

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з дисципліни
«Основи програмування на Python»

для бакалаврів спеціальності 123 "Комп'ютерна інженерія"
денної форми навчання

Частина I

2023

Методичні вказівки до лабораторних робіт з дисципліни «Основи програмування на Python» для бакалаврів спеціальності 123 "Комп'ютерна інженерія", денної форми навчання. Частина I / Укл.: А.В. Тіменко, Н.А. Куликовська – Запоріжжя: НУ «Запорізька політехніка», 2023. – 16 с.

Укладач: А.В. Тіменко, старший викладач
Н.А. Куликовська, старший викладач

Рецензент: М.Ю. Тягунова, к.т.н., доцент

Відповідальний
за випуск: А.В. Тіменко, старший викладач

Затверджено:
На засіданні кафедри
«Комп'ютерні системи та мережі»
Протокол № 5
від 30 січня 2023р.

Рекомендовано до видання
НМК факультету КНТ
Протокол № 6
від 31 січня 2023 р.

ЗМІСТ

1	ЛАБОРАТОРНА РОБОТА НАЛАШТУВАННЯ РYTHON.....	4
1.1	Теоретичні відомості	4
1.2	Хід роботи	6
1.3	Зміст звіту	6
1.4	Контрольні питання	7
2	ЛАБОРАТОРНА РОБОТА. ТИПИ ДАНИХ.....	8
2.1	Теоретичні відомості	8
2.2	Хід роботи	9
2.3	Зміст звіту	10
2.4	Контрольні запитання.....	10
3	ЛАБОРАТОРНА РОБОТА. СПИСКИ І КОРТЕЖІ ..	12
3.1	Теоретичні відомості	12
3.2	Завдання до виконання	13
3.3	Зміст звіту	14
3.4	Контрольні запитання.....	14
	ЛІТЕРАТУРА.....	16

1 ЛАБОРАТОРНА РОБОТА №1

НАЛАШТУВАННЯ PYTHON

Мета роботи: отримати навички налаштування середовища розробки для мови програмування Python.

1.1 Теоретичні відомості

Версії інтерпретатора Python для різних операційних систем доступні для безкоштовного завантаження за адресою <https://www.python.org/downloads/>

Для написання програм використовують текстові редактори або інтегровані середовища розробки, які включають в себе різні інструменти для роботи з кодом: засіб для написання коду (текстовий редактор), інтерактивний інтерпретатор, відлагоджувач тощо.

У режимі інтерактивного інтерпретатора команди вводяться у термінальному вікні (консольне вікно, вікно командного рядка) одна за одною і по натисненні клавіші Enter відразу виконуються з відображенням результату виконання.

Щоб відкрити вікно терміналу:

натисніть сполучення клавіш Win+R на клавіатурі, введіть команду cmd, натисніть ОК (для користувачів Windows);

натисніть сполучення клавіш Ctrl+Alt+T (для користувачів Linux Ubuntu).

У термінальному вікні, що з'явилося, введіть команду (у випадку використання Windows):

```
python
```

або (у випадку використання Linux Ubuntu можна вказати версію Python, наприклад, 3.9):

```
python3.9
```

Якщо на екрані з'явиться запрошення >>> до введення команд, значить система виявила встановлену версію Python (рис. 1.1)

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.14393]
(c) Корпорація Майкрософт (Microsoft Corporation), 2016. Усі права захищено.

C:\Users\user>python
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

Рисунок 1.1 – Встановлена версія Python

Режим інтерактивного інтерпретатора Python у термінальному вікні Windows: запрошення до введення команд (рис. 1.2).

```

teacher@teacher-VirtualBox: ~
teacher@teacher-VirtualBox:~$ python3.9
Python 3.9.0+ (default, Oct 20 2020, 08:43:38)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

Рисунок 1.2 – Режим інтерактивного інтерпретатора Python у термінальному вікні Windows

Режим інтерактивного інтерпретатора Python у термінальному вікні Linux Ubuntu: запрошення до введення команд

Введіть в інтерактивному режимі інтерпретатора наступний рядок

```
print('Hello, Python!')
```

натисніть Enter і переконайтеся в тому, що на екрані з'явилося повідомлення Hello, Python! (рис. 1.3).

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.14393]
(c) Корпорація Майкрософт (Microsoft Corporation), 2016. Усі права захищено.

C:\Users\user>python
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, Python!')
Hello, Python!
>>>
  
```

Рисунок 1.3 – Режим інтерактивного інтерпретатора Python у термінальному вікні Linux

Режим інтерактивного інтерпретатора Python у термінальному вікні Windows: виведення текстового повідомлення

Функція print() входить у стандартну бібліотеку Python. Вона виводить інформацію, вказану в дужках, на екран або записує у файл.

