Лабораторная Работа 1: Основы Функционального Программирования в Python

Цель:
Изучение основ функционального программирования на языке Python, включая понимание и применение ключевых концепций, таких как чистые функции, неизменяемость данных, лямбда-функции, а также функций `map`, `filter` и `reduce`.

Задачи:
1. Ознакомление с основными принципами функционального программирования.
2. Разработка навыков написания чистых функций и использования функций высшего порядка.
3. Понимание и применение лямбда-функций, а также встроенных функций `map`, `filter` и `reduce`.
4. Разработка навыков самостоятельного решения задач с использованием функционального подхода.
5. Подготовка к защите лабораторной работы, включая объяснение написанного кода и ответ на теоретические вопросы.

Индивидуальные задачи:
Каждому студенту предоставляется уникальная задача в соответствии с его номером в списке группы (смотреть в SSO). Задачи охватывают различные аспекты функционального программирования в Python.

1. Сумма Четных Квадратов
   - Отфильтровать четные числа, возвести их в квадрат и найти сумму.
2. Преобразование Температур
   - Преобразовать список температур из Цельсия в Фаренгейт, отфильтровать все значения ниже точки замерзания воды и найти их среднее значение.
3. Наибольший Общий Делитель (НОД)
   - Найти НОД для списка пар чисел.
4. Форматирование Строк
   - Преобразовать список строк в заглавный регистр, отфильтровать строки длиннее 5 символов и объединить их в одну строку.
5. Сумма Нечетных Индексов
   - Возвести в квадрат числа на нечетных позициях в списке и найти их сумму.
6. Разделение и Суммирование
   - Разделить каждое число в списке на его порядковый номер, отфильтровать полученные значения больше 1 и найти их сумму.
7. Количество Уникальных Слов
   - Подсчитать количество уникальных слов в предложении.
8. Произведение Положительных Чисел
   - Отфильтровать положительные числа и найти их произведение.
9. Минимальная Длина Строки
   - Найти строку с минимальной длиной в списке строк.
10. Сортировка и Выборка
    - Отсортировать список чисел по возрастанию, отфильтровать числа больше 50 и найти их медиану.
11. Обратное Преобразование Чисел
    - Преобразовать каждое число в списке в его обратное значение (1/x), отфильтровать значения меньше 1 и найти их среднее арифметическое.

12. Поиск Самой Длинной Строки
- Отфильтровать строки длиной более 10 символов и найти самую длинную из них.
13. Сумма Корней
- Найти квадратный корень каждого числа в списке, отфильтровать те, что больше 2, и найти их сумму.
14. Частота Символов
- Подсчитать частоту встречаемости каждого символа в строке.
15. Среднее Гармоническое
- Найти среднее гармоническое для списка чисел.

Критерии Оценки:
- Написание кода для решения индивидуальной задачи: 1 балл
- Объяснение и понимание написанного кода во время защиты: 2 балла
- Ответ на один из теоретических вопросов, выбранный преподавателем: 1 балл

Вопросы для Подготовки:
1. Что такое функциональное программирование и как оно отличается от императивного стиля программирования?
- Цель: Проверка понимания основных принципов функционального программирования.
2. Каковы преимущества использования функционального подхода в программировании?
- Цель: Оценка понимания студентом преимуществ функционального стиля, таких как более легкая отладка и тестирование.
3. Что такое чистая функция? Приведите пример.
- Цель: Убедиться, что студент понимает концепцию чистых функций, которые не производят побочных эффектов и всегда возвращают одинаковый результат для одних и тех же входных данных.
4. Что такое лямбда-функции в Python и как они используются?
- Цель: Проверить знание студентом лямбда-функций и их применение.
5. Как работают функции `map`, `filter` и `reduce`? Приведите примеры их использования.
- Цель: Понимание студентом этих важных функций высшего порядка и способности применять их на практике.
6. Как можно реализовать рекурсию в Python и какие есть ограничения?
- Цель: Оценить понимание рекурсии и связанных с ней ограничений, таких как максимальная глубина рекурсии.
7. Какие вы знаете способы оптимизации рекурсивных функций в Python?
- Цель: Проверка знаний студентов о техниках, таких как мемоизация, для оптимизации рекурсивных вызовов.
8. Как функциональное программирование может улучшить читаемость и поддерживаемость кода?
- Цель: Оценка понимания студентом влияния функционального программирования на общую структуру и качество кода.
9. В чем разница между изменяемыми и неизменяемыми типами данных в Python? Приведите примеры.
- Цель: Проверка понимания различий между изменяемыми и неизменяемыми типами данных и их влиянием на функциональное программирование.

10. Могут ли функциональные программы быть эффективными с точки зрения производительности? Объясните свой ответ.
   - Цель: Убедиться, что студент осознает потенциальные проблемы с производительностью и как их можно решить в функциональном программировании.

Пример для решения
Написание простых функций для решения математических задач, изучение и применение лямбда-функций, а также использование функций `map`, `filter`, и `reduce`.

Задача: "Анализ и Обработка Списка Чисел"

Описание задачи:

1. Дан список целых чисел.
2. Необходимо написать функцию, которая выполняет следующие действия:
   - Использует `filter` и лямбда-функцию для отбора всех положительных чисел из списка.
   - Применяет `map` с лямбда-функцией для возведения отфильтрованных чисел в квадрат.
   - Использует `reduce` для нахождения суммы полученных квадратов.

Пример:
Для списка `[-1, 2, -3, 4, 5]`:
   - Отфильтрованный список положительных чисел: `[2, 4, 5]`.
   - Квадраты этих чисел: `[4, 16, 25]`.
   - Сумма квадратов: `4 + 16 + 25 = 45`.

Решение:

```
from functools import reduce

def process_numbers(nums):
    # Фильтрация положительных чисел
    positive_nums = filter(lambda x: x > 0, nums)

    # Возведение в квадрат
    squared_nums = map(lambda x: x2, positive_nums)

    # Суммирование квадратов
    sum_of_squares = reduce(lambda x, y: x + y, squared_nums)

    return sum_of_squares

# Пример использования функции
nums = [-1, 2, -3, 4, 5]
result = process_numbers(nums)
print(result)  # Выведет 45
```

Эта задача обеспечивает комплексное понимание и применение основных концепций функционального программирования в Python, требуя от студентов использовать лямбда-функции, а также функции `map`, `filter`, и `reduce`.