6 ІНФОРМАТИКА

Поняття події та методу

За навчальною програмою 2017 року

Урок 19

Розділ 3
§ 16

Події. Методи

Коли ми складаємо програму, то намагаємось змоделювати певну реальну ситуацію або явище.

Загалом потрібно виконати такі кроки:

1. *виявити об’єкти, якими буде маніпулювати наша програма;*
2. *описати властивості об’єктів;*
3. *вказати методи (набір дій об’єкта);*
4. *запрограмувати обробку подій (що об’єкт повинен зробити у відповідь на подію).*

Розділ 3
§ 16

Поняття події

У реальному світі події відбуваються навколо нас безперервно. Наприклад, коли собаці Рексу дають команду, він виконує певні дії. У комп'ютерному світі подіями можуть бути натискання кнопки або переміщення миші.

***Подія** — це вплив на об'єкт, що відбувається в програмі.*

***Методи** — це дії, що можуть виконувати об'єкти даного класу.*

Розділ 3
§ 16

Поняття події

Якщо умовно,

об'єкти вважати іменниками	***то їхні методи — дієсловами***

Якщо уявити об'єкт як окрему річ, то його методи визначають, як він взаємодіє з іншими речами.

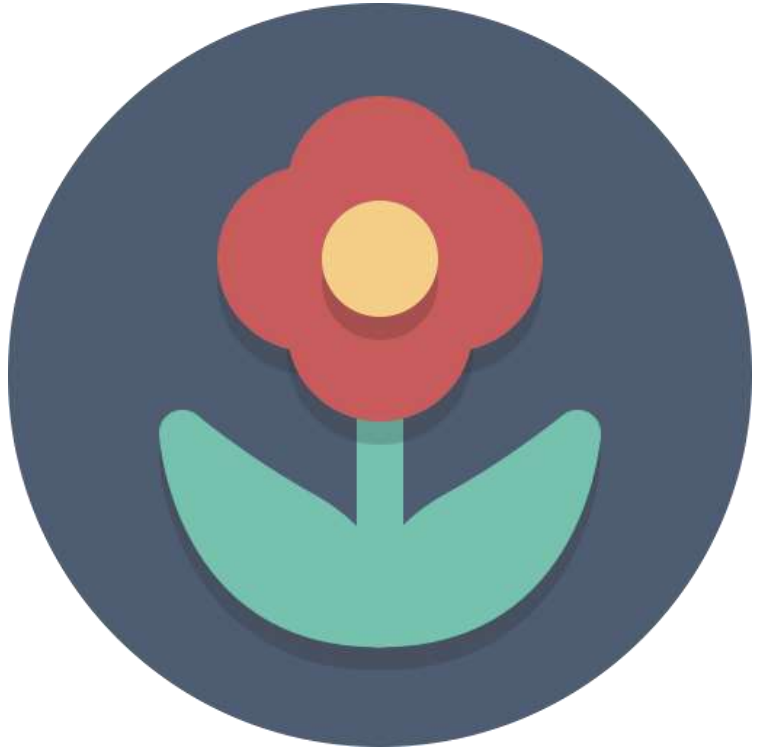
Якщо в програмі для класу Тварини визначити метод їсти, то їсти можуть усі об'єкти цього класу. Визначивши в класі властивості й методи, ми моделюємо об'єкти, властивості які вони мають і дії які можуть здійснювати

Розділ 3
§ 16

Поняття події

Методи виконуються об'єктом у відповідь на події. Коли з об'єктом Рекс відбулася подія Подано команду "Голос!" , він гавкає — виконує дію.

Нехай ми хочемо, щоб при натисканні на певну кнопку малювалася квітка. Для того щоб за допомогою програми намалювати квітку, необхідно дати опис цієї дії, тобто скласти набір покрокових інструкцій, які визначають порядок малювання квітки.



Розділ 3
§ 16

Поняття події

6

Розглянемо взаємозв'язок між властивостями, методами й подіями об'єктів класу Тварини.

