Practical

1. Create the class **Circle** with attributes **radius** and **color**. Inside your class, create the method **getDesc(self)** which should print the text "A **color** circle with radius **radius**.", using the values of the corresponding attributes**.**
Create class object(s) and test your class.

2. Create a class which has 1 attribute **my_str** of type String and 2 methods: **get_String(self)** and **print_String(self)**. The method **get_String(self)** returns the value of the attribute **my_str.** The method **print_String(self)** prints the value of the attribute **my_str,** making all the letters uppercase**.**
Create class object(s) and test your class.

3. Create the class **Employee,** which has the following attributes: **name**, **last_name** and a private attribute **monthly_salary**. Inside your class, create the method **getFullName(self),** which will return "**name last_name**", using the values of the corresponding attributes. Create the method **annualSalary(self)**, which will calculate the annual salary of the employee, using the values of the corresponding attributes and will return "High" in case the salary is >100 and "Low", otherwise.
Create class object(s) and test your class.

4. Create the class **Car** with the following attributes: **model**, **color** and **max_speed**. Inside the class, create the method **compareCar(self, car2)** which gets an object of type **Car** as an arument and returns the text "car1 is better than car2" if the **maxSpeed** attribute of your car is larger than the **maxSpeed** attribute of car2 and returns the text "car2 is better than car1"` otherwise.
Create class object(s) and test your class.

5. Create the class **Police_car** which has the following attributes: **owner**, **price** and a private attribute **pass_code.** Create a class attribute **tax_value** with a value 0.2. Create the method **tax(self),** which returns the tax you are supposed to pay for the car using the following formula: **tax_value * price**. Create the method **greeting(self)** which prints the text "Welcome to your car, **owner**", using the value of the attribute **owner,** only if the value of the attribute **pass_code** is "admin".

(OPTIONAL) Add set and get methods for the private attribute **pass_code**.

Create class object(s) and test your class.

Inheritance

6. Create the class **Animal.**
Attributes: name
Methods: __init__(self, name) -> creates the attribute name
move(self) -> prints “I can move”

Create the class **Dog** which inherits from the class **Animal**.
Attributes: -
Methods: __init__(self) -> calls the __init__ method of the **Animal** with a value “Dog”

Create an object of class **Dog** and call the method move() from the object.
Now, add a method move(), which prints “I can run really fast”.
Create an object of class **Dog** and call the method move() from the object.

7. Create the class **Animal.**
Attributes: name, legs (the number of legs)
Methods: __init__(self, name, legs) -> creates the attributes name and legs
getName(self) -> prints “My name is X”, using the value of the attribute **name** instead of X.
getLegs(self) -> prints “I have X legs”, using the value of the attribute **legs** instead of X.

Create the class **Exnik** which inherits from the class **Animal**
Attributes: -
Methods: __init__(self) -> calls the __init__ method of the Animal class with a value “Exnik”

Create an object of class **Exnik** and call the methods getLegs() and getName() from the object.

8. Create the class **Bird**
Attributes: name, weight
Methods: __init__(self, name, weight) -> creates the attributes name and weight
abstract method fly(self)

Create the class **Hav** which inherits from the class **Bird**
Attributes: -
Methods: __init__(self, name, weight) -> calls the __init__ method of the Bird class with the values of attributes name and weight
fly(self) -> prints “I believe I can fly”

Create an object of class **Exnik** and call the method fly() from the object.

Homework

1. Create the class **Person**.

   Attributes: **name**, **last_name**, **age**, **gender, student** (this is a boolean attribute i.e.it takes values True/False), as well as a private attribute **password**

   Methods:
   Greeting(self, second_person) - gets an object of type Person as an input and prints "Welcome dear **X**.", where **X** is the value of the **name** attribute of **second_person.**
   Goodbye(self) - prints "Bye everyone!"
   Favourite_num(self, num1) - gets an integer **num1** as an input and returns the text "My favourite number is **num1**", using the value of the attribute **num1**.
   Read_file(self, filename) - gets a String **filename** as an input and tries to read the file with the name "**filename.**txt**",** adding ".txt" at the end of the value of the attribute **filename.** Use the function **open()** to open the file. (try to do this on your own :))

   Add set and get methods for the attribute **password**.

2. Create a class Calculation.

Attributes: x, y
Methods: __init__(self, x, y) -> creates the attributes x and y
addition(self) -> prints the sum of the arguments x and y
subtraction(self) -> prints the difference of the arguments x and y

Create a class MyCalculation which inherits from the class Calculation.
Attributes: x, y
Methods: __init__(self, x, y) -> calls the __init__ metho of the Calculation class with parameters x and y
multiplication(self) -> prints the product of the arguments x and y
division(self) -> prints the quotient of the arguments x and y

Create an object of the MyCalculation class with attribute values 3 and 5 and call the methods addition, subtraction, multiplication and division on it.

3. Create a class My_Time

Attributes: t (of type str, which shows time, e.g. "10 AM")
Methods: __init__(self, t) -> creates the attribute t
printTime(self) -> prints "The current time is X", using the value of the attribute x instead of X

Create a class My_Date
Attributes: d (of type str, shows the date, e.g. “12.02.2018”)
Methods: __init__(self, d) -> creates the attribute d
printDate(self) -> prints “The current date is Y”, using the value of the attribute d instead of Y

Creates a class Date_Time which inherits from the classes My_Date and My_Time
Attributes: d, t
Methods: __init__(self, d, t) -> calls the __init__ method of the class My_Date with the parameter d and the __init__ method of the class My_Time with the parameter t.

Create an object of the class Date_Time with the attribute values “12 PM” and “13.03.2013” and call the methods printTime and printDate on it.

4. Create the class Model.

Attributes: name
Methods: __init__(self, name) -> creates the attribute name
printModel(self) -> prints “The model of the vehicle is X”, using the value of the attribute name instead of X.

Create the class Color.
Attributes: color
Methods: __init__(self, color) -> creates the attribute color
printColor(self) -> prints “The color of the vehicle is Y”, using the value of the attribute color instead of Y.

Create the class Car which inherits from the classes Model and Color
Attributes: model, color
Methods: __init__(self, model, color) -> calls the __init__ method of the class Model with the attribute model and calls the __init__ method of the class Color with the attribute color.

Create an object of type Car with values “BMW” and “red”, call printModel and printColor methods from the object.