

## 17. Декоратори

**T17.1** Побудувати декоратор, який коригує результат функції, що повертає число, так, щоб цей результат завжди був більше 0. Виконати перевірку роботи декоратора для деякої функції  $f$ .

**T17.2** Побудувати декоратор, який коригує результат функції, що повертає число, так, щоб цей результат  $x$  був у межах заданих границь  $a, b$  ( $a \leq x \leq b$ ). Виконати перевірку роботи декоратора для деякої функції  $f$ .

**T17.3** Побудувати декоратор, який перевіряє, чи дорівнює кількість позиційних параметрів кількості ключових параметрів функції, що декорується. Якщо не дорівнює, то ініціює виключення. За допомогою декоратора розв'язати задачу: скласти підпрограму зі змінною кількістю параметрів для обчислення функції

$$f(x_1, \dots, x_n, y_1, \dots, y_n) = (x_1 + \frac{1}{y_1}) * \dots * (x_n + \frac{1}{y_n});$$

Оформивши  $x_i$  як позиційні, а  $y_i$  – як ключові параметри.

**T17.4** Побудувати декоратор, який перевіряє, чи є всі параметри декорованої функції рядками. Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для функції, яка за декількома рядками повертає список рядків без повторів.

**T17.5** Побудувати декоратор, який перевіряє, чи належать усі параметри декорованої функції заданому типу (ім'я цього типу є параметром декоратора). Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для функції, яка обчислює середнє значення декількох числових змінних.

**T17.6** Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки позиційні параметри. Якщо ні, то ініціює виключення. За допомогою декоратора розв'язати задачу: скласти підпрограму зі змінною кількістю параметрів для обчислення функції

$$f(x_1, \dots, x_n) = \begin{cases} 1, & \text{якщо } \max_i x_i > \sum_{i=1}^n x_i, \\ \sum_{x_i > 0} x_i, & \text{в інших випадках;} \end{cases}$$

**T17.7** Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки ключові параметри. Якщо ні, то ініціює виключення. Виконати перевірку роботи декоратора для деякої функції  $f$ .

**T17.8** Побудувати декоратор, який перевіряє, чи містить функція, що декорується, тільки ключові параметри з ключами-рядками та значеннями-цілими числами. Якщо ні, то ініціює виключення. З використанням декоратора обчислити слово, що входить у рядок найчастіше (побудувати функцію, яка за декількома ключовими параметрами <слово>:<кількість входжень> повертає слово, яке має найбільшу кількість входжень).

**T17.9** Побудувати декоратор, який виконує збереження результатів виклику деякої функції, що залежить від одного натурального аргументу у списку (кешування результатів). Параметром декоратора буде найбільше можливе значення аргументу функції. Якщо виклик функції здійснюється для аргументу, для якого у списку вже є результат, він повертається із списку. Інакше здійснюється виклик функції, а її результат додається до списку. Використати цей декоратор для рекурсивної функції, яка обчислює  $n$ -те число Фібоначчі.

Вказівка: описати список з  $n$  елементів як нелокальну змінну (`nonlocal`) у функції-декораторі. Спочатку встановити значення всіх елементів списку у `None`.

**T17.10** Побудувати декоратор, який виконує збереження результатів виклику деякої функції, що залежить від декількох позиційних аргументів у словнику (кешування результатів). Ключами будуть кортежі із значень аргументів, а значеннями – значення функції. Якщо виклик функції здійснюється для аргументів, для яких у словнику вже є результат, він повертається із словника. Інакше здійснюється виклик функції, а її результат додається до словника. Використати цей декоратор для функції, яка перевіряє, чи є задане натуральне число простим та визначити щільність простих чисел на проміжку від 1 до  $n$  методом Монте-Карло. Метод Монте-Карло полягає у проведенні  $m$  випадкових випробувань ( $m \gg n$ ). Результати кожного випробування запам'ятовуються, а по закінченні доля випробувань, що мають істинне значення від всіх випробувань дає наближене значення ймовірності.

Вказівка: описати словник як нелокальну змінну (`nonlocal`) у функції-декораторі.

**T17.11** Побудувати декоратор, який здійснює трасування рекурсивної функції, показуючи номер рекурсивного виклику (глибину вкладень викликів) та значення параметрів перед викликом функції, а також номер рекурсивного

виклику та результат після виклику функції. З використанням декоратора виконати трасування рекурсивних функцій для обчислення факторіалу та чисел Фібоначчі.

Вказівка: описати глибину вкладення викликів як нелокальну змінну (nonlocal) у функції-декораторі.

**T17.12** Нехай результатом функції  $f$  є матриця у вигляді списку списків. Побудувати декоратор, який перевіряє, чи є ця матриця розрідженою. Якщо так, то замість списку списків повертає словник, що містить тільки ненульові елементи матриці. Якщо ні, то повертає результат функції. Будемо вважати, що матриця є розрідженою, якщо кількість ненульових елементів менше 0.1 кількості всіх елементів. Перевірити роботу цього декоратора для функції додавання двох матриць.

**T17.13** Нехай аргументами функції  $f$  є дві матриці. Функція сприймає матриці у вигляді списку списків. Кожним з двох фактичних параметрів може бути матриця у вигляді списку списків або у вигляді словника, що містить ненульові елементи матриці (для розрідженої матриці). Побудувати декоратор, який приводить обидва параметри до вигляду матриці як списку списків. Перевірити роботу цього декоратора для функції додавання двох матриць.

**T17.14** Нехай результатом функції  $f$  є список деяких елементів. Побудувати декоратор, який модифікує цей список так, щоб він не містив повторів. Перевірити роботу декоратора для функції, яка повертає список слів, що містяться у текстовому файлі.

**T17.15** Нехай функція  $f$  має декілька однотипних аргументів. Побудувати декоратор, який перемішує аргументи функції та подає їх на вхід функції у випадковому порядку. Перевірити роботу цього декоратора для функції побудови бінарного дерева пошуку (див. тему «Рекурсивні структури даних»).

**T17.16** Нехай функція  $f$  не повертає жодного результату, а друкує деякий рядок. Побудувати декоратор, який перед рядком показує його номер (номер виклику функції, що декорується). Перевірити роботу цього декоратора для функції виведення всіх рядків текстового файлу.