Об'єкт	Власти-вість	Метод	Подія
Слон Мавпа Лев	Маса Раціон	Їсти Спати Гарчати	Настала ніч Зголоднів

Подія: зголоднів

Метод: гарчати

Розділ 3
§ 16

Опис методів

Методи можуть змінювати значення властивостей (атрибутів) об'єкта, виконувати інші дії над об'єктами.

Синтаксис заголовка методу:

```
def ім'я_методу(self, <перелік параметрів>):
```

Замість параметра self у разі виконання методу підставляється ім'я конкретного об'єкта.

Розділ 3
§ 16

Опис методів

Ми вже створили клас Dog. Відомо, що більшість собак вміють сідати й подавати голос за командою. Включимо в клас Dog два види дій (сідати й подавати голос).

```
class Dog():
    def _ _init_ _(self, name, age):
        self.name = name
        self.age = age
    def sit(self ):          # Собака сідає за командою
        print(self.name(), ' сідає.')
    def voice(self ):    # Собака подає голос за командою
        print(self.name(), ' гавкає. Гав-гав!')
```

```
def sit(self ):
    .....
my_dog.list()
```

Опис методів

У класі Dog ми визначили два методи:

sit() **і** **voice()**

Поки що методи sit() і voice() обмежуються простим виведенням повідомлення про те, що собака сідає або гавкає. Оскільки цим методам не потрібна додаткова інформація (кличка або вік), вони визначаються з єдиним параметром self. Екземпляри, які будуть створені пізніше, зможуть викликати ці методи.

Розділ 3
§ 16

Виклик методів

Методу необхідно «знати», дані якого об'єкта йому потрібно буде опрацьовувати. Для цього йому як перший аргумент передається ім'я екземпляра класу.

Виклик методу для конкретного об'єкта має вигляд:

```
об'єкт.ім'я_методу(<перелік значень>)
```

Створимо екземпляр my_dog на основі класу Dog:

```
my_dog = Dog('Рекс', 5)
```

Розділ 3
§ 16

Виклик методів

Щоб викликати метод для екземпляра my_dog, укажемо ім'я екземпляра і метод, який викликається, розділивши їх крапкою:

```
my_dog.sit()
my_dog.voice()
```

*Під час обробки команди **my_dog.sit()** середовище **Python** шукає метод **sit()** у класі **Dog** і виконує його код. Так само здійснюється обробка команди **my_dog.voice()**.*

Об'єкт виконує команди. Отримуємо дії:

```
Рекс сідає.
Рекс гавкає. Гав-гав!
```

