

Практичне заняття №4

з навчальної дисципліни
Спеціалізовані мови програмування

на тему:

ОБЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Створення екземпляра класу

```
# Оголошення порожнього класу MyClass
class MyClass: pass
obj = MyClass()
# Об'єкт obj - це екземпляр класу MyClass,
# (він має тип MyClass)
print(type(obj)) # <class '__main__.MyClass'>
# MyClass – це клас, він є об'єктом, екземпляром метакласу type
# який є абстракцією поняття типу даних
print(type(MyClass)) # <class 'type'>
# Тому з класами можна виконувати операції як із об'єктами
# наприклад, копіювання
AnotherClass = MyClass
print(type(AnotherClass))
# тепер AnotherClass – це те ж саме, що і MyClass,
# і obj є екземпляром класу AnotherClass
print(isinstance(obj, AnotherClass)) # True
```

```
class MyClass:
```

```
    int_field = 8
```

```
    str_field = 'a string'
```

```
# Звернення до атрибутів класу
```

```
print (MyClass.int_field); print (MyClass.str_field)
```

```
# Створення двох екземплярів класу
```

```
object1 = MyClass (); object2 = MyClass ()
```

```
# Звернення до атрибутів класу через його екземпляри
```

```
print (object1.int_field); print (object2.str_field)
```

```
# Всі перераховані вище звернення до атрибутів насправді відносяться
```

```
# до двох одних і тих самих змінних
```

```
# Зміна значення атрибута класу
```

```
MyClass.int_field = 10
```

```
print (MyClass.int_field); print (object1.int_field); print(object2.int_field)
```

```
# Однак, аналогічно до глобальних і локальних змінних,
```

```
# присвоєння значення атрибуту об'єкта не змінює значення
```

```
# атрибута класу, а веде до створення атрибута даних (нестатичного поля)
```

```
object1.str_field = 'another string'
```

```
print(MyClass.str_field); print(object1.str_field);
```

```
print(object2.str_field)
```

Оголошення класу *MyClass* з двома атрибутами *int_field*, *str_field*, які є змінними

Атрибути-дані аналогічні полям. Їх не треба описувати: як і змінні, вони створюються в момент першого присвоювання.

Клас, який описує людину

class Person: pass

Створення екземплярів класу

alex = Person()

alex.name = 'Alex'; alex.age = 18

john = Person()

john.name = 'John'; john.age = 20

Атрибути-дані відносять тільки до окремих екземплярів класу

і ніяк не впливають на значення відповідних атрибутів-даних інших екземплярів

print(alex.name, 'is', alex.age); print(john.name, 'is', john.age)

Атрибутами класу можуть бути й методи-функції

Клас, який описує людину

class Person:

Перший аргумент, який вказує на поточний екземпляр класу,

прийнято називати self

def print_info(self):

print(self.name, 'is', self.age)

Створення екземплярів класу

alex=Person(); alex.name='Alex'; alex.age=18

john = Person(); john.name = 'John'; john.age = 20

Перевіримо, чим є атрибут-функція print_info класу Person

print(type(Person.print_info)) # функція (<class 'function'>)

Викличемо його для об'єктів alex і john

Person.print_info(alex)

Person.print_info(john)

Метод – функція, зв'язана з об'єктом. Всі атрибути класу, які є

функціями, описують відповідні методи екземплярів даного класу

print(type(alex.print_info)) # метод (<class 'method'>)

Виклик методу print_info

alex.print_info()

john.print_info()

Створимо клас, значення початкових атрибутів
(з методу `__init__`) якого залежить від
переданих аргументів при створенні об'єктів.
Далі ці атрибути об'єктів, створених на основі
даного класу, можна змінювати за допомогою
методів.

```
class Building:
    def __init__(self,w,c,n=0):
        self.what=w; self.color=c; self.numbers=n;
self.mwhere(n)
    def mwhere(self,n):
        if n <= 0: self.where = "відсутні"
        elif 0 < n < 100: self.where = "малий склад"
        else: self.where = "основний склад"
    def plus(self,p):
        self.numbers = self.numbers + p
        self.mwhere(self.numbers)
    def minus(self,m):
        self.numbers=self.numbers-m
        self.mwhere(self.numbers)
```

```
m1 = Building("дошки", "білі",50)
m2 = Building("дошки", "коричневі", 300)
m3 = Building("цегла","біла")
print (m1.what,m1.color,m1.where)
print (m2.what,m2.color,m2.where)
print (m3.what,m3.color,m3.where)
m1.plus(500);print (m1.numbers, m1.where)
```


Лекцію закінчено.
Дякую за увагу!