Щоб вийти з режиму інтерактивного інтерпретатора Python, натисніть Ctrl+Z і Enter у Windows, Ctrl+D у Linux Ubuntu або виконайте команду `exit()` .

1.2 Хід роботи

1 Встановіть Python 3.

2 Встановіть середовище програмування.

3 Запустіть інтерактивний інтерпретатор Python 3 і використайте його як калькулятор. Наприклад, обчисліть $19 * 81$. Запишіть цей добуток і натисніть Enter, щоб побачити результат. Python повинен вивести 1539.

4 Введіть число 43 і натисніть клавішу Enter. Чи з'явилось це число в наступному рядку?

5 Введіть `print(43)` і натисніть клавішу Enter. Чи з'явилось знову це число в наступному рядку?

6 В інтерактивному інтерпретаторі Python введіть 'Python' + 3. Знайдіть інформацію в мережі Інтернет про помилку, що виникла, за її назвою.

7 Перегляньте принципи Python, ввівши в термінальну сеансі команду `import this`. Знайдіть переклад тексту.

1.3 Міст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

1.4 Контрольні питання

1. Хто є розробником мови Python? Звідки походить назва мови програмування Python?
2. Для яких цілей використовується Python?
3. Як перейти в режим інтерактивного інтерпретатора?
4. Яке розширення мають файли із програмами, написаними на мові Python?
5. Як запустити програму, що міститься у файлі, на виконання у термінальному вікні?
6. У яких випадках, під час виконання програми, може з'явитися повідомлення Traceback (most recent call last)?
7. Який символ використовується у мові Python для позначення коментарів?

2 ЛАБОРАТОРНА РОБОТА №2 ТИПИ ДАНИХ

Мета роботи: навчитися створювати програми мови програмування Python.

2.1 Теоретичні відомості

Тип даних – це множина значень і операцій на цими значеннями. В Python типи даних можна розділити на вбудовані в інтерпретатор (built-in) і невбудовані, які можна використовувати при імпортуванні відповідних модулів.

До основних вбудованих типів в Python відносяться:

1. None – невизначене значення змінної.
2. Boolean Type – логічний тип.
3. Numeric Type – числа:
 - int – ціле число;
 - float – число з плаваючою точкою (дійсне число);
 - complex – комплексне число.
4. Sequence Type – послідовності:
 - list – список;
 - tuple – кортеж;
 - range – діапазон.
5. Text Sequence Type – рядки:
 - str – рядки.
6. Binary Sequence Types – бінарні послідовності:
 - bytes – байти;
 - bytearray – масиви байт 24;
 - memoryview – спеціальні об'єкти для доступу до внутрішніх даних об'єкта через protocol buffer.
7. Set Types – множини:
 - set – множина;
 - frozenset – незмінювана множина.
8. Mapping Types – словники:
 - dict – словник.

Враховуючи неявну типізацію мови Python при оголошенні змінної, їй повинно бути надане значення (вона має бути

ініціалізована). Щоб оголосити та ініціалізувати змінну необхідно: написати її ім'я, потім поставити оператор присвоєння (знак дорівнює) і вказати значення, з яким дана змінна буде створена.

При ініціалізації змінної на рівні інтерпретатора відбувається наступне:

- створюється цілочисельний об'єкт 10 (можна представити, що в цей момент виділяється комірка пам'яті певного об'єму і в неї записується число 10);

- за оператором присвоєння = відбувається зв'язування змінної z і цілочисельного об'єкту 10 (визначається адреса отриманого об'єкту 10, яка присвоюється змінній z, тобто в результаті змінна z посилається на об'єкт 10).

Якщо, змінній за оператором присвоєння надається будь-який вираз ($a = 5 + 10$), то спочатку інтерпретатор обчислює значення виразу справа ($5 + 10$), створює об'єкт з отриманим значенням (15), а далі аналогічно до попереднього.

2.2 Хід роботи

Виконати завдання:

1. Збережіть будь-яке повідомлення у змінній і виведіть це повідомлення. Потім замініть значення змінної іншим повідомленням і виведіть нове повідомлення. Програму збережіть у файлі, ім'я якої підпорядковується стандартним правилам Python по використанню малих літер і символів підкреслення - наприклад, `simple_messages.py`.

2. Збережіть ім'я користувача у змінній і виведіть повідомлення, призначене для конкретної людини. Повідомлення повинно бути, наприклад, таким: "Hello, Sasha, would you like to learn some Python today?".

3. Знайдіть відому цитату, яка вам подобається. Збережіть ім'я автора вислову у змінній `famous_person`. Складіть повідомлення і збережіть його у новій змінній з ім'ям `message`. Виведіть своє повідомлення. Результат повинен виглядати приблизно так (включаючи лапки): "Albert Einstein once said, "A person who never made a mistake never tried anything new."".

