**ЗАКЛАД ВИЩОЇ ОСВІТИ**

**«УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА»**

**Факультет суспільних і прикладних наук**

**Кафедра інформаційних технологій**

ПРОГРАМУВАННЯ МОВОЮ PYTHON

**ПРАКТИЧНА РОБОТА № 4**

Виконав:

студент групи ІПЗс-21-1,

Бойчук Богдан Романочич

Перевірив:

викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата \_\_\_\_\_\_\_\_\_\_

**Івано-Франківськ**

**2024**

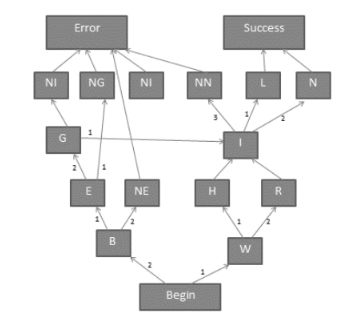
**Тема:** ООП. Наслідування

**Мета:** ознайомитися з особливостями реалізації наслідування (звичайного та множинного) атрибутів класа в ООП на мові Python. Об’єкт дослідження – принципи ООП, клас.

**Хід роботи**

**Завдання 1.**

Для класів Error і Success задати оброблювачі повідомлень, які при зверненні до них повертають повідомлення «Error» або «Success» (успіх), відповідно. Викликати обробник для класів «Begin» і «B». Визначити, які повідомлення виводяться при цьому? Пояснити чому.



**Код:**

class Error:

    def message\_handler(self):

        return "Error"

class Success:

    def message\_handler(self):

        return "Success"

# Класи, що наслідують Error

class NG(Error):

    pass

class NI(Error):

    pass

class NN(Error):

    pass

class NE(Error):

    pass

class I:

    pass

class G(NI, I):

    pass

class E(G):

    pass

class B(NE, E):

    pass

class L(Success):

    pass

class N(Success):

    pass

class I(NN, L, N):

    pass

class H(I):

    pass

class R(I):

    pass

class W(R, H):

    pass

class Begin(B):

    pass

begin = Begin()

b = B()

print("Begin state handler:", begin.message\_handler())

print("B state handler:", b.message\_handler())

ng = NG()

ni = NI()

g = G()

e = E()

ne = NE()

w = W()

r = R()

h = H()

i = I()

l = L()

n = N()

success = Success()

error = Error()

print("NG state handler:", ng.message\_handler())

print("NI state handler:", ni.message\_handler())

print("G state handler:", g.message\_handler())

print("E state handler:", e.message\_handler())

print("NE state handler:", ne.message\_handler())

print("W state handler:", w.message\_handler())

print("R state handler:", r.message\_handler())

print("H state handler:", h.message\_handler())

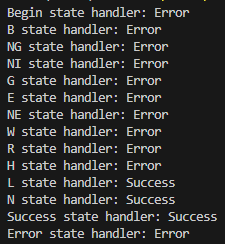
print("L state handler:", l.message\_handler())

print("N state handler:", n.message\_handler())

print("Success state handler:", success.message\_handler())

print("Error state handler:", error.message\_handler())

**Результат:**



**Пояснення:**

Коли викликається метод message\_handler для об'єкта класу Begin, Python буде шукати цей метод у класах у наступному порядку:

* Begin → немає метода message\_handler, тому шукаємо в базовому класі.
* B → немає метода message\_handler, шукаємо далі.
* NE → є метод message\_handler, який успадкований від Error.

Тому результатом для об'єкта begin буде "Error".

Аналогічно, для об'єкта класу B порядок пошуку буде таким:

* B → немає метода message\_handler, шукаємо в базовому класі.
* NE → є метод message\_handler, який успадкований від Error.

Результатом для об'єкта b також буде "Error".