

Practical

1. Ստեղծեք **Circle** class-ը, որն ունի հետևյալ attribute-ները՝ **radius** և **color**: Class-ի ներսում ստեղծեք **getDesc(self)** մեթոդը, որը կտաի “A **color** circle with radius **radius**.”`

օգտագործելով համապատասխան attribute-ների արժեքները:

Փորձարկեք class-ի աշխատանքը ստեղծելով այդ class-ի object(ներ):

2. Ստեղծեք class որն ունի մեկ String տիպի attribute ` **my_str** և երկու method-ներ՝ **get_String(self)** և **print_String(self)**: **get_String** method-ը վերադարձնում է **my_str** attribute-ի արժեքը, իսկ **print_String** method-ը տպում է **my_str** attribute-ի արժեքը՝ դարձնելով բոլոր տառերը մեծատառ:

Փորձարկեք class-ի աշխատանքը ստեղծելով այդ class-ի object(ներ):

3. Ստեղծեք **Employee** class-ը, որն ունի հետևյալ attribute-ները՝ **name**, **last_name** և private attribute **monthly_salary**: Ստեղծեք **getFullName(self)** մեթոդը, որը կվերադարձնի “name last_name” ` օգտագործելով համապատասխան attribute-ների արժեքները: Ստեղծեք **annualSalary(self)** մեթոդը, որը կհաշվարկի տարեկան աշխատավարձը, օգտագործելով համապատասխան attribute-ների արժեքները, և կվերադարձնի “High”, եթե արժեքը >100 ու “Low”՝ հակառակ դեպքում:

Փորձարկեք class-ի աշխատանքը ստեղծելով այդ class-ի object(ներ):

4. Ստեղծեք **Car** class-ը, որն ունի հետևյալ attribute-ները՝ **model**, **color** և **max_speed**: Ստեղծեք **compareCar(self, car2)** մեթոդը, որը ստանում է ևս մեկ Car տիպի object որպես argument ու կվերադարձնում է “car1 is better than car2” եթե ձեր մեքենայի **maxSpeed** attribute-ի արժեքը մեծ է car2-ի **maxSpeed** attribute-ի արժեքից և “car2 is better than car1”՝ հակառակ դեպքում:

Փորձարկեք class-ի աշխատանքը ստեղծելով այդ class-ի object(ներ):

5. Ստեղծեք **Police_car** class-ը, որն ունի հետևյալ attribute-ները՝ **owner**, **price** և private attribute **pass_code**: Ստեղծեք class attribute **tax_value**, որի արժեքն է 0.2: Ստեղծեք **tax(self)** մեթոդը, որը հաշվում ու վերադարձնում է տվյալ մեքենայի համար վճարվելիք հարկը՝ **tax_value * price**: Ստեղծեք **greeting(self)** մեթոդը, որը կտաի “Welcome to your car, OWNER” OWNER-ի փոխարեն օգտագործելով **owner** փոփոխականի արժեքը, եթե **pass_code** փոփոխականի արժեքը “admin” է:

(OPTIONAL) Ավելացրեք set և get ֆունկցիաներ **pass_code** private փոփոխականի համար:

Փորձարկեք class-ի աշխատանքը ստեղծելով այդ class-ի object(ներ):

Inheritance

6. Ստեղծեք Animal class-ը:

Attributes: name

Methods: __init__(self, name) -> ստեղծում է name attribute

move(self) -> տպում է "I can move"

Ստեղծեք Dog class-ը, որը ժառանգում է Animal-ից:

Attributes: -

Methods: __init__(self) -> կանչում է Animal class-ի __init__ method-ը "Dog" արժեքով

Ստեղծեք Dog class-ի object ու կանչեք իր վրա move() method-ը:

Հիմա Dog class-ին ավելացրեք move(self) method-ը, որը տպում է "I can run really fast".

Ստեղծեք Dog class-ի object ու կանչեք իր վրա move() method-ը:

7. Ստեղծեք Animal class-ը:

Attributes: name, legs (նոքերի քանակ)

Methods: __init__(self, name, legs) -> ստեղծում է name ու legs attribute-ներ

getName(self) -> տպում է "My name is X", X-ի փոխարեն օգտագործելով name attribute-ի արժեքը

getLegs(self) -> տպում է "I have X legs", X-ի փոխարեն օգտագործելով legs attribute-ի արժեքը

Ստեղծեք Exnik class-ը, որը ժառանգում է Animal-ից:

Attributes: -

Methods: __init__(self) -> կանչում է Animal class-ի __init__ method-ը "Exnik" ու 4 արժեքներով

Ստեղծեք Exnik class-ի object ու կանչեք իր վրա getLegs() ու getName() method-ները:

8. Ստեղծեք Bird abstract class-ը:

Attributes: name, weight

Methods: __init__(self, name, weight) -> ստեղծում է name ու weight attribute-ներ
abstract method fly(self)

Ստեղծեք Hav class-ը, որը ժառանգում է Bird-ից:

Attributes: -

Methods: __init__(self, name, weight) -> կանչում է Bird class-ի __init__ method-ը name ու weight attribute-ներով

fly(self) -> print է անում "I believe I can fly"

Ստեղծեք Hav class-ի object ու կանչեք իր վրա fly method-ը:

Homework

1. Ստեղծեք **Person** class-ը.