4. Збережіть ім'я користувача у змінній і додайте на початку і у кінці імені кілька пропусків. Простежте за тим, щоб кожна керуюча послідовність (`\t` і `\n`) зустрічалася принаймні один раз. Виведіть ім'я,

щоб було видно пропуски на початку і у кінці рядка. Потім виведіть його знову з використанням кожної з функцій видалення пропусків: `lstrip()`, `rstrip()` і `strip()`.

5. Використайте функцію `print()` для виведення повної домашньої адреси. У першому рядку виведіть власне ім'я та прізвище. У кожному наступному рядку виведіть окремі елементи адреси (країна, індекс, назва населеного пункту, вулиця, номер будинку тощо).

6. Виконайте переведення одиниць вимірювання відстаней. Значення відстані вказано у метрах. У кожному новому рядку програма виводить значення відстані, представлене у: дюймах, футах, милях, ярдах тощо. Числові дані на екрані мають бути у відформатованому вигляді: два знаки після десяткової крапки. Використайте функцію `format()`. Потрібні значення одиниць вимірювання знайдіть у мережі Інтернет.

7. Обчисліть тривалість якоїсь події. Припустимо, учнівські канікули тривали кілька днів. На екран треба вивести у відформатованому вигляді (вирівнювання за лівим краєм, ширина поля: 10 знаків) загальну тривалість цієї події у годинах, хвилинах, секундах.

2.3 Зміст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

2.4 Контрольні запитання

1. Які з наведених нижче елементів є операторами, а які - значеннями?

```
*
'Hello'
-45.7
-
/
5
```

+

2. Що із наведеного нижче є змінною, а що - рядком?

```
letter
'letter'
```

3. Що буде містити змінна `result` після виконання наступних інструкцій?

```
result = 20
result += 5
```

4. Якими будуть результати (у місці ...) обчислення двох наступних виразів?

```
>>> 'snowflake' + 'snowflakesnowflake'
...
>>> 'snowflake' * 3
...
```

5. Чому виникає помилка при спробі обчислення поданого нижче виразу? Як виправити цю помилку?

```
print('I got' + 12 + 'points')
```

6. Які будуть результати обчислення таких виразів:

```
"I'd like to travel to Shanghai"[0:21]
"I'd like to travel to Shanghai"[:21]
"I'd like to travel to Shanghai"[22:]
"I'd like to travel to Shanghai"[12:18]
"I'd like to travel to Shanghai".upper()
"I'd like to travel to Shanghai".upper().lower()
"I'd like to travel to Shanghai".split()
'>>'.join("I'd like to travel to Shanghai".split())
```

3 ЛАБОРАТОРНА РОБОТА №3 СПИСКИ І КОРТЕЖІ

Мета роботи: отримати навички програмування на мові Python з використанням списків і кортежів.

3.1 Теоретичні відомості

Список — впорядкована змінна послідовність об'єктів довільного типу. У мові Python списки представлені типом `list`.

Кортеж - незмінний аналог списків, якому відповідає тип даних `tuple`.

Змінна послідовність — послідовність, в якій після її створення можлива зміна кількості елементів або привласнення об'єктів.

Лістинг 3.1 – Створення списку

```
>>> s = []
>>> type(s)
<class 'list'>

>>> s = [1, 2, 3]
>>> type(s)
<class 'list'>

>>> s = list('iterable_obj')
>>> type(s)
<class 'list'>
>>> s
['i', 't', 'e', 'r', 'a', 'b', 'l', 'e', '_', 'o', 'b', 'j']
```

Лістинг 3.1 - Створення кортежу

```
>>> s = ()
>>> type(s)
<class 'tuple'>

>>> s = 1,
>>> type(s)
<class 'tuple'>

>>> s = tuple('iterable_obj')
>>> type(s)
<class 'tuple'>
```

```
>>> s
('i', 't', 'e', 'r', 'a', 'b', 'l', 'e', '_', 'o', 'b', 'j')
```

Використовуючи конструктори `list()` і `tuple()`, можна виконувати перетворення послідовностей у перелік чи кортеж.

3.2 Завдання до виконання

Виконайте в інтерактивному інтерпретаторі такі завдання:

1. Збережіть імена кількох своїх друзів у списку з ім'ям `names`. Виведіть ім'я кожного друга, звернувшись до кожного елементу списку (по одному разу).

2. Створіть список з типами транспортних засобів. Використайте список для виведення твердження, на зразок: «Я хотів би купити велосипед.»

