

ПРАКТИЧНА РОБОТА 07
Основи об'єктно-орієнтованого програмування мовою Python
План

1. Класи та об'єкти в мові Python.
2. Побудова ієрархії наслідування класів.
3. Структури даних у мові Python.

Система оцінювання

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Класи та об'єкти в мові Python (задачі 1-2)	0,5*
3.	Побудова ієрархії наслідування класів (задачі 3-6)	1,7*
4.	Структури даних у мові Python (задачі 7-9)	1,4*
5.	<i>Здача звіту</i>	0,4
	Всього	5

* – діє бонусна система

1. 0,2 бала **(Класи та об'єкти)** Класичною вправою для людей, які вчаться писати класи є моделювання особи (**клас *Person***). Ми знайомі з рисами та поведінкою людей, тому визначте клас `Person()`, в якому функція `__init__()` встановлює кілька атрибутів для особи. Тут можна розглядати ім'я, вік, місце народження та ін. інформацію, що хочеться знати про людину.

Напишіть метод, який буде представляти цю людину, наприклад, виводити привітання «Привіт, мене звати _____». Також можете доповнити клас методом, який буде збільшувати вік особи на 1. У більш складній версії цього методу потрібно зберігати дату народження, а не вік, а потім обчислювати вік як різницю між поточною датою та датою народження.

Створіть об'єкт-особу, заповніть інформацію про неї та виведіть її на екран.

2. 0,3 бала **(Класи та об'єкти)** Створіть клас `Ship`, у якому визначено метод `__init__()`, який встановлює для кожного об'єкта цього класу початкові значення координат (0; 0). Також у ньому записано метод `move_up()`, який збільшує у-координату корабля на 1.

Створіть корабель (об'єкт класу `Ship`) та виведіть на екран інформацію про об'єкт та його у-координату. Пересуньте корабель вгору та знову виведіть його у-координату.

Створіть корабельний флот з 6 кораблів, кожен з яких знаходиться в різних координатах та покажіть, що кожен корабель – це окремий об'єкт. Перемістіть кілька з них.

Доповніть клас `Ship` методом, який буде обчислювати відстань між кораблями. Виведіть відстані від першого корабля флотилії до решти кораблів.

Придумайте та введіть у клас додаткові атрибути, що описують корабель, наприклад, довжину корпусу корабля, розмір команди, назву корабля, його максимальну швидкість тощо. Виведіть розширений опис наявних кораблів.

3. 0,3 бала **(Наслідування)** Якщо Ви хочете змодельовати космічний корабель, можна написати абсолютно новий клас. Проте космічний корабель є спеціальним видом корабля, тому можна успадкувати атрибути та поведінку класу Ship, а потім доповнити наш новий клас додатковими атрибутами та методами, що стосуються космічного корабля.

У фантастиці розглядають кілька понять для космічних суден, зокрема [зореліт](#) (starship, starcraft або interstellar spacecraft) та [космічний апарат](#) (Spacecraft). Спрощено кажучи, зореліт є підкатегорією космічних апаратів. Таким чином, можна вибудувати ієрархію наслідування: Корабель – Космічний апарат – Зореліт. Реалізуйте цю ієрархію наслідування у вигляді відповідних класів. Створіть об'єкти кожного із запропонованих класів.

4. 0,6 бала **(Ієрархія наслідування)** Подумайте над тим, що кожен корабель може мати свою спеціалізацію. Знайдіть класифікацію кораблів та спробуйте перенести її в код. Зокрема, це можуть бути транспортні судна, екскурсійний лайнер, дослідницький або бойовий корабель. Побудуйте схему, що відображатиме, на Вашу думку, класифікацію кораблів, в основі якої буде лежати базове поняття «корабель». Додайте діаграму класів до звіту. Виділіть для кожного елементу схеми атрибути та методи, що будуть характерні саме для нього та його дочірніх елементів. Спробуйте перенести дану схему в код.
5. 0,2 бала **(Модулі та класи)** Класи з часом розростаються новим функціоналом, тому з ними стає важче працювати. Мова Python дозволяє зберігати Ваші класи в окремий файл – *модуль*, а потім імпортувати його до Вашої програми. Створіть окремий модуль, що міститиме вище описані класи, а потім імпортуйте його до Вашої програми, в яку перенесеться код роботи з об'єктами цих класів.
6. 0,6 бала **(Комплексні основи ООП)** Створіть новий клас Student, породжений класом Person. Визначте атрибути, які має лише студент, і не мають інші люди. Наприклад, студент відвідує навчальний заклад, має рік вступу, перелік поточних/пройдених курсів з оцінками, середній бал за кожну сесію (GPA) та ін.

Створіть об'єкт класу Student та покажіть, що наслідування виконується коректно. Задайте деяким атрибутам для студента значення, записані лише в класі Person або лише в класі Student. Виведіть значення цих атрибутів.