Attributes: **name**, **last_name**, **age**, **gender**, **student** (սա boolean attribute է՝ այսինքն ընդունում է True/False արժեքներ), ինչպես նաև private attribute **password**

Methods:

Greeting(self, second_person) - ստանում է Person տիպի object որպես input, տպում է "Welcome dear X." `որտեղ X-ը **second_person**-ի name-ն է::

Goodbye(self) - տպում է "Bye everyone!"

Favourite_num(self, num1) - ստանում է integer տեսակի **num1**-ը որպես input և վերադարձնում է "My favourite number is **num1**" `օգտագործելով **num1** attribute-ի արժեքը.

Read_file(self, filename) - ստանում է String տիպի **filename** փոփոխականը ու փորձում է կարդալ այդ անունով ֆայլը՝ **filename**-ի վերջում ավելացնելով ".txt" ("**filename.txt**"). Կարդալու համար օգտագործեք open() ֆունկցիան: (փորձեք ինքնուրույն հասկանալ թե ինչպես :))

Ավելացրեք համապատասխան set ու get method-ներ password private attribute-ի համար:

2. Ստեղծեք Calculation class-ը:

Attributes: x, y

Methods: **__init__(self, x, y)** -> ստեղծում է x ու y attribute-ները

addition(self) -> տպում է x ու y արգումենտների արժեքների գումարը

subtraction(self) -> տպում է x ու y արգումենտների արժեքների տարբերությունը

Ստեղծեք MyCalculation class-ը, որը ժառանգում է Calculation class-ից:

Attributes: x, y

Methods: **__init__(self, x, y)** -> կանչում է Calculation class-ի **__init__** method-ը x ու y attribute-ներով

multiplication(self) -> տպում է x ու y արգումենտների արժեքների արտադրյալը

division(self) -> տպում է x ու y արգումենտների արժեքների քանորդը

Ստեղծեք MyCalculation class-ի object 3 ու 5 attribute-ների արժեքներով ու կանչեք իր վրա addition, subtraction, multiplication ու division method-ները:

3. Ստեղծեք My_Time class-ը:

Attributes: t (str, ցույց է տալիս ժամանակ, օրինակ՝ “10 AM”)

Methods: __init__(self, t) -> ստեղծում է t attribute

printTime(self) -> տպում է “The current time is X”, X-ի փոխարեն օգտագործելով t attribute-ի արժեքը

Ստեղծեք My_Date class-ը:

Attributes: d (str, ցույց է տալիս ամսաթիվ, օրինակ՝ “12.02.2018”)

Methods: __init__(self, d) -> ստեղծում է d attribute

printDate(self) -> տպում է “The current date is Y”, Y-ի փոխարեն օգտագործելով d attribute-ի արժեքը

Ստեղծեք Date_Time class-ը, որը ժառանգում է My_Date ու My_Time class-ներից:

Attributes: t, d

Methods: __init__(self, d, t) -> կանչում է My_Date class-ի __init__ method-ը d attribute-ով ու My_Time class-ի __init__ method-ը t attribute-ով

Ստեղծեք Date_Time class-ի object “12 PM” ու “13.03.2013” attribute-ների արժեքներով ու կանչեք իր վրա printTime ու printDate method-ները:

4. Ստեղծեք Model class-ը:

Attributes: name

Methods: __init__(self, name) -> ստեղծում է name attribute

printModel(self) -> տպում է “The model of the vehicle is X”, X-ի փոխարեն օգտագործելով name attribute-ի արժեքը

Ստեղծեք Color class-ը:

Attributes: color

Methods: __init__(self, color) -> ստեղծում է color attribute

printColor(self) -> տպում է “The color of the vehicle is Y”, Y-ի փոխարեն օգտագործելով color attribute-ի արժեքը

Ստեղծեք Car class-ը, որը ժառանգում է Model ու Color class-ներից:

Attributes: model, color

Methods: __init__(self, model, color) -> կանչում է Model class-ի __init__ method-ը model attribute-ով ու Color class-ի __init__ method-ը color attribute-ով

Ստեղծեք Car class-ի object “BMW” ու “red” attribute-ների արժեքներով ու կանչեք իր վրա printModel ու printColor method-ները:

