

Лабораторна робота №8

Тема: Вбудовані функції. Робота модулів.

Мета роботи: закріпити та розширити навички роботи з вбудованими функціями Python, навчитися створювати власні модулі.

1. Теоретичні відомості:

Вбудовані функції

В минулому ми вже користувались вбудованими функціями, наприклад `print()`, `input()`, `int()`, `float()`, `str()` та інші. Програміст не бачить їх код та не знає як вони працюють, бо їх коди сховані в «недрах» мови програмування. Достатньо лише знати *інтерфейс функції*.

Розглянемо інтерфейси деяких вбудованих функцій:

- Функція `ord()` (Order) приймає в якості аргумента один символ в лапках та повертає його номер у таблиці Unicode.

```
>>> ord('r')
114
>>> ord('9')
57
>>> ord('§')
37
>>> ord('y')
1091
```

- Функція `chr()` (Character) приймає в якості аргумента ціле число та передає символ з таблиці Unicode за цим номером.

```
>>> chr(10)
'\n'
>>> chr(114)
'r'
>>> chr(37)
'§'
```

- Функція `len()`, окрім масивів, може також приймати строки. Результатом буде кількість символів у данному рядку.

```
>>> len('abcde')
5
>>> len('qw_er ty')
8
>>> a = 'string'
>>> len(a)
6
```

Щодо математичних функцій, не завжди доцільно викликати модуль `math`, так як існують вбудовані аналоги функцій.

- Функція *abs()* приймає число та повертає його абсолютне значення, тобто його значення по модулю.

```
>>> abs(-17.51)
17.51
>>> abs(17.52)
17.52
```

- Функція *round()* округляє отримане число до вказаного знаку. Першим аргументом функція приймає число, другим – знак після дробу, до якого буде округлення. Якщо аргумент буде всього один, то число буде округлятися до цілої частини. Також другий аргумент може бути від’ємним – тоді число буде округлятися до десятків/сотень/тисяч/т.д.

```
>>> round(12.358936, 2)
12.36
>>> round(-17.94)
-18
>>> round(15743, -2)
15700
```

- Функція *divmod()* приймає два числа та повертає кортеж з двох елементів – результату ділення націло першого аргумента на другий та залишок від їх ділення.

```
>>> divmod(27, 5)
(5, 2)
```

Аналогом цієї функції є два оператори: `//` (ділення націло) та `%` (залишок від ділення націло).

```
>>> 27//5
5
>>> 27%5
2
```

- Функції *max()* та *min()* можуть працювати з нескінченною кількістю аргументів. Перша повертає серед них найбільший, друга – найменший. При цьому обидві ці функції можуть приймати строки як аргумент. У такому випадку сортування відбуватиметься не за значенням, а за алфавітом. Також в функції можливо передавати переліки елементів. У цьому випадку результатом є відповідний елемент масиву.

```
>>> max(4, 2, 12)
12
>>> min(4, 2, 12)
2
```

```
>>> max('abc', 'bcdef', 'f')
'f'
>>> min('100', '1', '01')
'01'

>>> s = [1,2,3,4,5]
>>> max(s)
5
>>> min(s)
1
```

- Функція *sum()* може отримувати тільки перелічок елементів і тільки з числами. Повертає функція суму всіх елементів переданого масиву.

```
>>> s = [1,2,3,4,5]
>>> p = (0, 12, 23, 20)
>>> sum(s)
15
>>> sum(p)
55
```

Створення власної бібліотеки

Для створення власного модулю робимо файл з розширенням *.py*, який буде складатись тільки з функцій. Назва цього файлу – це назва, за якою її можна імпортувати. а назви функцій, записаних у цьому файлі – це назви функцій модулю. Після того, як наша бібліотека буде цілком готова, зберігаємо її в ту саму папку, де знаходиться інтерпретатор мови Python (там же знаходяться всі інші бібліотеки) або у папку проекту (в цьому випадку цим модулем можна користуватися лише в рамках директорії проекту).

Власні бібліотеки можна імпортувати до файлу так само, як і вбудовані додаткові бібліотеки.

2. Хід роботи

- 1) Виконати завдання на мові Python згідно з варіантом. Написати програму та перевірити її виконання.
- 2) Скласти звіт, який має мати:
 - найменування та мету роботи
 - завдання за варіантом
 - код програми
 - результат виконання програми

3. Завдання:

Варіант 1

Напишіть програму, яка буде циклічно запитувати у користувача символ. Програма повинна:

- 1) Повертати елемент за номером в таблиці Unicode, якщо введене число;
- 2) Повертати номер в таблиці Unicode, якщо введений символ;
- 3) Завершувати програму, якщо введений 0;
- 4) Висвічувати помилку, якщо введена неприпустима форма.

Варіант 2

Напишіть програму, яка буде запитувати у користувача п'ять чисел та розміщувати їх у одному масиві. Запитувати елемент знов, якщо введено не число. Програма повинна виводити мінімальне та максимальне число. Не використовувати вбудовані функції `min()` та `max()`.

Варіант 3

Напишіть програму, яка буде запитувати число і строку. Порахувати кількість символів у строці. Якщо кількість символів менша за введене число, то доставити до строки зірочки, щоб символів було кількістю з введене число. Нову строку вивести.

4. Контрольні питання:

- 1) Як завантажити сторонній модуль на комп'ютер?
- 2) Пояснити інтерфейси 3 вбудованих функцій на вибір.
- 3) Як використовувати власноруч створену бібліотеку?