Доповніть архітектуру додатку класами, які описують кафедру навчання студента. Кожна кафедра має назву та перелік викладачів, які на ній працюють. Кожен з викладачів має свій набір курсів для викладання. Загалом кафедра пропонує набір курсів (потрібен клас для відповідного опису), які студент має завершити протягом свого навчання. Кожен курс мінімально має свій код, назву, кількість навчальних

кредитів (від 1 до 6), прикріплених до нього викладачів (достатньо ППБ), час початку та час завершення.

Збережіть класи Person та Student в окремому файлі (person.py), а код, що використовує ці класи, в чотирьох окремих файлах:

- 1) Тут використовуйте синтаксис «from module_name import ClassName», щоб запустити Вашу програму.
- 2) Тут використовуйте синтаксис «import module_name».
- 3) Використовуйте синтаксис «import module_name as different_local_module_name».
- 4) Використовуйте синтаксис «import *».

7. 0,4 бала **(Задача на структури даних)** Подумайте про Ваших одногрупників, які зараз знаходяться в кабінеті. Напишіть повідомлення, яке вітається з кожним з них. Збережіть їх імена та прізвища у списку та виведіть привітання кожного по цьому списку в алфавітному порядку за іменем. Також надрукуйте повідомлення, звертаючись до кожного в алфавітному порядку за прізвищем.

Зверніть увагу на дні народження одногрупників зі списку. Виведіть на екран їх імена та дні народження від наймолодшого до найстаршого. Якщо день народження когось зі списку відбудеться в наступні 30 днів від нинішньої дати, покажіть спеціальне повідомлення для цієї особи. Збережіть інформацію про людей у словнику, якщо досі цього не зробили.

Примітка. Не забувайте, що для [сортування](#) може використовуватися

- метод sort(), який переставляє елементи в самій структурі даних
- метод sorted(), який повертає відсортовану копію
- функція attrgetter(), яка генерує функцію, що бере об'єкт і повертає певний атрибут з цього об'єкта

8. 0,4 бала **(Задача на структури даних)** Інформація щодо пакету акцій зберігається в наборі атрибутів – даті придбання, ціні придбання кількості та символу на біржі. Ми можемо записати цю інформацію в кортеж для кожного пакету акцій та виконувати з цими пакетами прості операції. Нехай у нас є портфоліо:

Дата придбання	Ціна придбання	Кількість	Символ на біржі	Поточна ціна
25 Jan 2018	43.50	25	CAT	92.45
25 Jan 2018	42.80	50	DD	51.19
25 Jan 2018	42.10	75	EK	34.87
25 Jan 2018	37.58	100	GM	37.58

Ми можемо представити цю інформацію у вигляді списку кортежів

```
portfolio= [ ( "25-Jan-2018", 43.50, 25, 'CAT', 92.45 ),  
              ( "25-Jan-2018", 42.80, 50, 'DD', 51.19 ),  
              ( "25-Jan-2018", 42.10, 75, 'EK', 34.87 ),  
              ( "25-Jan-2018", 37.58, 100, 'GM', 37.58 )  
            ]
```

Напишіть функцію, яка переглядає кожен пакет акцій та множить кількість на ціну їх покупки, щоб визначити загальну ціну портфоліо на момент придбання. Реалізуйте аналогічну функцію для обчислення поточної вартості портфоліо.

9. ^{0,6 бала} *(Задача на структури даних)* Змодельуйте чергу в ресторані самообслуговування. Час обслуговування генеруватиметься випадковим чином у заданих далі діапазонах. Сумарний час замовлення їжі складається з трьох частин:
- 1) Підбору свого меню – збиранні на підносі бажаних страв. Нехай існує великий стіл зі стравами, які пропонує заклад харчування, причому кількість порцій цих страв обмежена. Вважатимемо, що людина витрачає від 2 до 5 секунд на підбір однієї страви для себе, кількість таких страв обмежена до 6. Зібравши свій піднос, клієнт стає в чергу.
 - 2) Очікування в черзі, поки ще не обслужені клієнти попереду.
 - 3) Обслуговування на касі, яке може займати від 5 до 35 секунд.

Створіть чергу з 12 клієнтів, кожен з яких має власне меню. Виведіть по кожному з них наступну інформацію:

- 1) Номер клієнта
- 2) Номер клієнта у черзі
- 3) Час, коли клієнт став у чергу
- 4) Тривалість очікування на розрахунок за страви
- 5) Тривалість обслуговування на касі
- 6) Повний час обслуговування

Наприклад, у такому форматі:

# клієнта	# у черзі	Час	Тривалість	Тривалість	Час
4	6	00:21	25 секунд	31 секунда	01:16