

Лабораторна робота 2

Наслідування та поліморфізм у об'єктах пайтон

Виконала:
студентка групи МІТ-31
Півторак Каріна

Мета: ознайомитися із поняттям наслідування в ооп, навчитися створювати класи та об'єкти з наслідуванням у пайтон

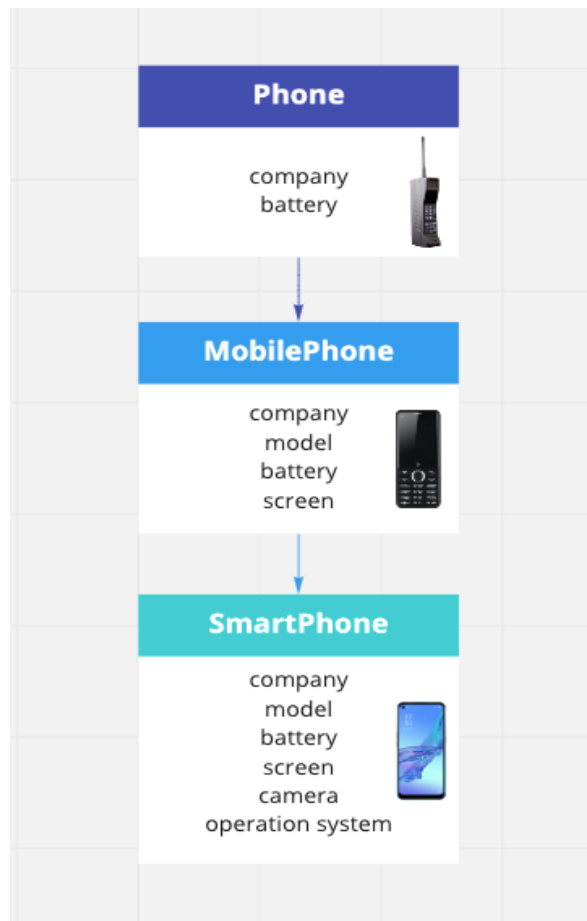
Завдання:

6	Предметна область: описати ієрархію мобільних телефонів для симуляції роботи мережі зв'язку (три види телефонів, від найпростішого до найскладнішого). Кожен телефон має батарею живлення, яка встановлюється йому в момент створення і не може бути передана іншому телефону.
---	---

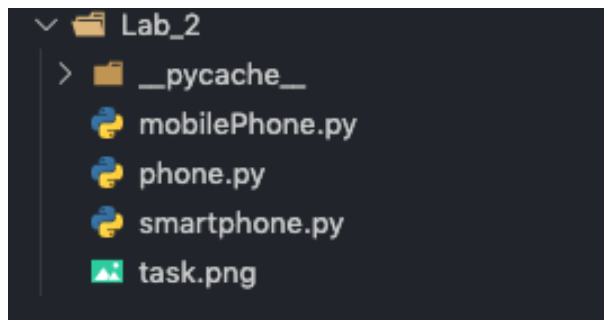
Репозиторій: <https://github.com/KariSpace/python-oop-labs.git>

Хід виконання лабораторної роботи

Схема ієрархії:



Для зручності створюємо окремий файл під кожен клас



phone.py:

```
import re
class Phone:

    def __init__(self, company, battery):
        self.company = company
        self.type = "antenna"
        self.battery = battery

    def __str__(self):
        """Return a descriptive string for this instance, invoked by
        print() and str()"""
        return f'\nThis is a {self.type} phone, with battery :
{self.battery}'

    def call(self, phone_number):
        regex = r'\d{3}-\d{3}-\d{4}'

        if re.search(regex, phone_number):
            print(f'\nCalling to {phone_number}...')
        else:
            print("Invalid phone number")

# m1 = Phone("Siemens", 500)
# m1.call("349-345-0985")
# print(m1)
```

Цей клас має два поля: компанія та батарея

І два методи:

`__str__` - магічний метод
`call` - метод класу phone

Запускаємо код:

```
This is a antenna phone, with battery : 500
Calling to 349-345-0985...
(base) Kari:OOP kari$
```

mobilePhone.py:

```
import re
from phone import Phone

class MobilePhone(Phone):

    def __init__(self, company, battery, screen):
        super().__init__(company, battery)

        self.screen = screen
        self.type = "mobile"

    def __str__(self):
        """Return a descriptive string for this instance, invoked by print()
        and str()"""
        return f'\nThis is a {self.type} phone, with battery : {self.battery},
        and {self.screen} screen'

    def send_sms(self, phone_number, sms_text):
        regex = r'\d{3}-\d{3}-\d{4}'
        if re.search(regex, phone_number):
            print(f'Sending sms to {phone_number} with text: {sms_text}')
        else:
            print("Cant send sms to invalid phone number")

m1 = MobilePhone("Samsung", 900, "collor")
m1.call("899-455-9095")
m1.send_sms("899-455-9095", "Hi! How are you?")
print(m1)
```

Цей клас має два поля: компанія та батарея

І два методи:

- `__str__` - магічний метод класу MobilePhone
- `send_sms` - метод класу MobilePhone

Запускаємо код:

```
Calling to 899-455-9095...

Sending sms to 899-455-9095 with text: Hi! How are you?

This is a mobile phone, with battery : 900, and collor screen
(base) Kari:00P kari$
```

smartphone.py:

```
import random
from mobilePhone import MobilePhone
from phone import Phone

class SmartPhone(MobilePhone):

    def __init__(self, company, model, battery, screen, camera, operation_system):
        MobilePhone.__init__(self, company, battery, screen)
        self.model = model
        self.screen = screen
        self.type = "smart"
        self.camera = camera
        self.operation_system = operation_system

    def __str__(self):
        """Return a descriptive string for this instance, invoked by print() and
        str()"""
        return f'\nThis is a {self.type} phone, with battery : {self.battery}, and
        {self.screen} screen, {self.camera} camera, {self.operation_system}
        operation_system,'

    def update_os(self):
        if random.choice([True, False]):
            print(f'Started updating your {self.operation_system}...\nDone!')
        else:
            print("Everything was updated")

m1 = SmartPhone("Samsung", "s20", 2500, "Full HD", "20MP", "Android")
m1.call("899-455-9095")
m1.send_sms("899-455-9095", "Hi! How are you?")
m1.update_os()
print(m1)
```

Цей клас має два поля: компанія та батарея

І два методи:

`__str__` - магічний метод класу MobilePhone
`update_os` - метод класу SmartPhone

Запускаємо код:

```
(base) Kari:00P kari$ /usr/local/bin/python3 /Users/kari/Desktop/Kari_Labs/00P/Lab_2/smartphone.py  
Calling to 899-455-9095...  
Sending sms to 899-455-9095 with text: Hi! How are you?  
Started updating your Android...  
Done!  
  
This is a smart phone, with battery : 2500, and Full HD screen, 20MP camera, Android operation_system,  
(base) Kari:00P kari$
```

Висновок: Було створено три класи, кожен з яких наслідує попередній. Екземпляри цих класів мають як свої власні методи, так і ті, що були унаслідовані від попереднього.