Розділ 3
§ 16

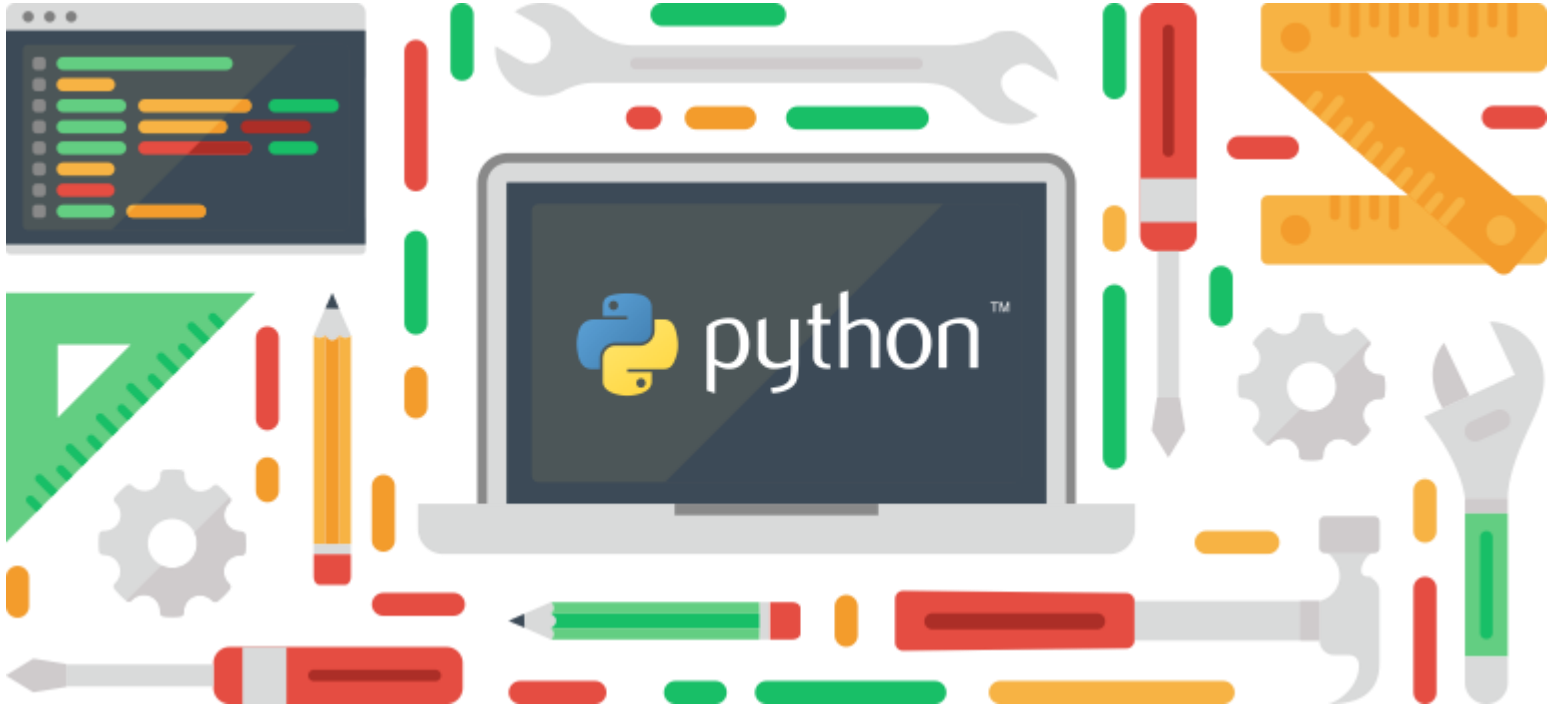
Робота з класами та екземплярами

6

Класи використовують для моделювання реальних ситуацій.

Можна розробити таку програму, щоб за певним алгоритмом змінювався стан об'єктів.

Розглянемо, як можна змінювати атрибути, пов'язані з конкретним екземпляром класу.



Розділ 3
§ 16

Робота з класами та екземплярами

Опишемо клас, що представляє автомобіль.

```
class Car():
        def _ _init_ _(self, model, mileage):
                self.mileage = mileage
                                # Пробіг автомобіля в кілометрах
                self.model = model
my_car = Car('Sens',  2000)
print(my_car.model, ' ', my_car.mileage)
```

1

2

3

*У класі **Car** визначаємо метод **_ _init_ _**, який набуває параметрів: **model**, **mileage** і зберігає в атрибутах екземплярів класу (1).*

Розділ 3
§ 16

Робота з класами та екземплярами

*Створюємо екземпляр класу **Car**, який зберігається в змінну **my_car**. При створенні об'єкта вказуємо відповідно до списку атрибутів назву моделі та пробіг (2).*

*Оператор **print** виводить у вікно консолі значення атрибутів об'єкта my_car (3): Sens 2000*

Ми вже змінювали значення атрибута, звертаючись до екземпляра. Розглянемо приклади.

*Атрибуту **mileage** об'єкта **my_car** надамо значення 120.*

```
my_car.mileage = 120
```

Робота з класами та екземплярами

До опису можна включити методи, які змінюють деякі атрибути.

