the given Account
        +credit(amount:int):int•
                                               else print "Amount exceeded balance"
        +debit(amount:int):int
                                               return balance
        +transferTo(another:Account,
          amount:int):int ◆
                                               "Account[id=?,name=?,balance=?]"
        +toString():String •
In [ ]:
        display.Image('oop-6.png')
Out[ ]:
```

file:///D:/Users/d.baimbetov/AUES/Materials/For\_GitHub/AUES\_ST/Labs/Lab-3/lab3.html

```
Ball
         -x:float
         -y:float
                                                  Each move step advances x and y
         -radius:int
                                                  by \Delta x and \Delta y. \Delta x and \Delta y could be
         -xDelta:float
                                                  positive or negative.
         -yDelta:float
         +Ball(x:float,y:float,radius:int
            xDelta:float,yDelta:float)
         +getX():float
         +setX(x:float):void
         +getY():float
         +setY(y:float):void
         +getRadius():int
         +setRadius(radius:int):void
                                                  Move one step:
         +getXDelta():float
                                                  x += \Delta x; y += \Delta y;
         +setXDelta(xDelta:float):void
         +getYDelta():float
                                                  \Delta x = -\Delta x
         +setYDelta(yDelta:float):void
         +move():void•-
                                                  \Delta y = -\Delta y
         +reflectHorizontal():void◆
         +reflectVertical():void◆
                                                  "Ball[(x,y), speed=(\Delta x, \Delta y)]"
         +toString():String•
In [ ]:
        display.Image('oop-7.png')
Out[]:
                        MyPoint
        -x:int = 0
        -y:int = 0
                                                  Return a 2-element int[] of {x,y}
        +MyPoint()
        +MyPoint(x:int,y:int)
                                                   (x,y)"
        +getX():int
        +setX(x:int):void
        +getY():int
                                                  Distance from this point to the
        +setY(y:int):void
                                                  given point at (x,y).
        +getXY():int[2]
        +setXY(x:int,y:int):void
                                                  Distance from this point to the
        +toString():String●
                                                  given instance of MyPoint.
        +distance(x:int,y:int):double
        +distance(another:MyPoint):double
                                                  Distance from this point to (0,0)
        +distance():double ◆
In [ ]:
        display.Image('oop-9.png')
Out[]:
```

```
Point
 -x:float = 0.0f
 -y:float = 0.0f
 +Point(x:float,y:float)
 +Point()
 +getX():float
 +setX(x:float):void
 +getY():float
 +setY(y:float):void
 +setXY(x:float,y:float):void
 +getXY():float[2]
                                           (x,y)
 +toString():String
           extends
              MovablePoint
-xSpeed:float = 0.0f
-ySpeed:float = 0.0f
+MovablePoint(x:float,y:float,
  xSpeed:float,ySpeed:float)
+MovablePoint(xSpeed:float,ySpeed:float)
+MovablePoint()
+getXSpeed():float
+setXSpeed(xSpeed:float):void
                                             "(x,y),speed=(xs,ys)"
+getYSpeed():float
+setYSpeed(ySpeed:float):void
+setSpeed(xSpeed:float,ySpeed:float):void
                                              x += xSpeed;
+getSpeed():float[2]
                                              y += ySpeed;
+toString():String•
                                              return this;
+move():MovablePoint •
```

```
In [ ]:
```