

## Приклад №18

### Спадкування

```
example18.py > ...
1  # Спадкування
2
3  class Table():
4      item = "table"
5      amount = 0
6
7      def __init__(self, w=1.5, h=1.5, material="wood"):
8          self.width = w
9          self.height = h
10         self.material = material
11         Table.amount += 1
12
13     def __del__(self):
14         Table.amount -= 1
15
16
17 class Desktable(Table):
18     def square(self):
19         return self.width*self.height
20
```

```
table1 = Desktable()
table2 = Table(2, 2.5, "iron")
table3 = Desktable(1, material="plastic")
```

```
print(Table.amount)
print("table 1:", table1.__dict__)
print("table 2:", table2.__dict__)
print("table 3:", table3.__dict__)
```

```
3
table 1: {'width': 1.5, 'height': 1.5, 'material': 'wood'}
table 2: {'width': 2, 'height': 2.5, 'material': 'iron'}
table 3: {'width': 1, 'height': 1.5, 'material': 'plastic'}
>>>
```

```
print(table1.square())
```

```
>>> 2.25
```

```
try:
    print(table2.square())
except:
    print("Клас Table не має функції square")
```

```
>>> Клас Table не має функції square
```

## Поліморфізм

```
# Поліморфізм

class Vehicle:
    def print_details(self):
        print("Це батьківський метод класа Vehicle")

class Car(Vehicle):
    def print_details(self):
        print("Це дочірній метод класа Car")

class Cycle(Vehicle):
    def print_details(self):
        print("Це дочірній метод класа Cycle")
```

```
machine1 = Vehicle()
machine2 = Car()
machine3 = Cycle()
```

```
machine1.print_details()
machine2.print_details()
machine3.print_details()
```

```
>>> Це батьківський метод класа Vehicle
      Це дочірній метод класа Car
      Це дочірній метод класа Cycle
```

## Інкапсуляція

```
# Інкапсуляція

class Person():
    def __init__(self, name, surname, age):
        self.name = name
        self.__surname = surname
        self._age = age

    def get_surname(self):
        return self.__surname
```

```
human = Person("Іван", "Крузенштерн", 42)
print(human.name)
```

```
print(human.name)
```

```
>>> Іван
```

```
print(human._age) # Ця функція буде недоступна при імпорті класу до іншого файлу
```

```
>>> 42
```

```
try:
    print(human.__surname)
except:
    print("Атрибут surname має приватний модифікатор доступу\n\
Для його виведення потрібно використовувати метод get_surname")
```

```
>>> Атрибут surname має приватний модифікатор доступу
Для його виведення потрібно використовувати метод get_surname
```

```
print(human.get_surname())
```

```
>>> Крузенштерн
```