Додамо до опису класу Car() метод update_mileage() для зміни значення пробігу.

```
class Car():
  ...
      def update_mileage(self, km):
                                  # Встановлення значення пробігу
          self. mileage = km
my_car = Car('Sens', 100)
my_car.update_mileage(200)
print (my_car.model, ' ', my_car.mileage)
```

Буде надруковано: Sens 200.

Робота з класами та екземплярами

Змінимо код методу update_mileage(), щоб він отримував величину пройденого шляху й додавав її до поточних показників одометра.

```
class Car():
  ...
    def up_mileage(self, km):
                                # Збільшення значення пробігу
        self.mileage = self.mileage+km
my_car = Car('Sens', 100)
km = 50
my_car.up_mileage(km)
print(my_car.mileage)
```

Буде надруковано: Sens 150.

Розділ 3
§ 16

Робота з класами та екземплярами

6

Змінюючи атрибути, пов'язані з певним екземпляром класу, можна змінювати стан об'єктів у програмі.

Розділ 3
§ 16

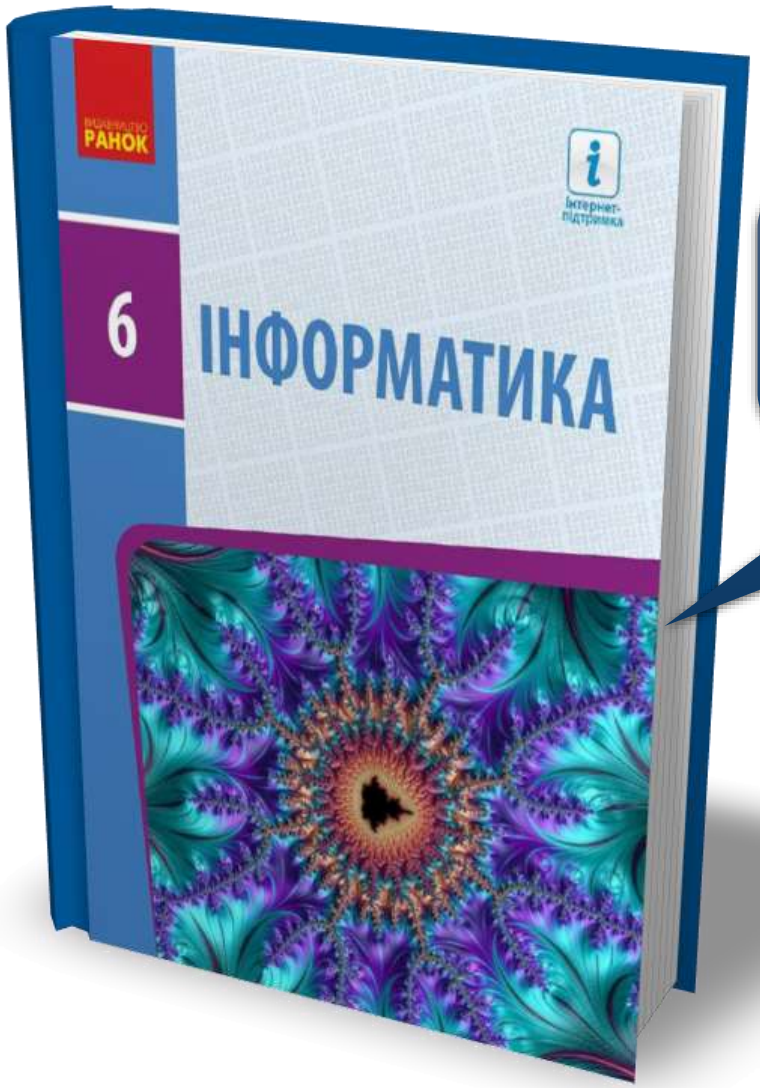
Питання для самоперевірки

6

1. *У чому різниця між властивостями й методами об’єкта?*
2. *Які властивості й методи можуть мати такі об’єкти: учень, учитель, країна, браузер?*
3. *Поясніть структуру заголовка методу* `def up_mileage(self, km)`.
4. *Як записати виклик методу* `def up_mileage(self, km)`*?*

