

Кафедра компьютерной инженерии и моделирования

Порозов Кирилл Сергеевич

отчет по практической работе №7  
по дисциплине «ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ»

Направление подготовки:  
09.03.04 "Программная инженерия"



Оценка -

Симферополь, 2022

## **Практическая работа №7. Тема: "Функциональное программирование"**

**Цель работы:** Изучить парадигму функционального программирования, преимущества и недостатки парадигмы, научиться создавать простейшие приложения на языке высокого уровня, реализующие функциональный подход к созданию программ.

**Описание ключевых понятий:**

(при необходимости)

**Перед выполнением практической работы изучена следующая литература:**

1. Презентация лектора курса: «Функциональное программирование»
2. Прослушана видеолекция и посещены практические занятия Милюкова Виктора Васильевича.
3. Прочитаны материалы статей: [Функциональное программирование — Википедия \(wikipedia.org\)](https://ru.wikipedia.org/wiki/Функциональное_программирование) ; [Функциональное программирование для всех / Хабр \(habr.com\)](https://habr.com/ru/search/?q=функциональное%20программирование) ; [Функциональное программирование на Python | by Maria Hladka | NOP::Nuances of Programming | Medium](https://medium.com/@mhladka/функциональное-программирование-на-python-by-maria-hladka-nop-nuances-of-programming-1-medium) ; [Кодим на Python по-функциональному: Познаем силу функциональной парадигмы программирования — Хакер \(xakep.ru\)](https://xakep.ru/ru/articles/python-funktsionalnoe-programmirovaniye/)

**Написана программа на языке Python, реализующая алгоритм приведения вложенных списков в один, соблюдающая все ограничения функционального программирования:**

Фото кода:

```

"""Программа принимает на вход список, состоящий из других списков, и возвращает обычный список,
в котором присутствуют все элементы из вложенных списков.
Эта операция производится при помощи рекурсии."""
def flatten(entered_list: list):
    if entered_list == []:
        return entered_list
    if isinstance(entered_list[0], list):
        return(flatten(entered_list[0]) + flatten(entered_list[1:]))
    return(entered_list[:1] + flatten(entered_list[1:]))

if __name__ == "__main__":
    s = [[1, 2], [3, 4], [3, [4, [5, 6, 7, [8, 9, [10, 11], 12]]], 13], 14, [15, 16, 17]]
    f = flatten
    # передаем в функцию копию листа s, т.к. изменять его по парадигме нельзя.
    new_s = f(s[:])
    # Используем ф-цию высшего порядка map, для преобразования элементов массива в строки
    print(list(map(str, new_s)))

"""Основные принципы Функционального программирования:
1) Функции являются объектами первого класса (First Class Object).
Это означает, что с функциями вы можете работать, также как и с данными – передавать их в качестве аргументов другим
функциям, присваивать переменным и т.п.
2) Использование рекурсии в качестве основной структуры контроля потока управления. В некоторых языках не существует
иной конструкции цикла, кроме рекурсии.
3) Акцент на обработке списков (lists, отсюда название Lisp – LISt Processing). Списки с рекурсивным обходом подсписков
часто используются в качестве замены циклов.
4) Используются функции высшего порядка (High Order Functions). Функции высшего порядка – функции, которые могут
в качестве аргументов принимать другие функции.
Функции высшего порядка принимают в качестве аргументов другие функции. В стандартную библиотеку Python входит
достаточно много таких функций, в качестве примера приведем функцию map.
Она принимает функцию и Iterable объект, применяет функцию к каждому элементу Iterable объекта и возвращает Iterator
объект, который итеративно возвращает все модифицированные после функции элементы.
6) Акцент на том, что должно быть вычислено, а не на том, как вычислять.

Вывод: все концепции соблюдены."""

```

main (2) ×

```

"C:\Users\User\Pictures\уник\2 курс\Парадигмы программирования\venv\Scripts\python.exe" "C:/Users/User/Pictures/уник/2
['1', '2', '3', '4', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12', '13', '14', '15', '16', '17']

```

Более подробно – на странице задания на гитхабе: [JustBlood/Prog\\_Paradigms \(github.com\)](https://github.com/JustBlood/Prog_Paradigms)

**Проект представлен преподавателю в электронной форме, продемонстрирована работоспособность программы, разьяснены детали программного кода.**

**Вопросы, заданные преподавателем:**