

**Львівський національний університет імені Івана Франка**

**Факультет електроніки та комп'ютерних технологій**

# **ЗВІТ**

**Про виконання лабораторної роботи №2**

**Декоратор @property, використання getter, setter, deleter**

**Виконав:**

**Студент групи ФЕП-11**

**Сворень Ярослав**

**Перевірив:**

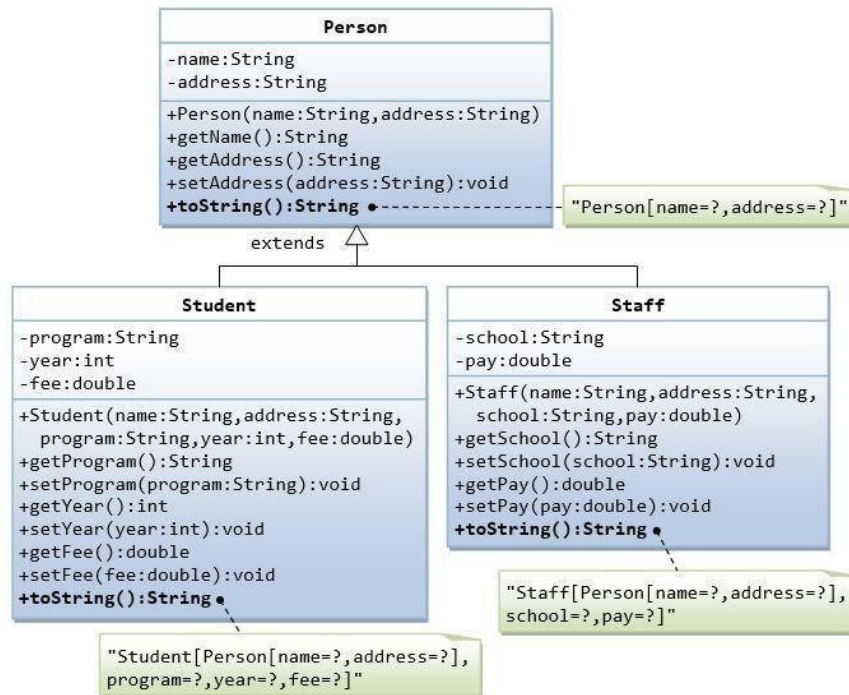
**Ас. Сінькевич О. О.**

**Львів 2020**

**Практичне Завдання: 1. Розібрати UML-діаграму, щоб правильно оформити зв'язки між класами. 2. Описати цю діаграму на мові Python.**

## Хід Роботи:

### 1. Ознайомлюємось з діаграмою.



### 2. Перепишемо діаграму на мові Python.

```
1 class Person():
2     def __init__(self, name="Ivan", address="Universitetska 1"):
3         self.__name = name
4         self.__address = address
5     @property
6     def get_name(self):
7         return self.__name
8     @property
9     def get_address(self):
10        return self.__address
11    @get_address.setter
12    def set_address(self, address:str):
13        self.__address = address
14    def to_string(self):
15        print("Person", self.get_name + ", ", self.get_address, "\n")
16
```

```

17 class Student(Person):
18     def __init__(self, name="Ivan", address="Universitetska 1", program="Software Engineering", year=2020, fee=31103.5):
19         super().__init__(name, address)
20         self.__program = program
21         self.__year = year
22         self.__fee = fee
23     @property
24     def get_program(self):
25         return self.__program
26     @get_program.setter
27     def set_program(self, program: str):
28         self.__program = program
29     @property
30     def get_year(self):
31         return self.__year
32     @get_year.setter
33     def set_year(self, year: int):
34         self.__year = year
35     @property
36     def get_fee(self):
37         return self.__fee
38     @get_fee.setter
39     def set_fee(self, fee: float):
40         self.__fee = fee
41     def to_string(self):
42         print("Student", self.get_name + ", ", self.get_address + ", ", self.get_program + ", ", str(self.get_year) + ", ", str(self.set_fee), "\n")

```

```

44 class Staff(Person):
45     def __init__(self, name="Ivan", address="Universitetska 1", school="Ivan Franko National University of Lviv", pay=15675.5):
46         super().__init__(name, address)
47         self.__school = school
48         self.__pay = pay
49     @property
50     def get_school(self):
51         return self.__school
52     @get_school.setter
53     def set_school(self, school: str):
54         self.__school = school
55     @property
56     def get_pay(self):
57         return self.__pay
58     @get_pay.setter
59     def set_pay(self, pay: float):
60         self.__pay = pay
61     def to_string(self):
62         print("Staff", self.get_name + ", ", self.get_address + ", ", self.get_school + ", ", str(self.get_pay), "\n")
63
64 yaroslav1 = Person()
65 yaroslav1.to_string()
66 yaroslav2 = Student("Yaroslav", "Liberty Avenue 19")
67 yaroslav2.to_string()
68 yaroslav3 = Staff("Oleh", "Dragomanova 50")
69 yaroslav3.to_string()

```

```

PS C:\Users\Charlie> python -u "d:\Programs\Patterns\lab2\Lab2PY.py"
Person Ivan, Universitetska 1

```

```

Student Yaroslav, Liberty Avenue 19, Software Engineering, 2020, 31103.5

```

```

Staff Oleh, Dragomanova 50, Ivan Franko National University of Lviv, 15675.5

```

**Висновок:** на цій лабораторній роботі я навчився працювати з декоратором `@property`, зрозумів принцип роботи геттерів і сеттерів для атрибута з функції.