

Лабораторная работа №1

Функциональное программирование на Python

Цель работы

Изучить основные принципы функционального программирования и научиться применять функциональные конструкции языка Python для решения различных задач.

Для каждого задания создайте отдельную программу с комментариями.

Используйте только функциональные конструкции (лямбда-функции, `map()`, `filter()`, `reduce()` и т.д.).

Предоставьте примеры работы программы с выводом результатов на экран.

Вариант 1

Реализуйте функцию для вычисления n -го числа Фибоначчи с использованием рекурсии и лямбда-функции.

Функция: `fib_recursive(n)` — возвращает n -ое число Фибоначчи.

Вариант 2

Реализуйте функцию для вычисления факториала числа с использованием функции `reduce()`.

Функция: `factorial_reduce(n)` — возвращает факториал числа n .

Вариант 3

Реализуйте функцию для фильтрации списка и оставления только положительных чисел.

Функция: `filter_positive(lst)` — возвращает новый список с положительными числами.

Вариант 4

Реализуйте функцию для вычисления суммы квадратов всех элементов списка с использованием `map()`.

Функция: `sum_of_squares(lst)` — возвращает сумму квадратов элементов списка.

Вариант 5

Реализуйте функцию, которая находит все четные числа в списке с помощью `filter()` и лямбда-функции.

Функция: `filter_even(lst)` — возвращает список четных чисел.

Вариант 6

Реализуйте функцию для поиска максимального элемента в списке с помощью функции `reduce()`.

Функция: `find_max(lst)` — возвращает максимальный элемент списка.

Вариант 7

Реализуйте функцию для подсчета количества уникальных элементов в списке.

Функция: `count_unique(lst)` — возвращает количество уникальных элементов списка.

Вариант 8

Реализуйте функцию для перемножения всех элементов списка с использованием `reduce()`.

Функция: `product_of_elements(lst)` — возвращает произведение всех элементов списка.

Вариант 9

Реализуйте функцию для нахождения среднего арифметического всех элементов списка.

Функция: `average(lst)` — возвращает среднее арифметическое элементов списка.

Вариант 10

Реализуйте функцию, которая возвращает элементы списка, делящиеся на 3, с использованием `filter()` и лямбда-функции.

Функция: `filter_divisible_by_three(lst)` — возвращает список элементов, кратных 3.

Вариант 11

Реализуйте функцию для сортировки списка строк по длине строки с использованием `sorted()` и лямбда-функции.

Функция: `sort_by_length(lst)` — возвращает отсортированный список строк.

Вариант 12

Реализуйте функцию для преобразования списка чисел в список их кубов с использованием `map()`.

Функция: `cubes(lst)` — возвращает список кубов элементов.

Вариант 13

Реализуйте функцию для вычисления наибольшего общего делителя (НОД) двух чисел с использованием рекурсии.

Функция: `gcd(a, b)` — возвращает НОД чисел `a` и `b`.

Вариант 14

Реализуйте функцию для нахождения пересечения двух списков с использованием `filter()` и лямбда-функции.

Функция: `list_intersection(lst1, lst2)` — возвращает список пересечения.

Вариант 15

Реализуйте функцию для нахождения объединения двух списков с использованием `set()` и функциональных конструкций.

Функция: `list_union(lst1, lst2)` — возвращает список объединения.

Вариант 16

Реализуйте функцию для поиска разницы между двумя списками с использованием `filter()` и лямбда-функции.

Функция: `list_difference(lst1, lst2)` — возвращает список элементов, которые есть в первом списке, но отсутствуют во втором.

Вариант 17

Реализуйте функцию для перевода списка строк в верхний регистр с использованием `map()`.

Функция: `to_uppercase(lst)` — возвращает список строк в верхнем регистре.

Вариант 18

Реализуйте функцию для нахождения минимального элемента в списке с использованием `reduce()`.

Функция: `find_min(lst)` — возвращает минимальный элемент списка.

Вариант 19

Реализуйте функцию для вычисления суммы чисел в списке, которые делятся на 5.

Функция: `sum_divisible_by_five(lst)` — возвращает сумму элементов, кратных 5.

Вариант 20

Реализуйте функцию для разворота списка строк с использованием `map()` и лямбда-функции.

Функция: `reverse_strings(lst)` — возвращает список строк в обратном порядке.

Вариант 21

Реализуйте функцию для подсчета количества четных чисел в списке с использованием `filter()`.

Функция: `count_even(lst)` — возвращает количество четных чисел.

Вариант 22

Реализуйте функцию для нахождения всех простых чисел в списке.

Функция: `find_primes(lst)` — возвращает список простых чисел.

Вариант 23

Реализуйте функцию для нахождения элементов списка, которые являются палиндромами, с использованием `filter()`.

Функция: `find_palindromes(lst)` — возвращает список палиндромов.

Вариант 24

Реализуйте функцию для подсчета длины каждого элемента списка строк с использованием `map()`.

Функция: `length_of_strings(lst)` — возвращает список длин строк.

Вариант 25

Реализуйте функцию для вычисления суммы чисел Фибоначчи до n-го числа включительно с использованием рекурсии.

Функция: `sum_fibonacci(n)` — возвращает сумму чисел Фибоначчи до n.

Вопросы для самопроверки:

Что такое функциональное программирование и чем оно отличается от императивного?

Как используются функции высшего порядка в Python?

Какие преимущества дает использование функциональных конструкций?