3. Створіть список `years_list`, що містить рік, в який ви народилися, і кожен наступний рік аж до вашого п'ятого дня народження. Наприклад, якщо ви народилися в 1995 році, список буде виглядати так: `years_list = [1995, 1996, 1997, 1998, 1999, 2000]`. Виведіть на екран, у якому із років, що міститься у списку `years_list`, вам виповнилося 3 роки? Пам'ятайте, у перший рік вам було 0 років. Додайте у кінець списку ще один рік і виведіть список на екран. У якому із років, перерахованих у списку `years_list`, вам було найбільше років?

4. Створіть список `things`, що містить три елементи: `'wallet'`, `'mirror'`, `'umbrella'`. Виведіть на екран той елемент у списку `things`, який має відношення до дощу, написавши його з великої літери, а потім виведіть список. Переведіть «дощовий» елемент списку `things` у верхній регістр цілком і виведіть список. Видаліть річ, яка захищає від дощу, зі списку `things`, а потім виведіть список на екран.

5. Створіть список, який називається `languages` і містить елементи `'Georgian'`, `'Estonian'` і `'Ukrainian'`. Напишіть останній елемент списку `languages` з малої літери, потім «переверніть» його і напишіть з великої літери.

6. Збережіть кілька понять, пов'язаних з комп'ютерами та Інтернетом, у кортежі і у списку. Назвіть відповідно їх `hardware` (кортеж) і `software` (список). Виведіть усі назви по черзі.

Спробуйте замінити один з елементів у списку і у кортежі, і зробіть висновок щодо можливості зміни елементів для цих двох типів.

Напишіть програми у середовищі програмування для розв'язування таких завдань:

1. Збережіть назви мов світу (Ukrainian, French, Bulgarian, Norwegian, Latvian або інші) у списку. Простежте за тим, щоб елементи у списку не зберігались в алфавітному порядку. Застосуйте функції `sorted()`, `reverse()`, `sort()` до списку. Виведіть список на екран до і після використання кожної із функцій.

2. На вхід програми подається один рядок з цілими числами. Числа розділені пропусками. Необхідно вивести суму цих чисел. Наприклад, якщо був введений рядок чисел 2 -1 9 6, то результатом роботи програми буде їх сума 16.

3. Дано список з такими елементами: `cities = ['Budapest', 'Rome', 'Istanbul', 'Sydney', 'Kyiv', 'Hong Kong']`. Сформуйте з елементів списку повідомлення, у якому перед останнім елементом буде вставлено слово `and`. Наприклад, у нашому випадку, повідомлення буде таким: `Budapest, Rome, Istanbul, Sydney, Kyiv and Hong Kong`. Програма має працювати з будь-якими списками, довжина яких є 6.

3.3 Зміст звіту

Звіт повинен мати такі складові:

- титульний лист;
- мета роботи;
- результат виконання завдань;
- відповіді на контрольні питання.

3.4 Контрольні запитання

1. Що означають дужки `[]`?
2. У змінній `letters` міститься список `['a', 'b', 'c', 'd']`. Яке значення виразу `letters[int('3' * 2) // 11]`?
3. У змінній `letters` міститься список `['a', 'b', 'c', 'd']`. Яке значення виразу `letters[-1]`?

4. У змінній `letters` міститься список `['a', 'b', 'c', 'd']`. Яке значення виразу `letters[:2]`?
5. У змінній `letters` міститься список `[2, 4, 6, 8, 10]`. Як присвоїти значення `'zero'` в якості третього елемента даного списку?
6. У змінній `metering` міститься список `[3.14, 'inch', 2.54, 'inch', True]`. Яке значення виразу `metering.index('inch')`?
7. Як буде виглядати список, який зберігається у змінній `metering = [3.14, 'inch', 2.54, 'inch', True]`, після виконання команди `metering.append(99)`?
8. Як буде виглядати список, який зберігається у змінній `metering = [3.14, 'inch', 2.54, 'inch', True]`, після виконання команди `metering.remove('inch')`?
9. В чому різниця між функціями `append()` та `insert()`, передбаченими для списків?
10. Назвіть способи видалення значень із списку.
11. Назвіть декілька спільних (відмінних) ознак для рядків і списків.
12. Змінні, які «зберігають» список, насправді, не зберігають в собі безпосередньо сам список. Що тоді вони містять?
13. Чим кортежі відрізняються від списків?
14. Як створити кортеж, який міститиме єдине значення у вигляді цілого числа 24?
15. Як перетворити список в кортеж?

ЛІТЕРАТУРА

1. Python В. К. Python Cookbook / Brian K. Python, David Beazley, Brian K. Jones. – [S. 1.] : O'Reilly Media, Incorporated, 2012. - 704 p.
2. Мова програмування Python для інженерів і науковців: Навчальний посібник. – Ivano-Frankivsk, Ukraine : ІФНТУНГ, 2019. – 275 с.
3. Barry P. Head First Python / Paul Barry. – [S. 1.] : O'Reilly Media, Incorporated, 2015. – 624 с.