Розділ 3
§ 16

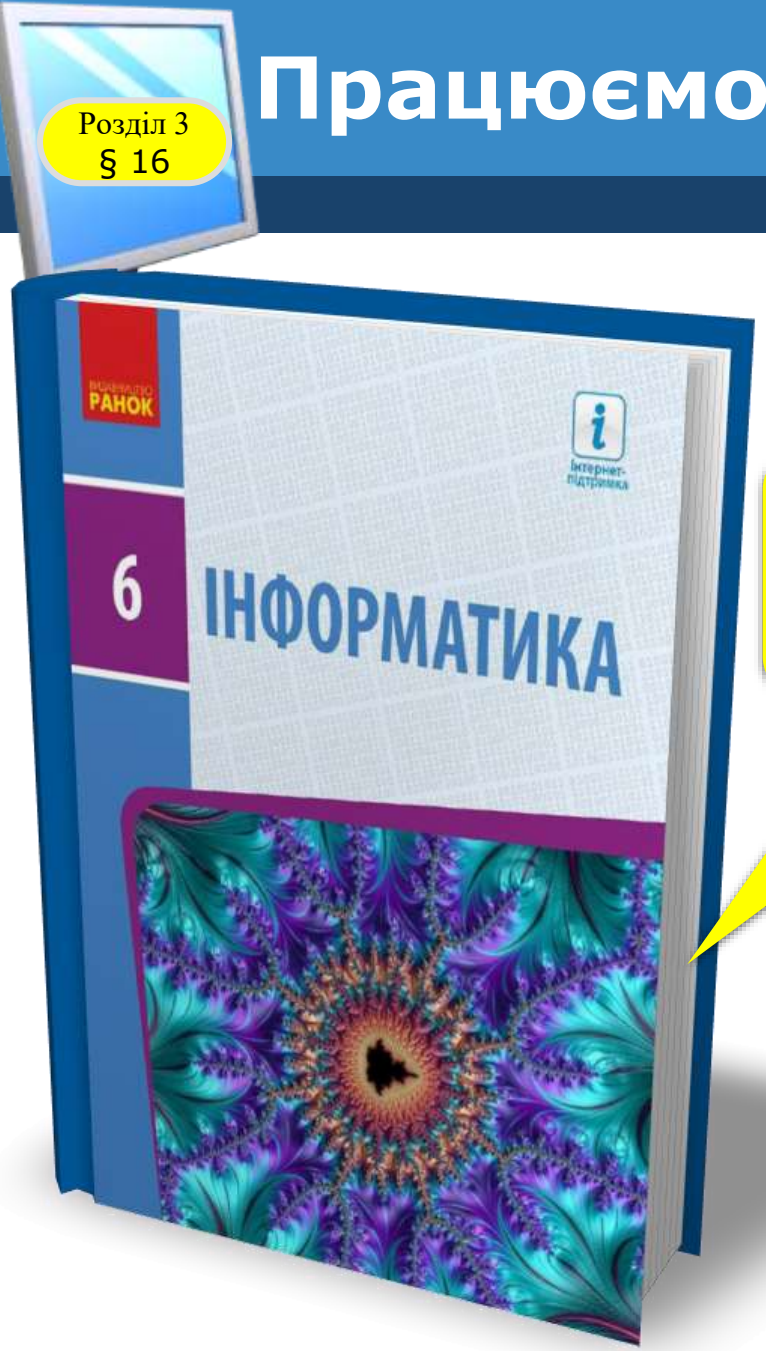
Домашнє завдання



Проаналізувати
§ 16, ст. 103-108

Розділ 3
§ 16

Працюємо за комп’ютером



Сторінка 107-108

6 ІНФОРМАТИКА

Дякую за увагу!

За навчальною програмою 2017 року

Урок 19