

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №12
дисциплины «Программирование на Python»

Выполнила:
Кубанова Ксения Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

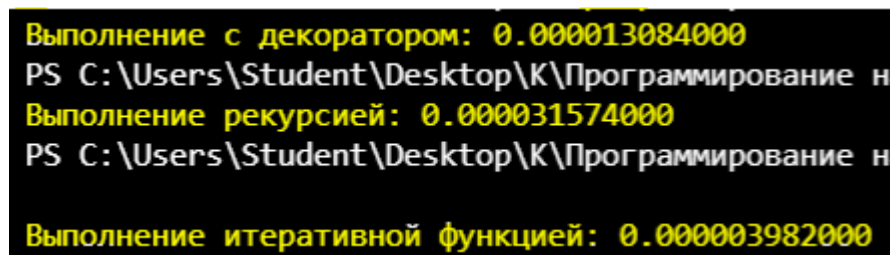
Тема: рекурсия

Цель: приобретение навыков по работе с рекурсией.

Порядок выполнения работы

Задание 7.

Требуется изучить работу со стандартным пакетом Python `timeit`. Оценить с помощью этого модуля скорость работы итеративной и рекурсивной версий функций `factorial` и `fib`. Во сколько раз изменится скорость работы рекурсивных версий функций `factorial` и `fib` при использовании декоратора `lru_cache`?



```
Выполнение с декоратором: 0.000013084000
PS C:\Users\Student\Desktop\K\Программирование н
Выполнение рекурсией: 0.000031574000
PS C:\Users\Student\Desktop\K\Программирование н
Выполнение итеративной функцией: 0.000003982000
```

Рисунок 1 – время выполнения функций

Основываясь на полученном времени при написании функций (рисунок 1, выделено жёлтым цветом) можно прийти к тому, что самое времязатратное – это использование рекурсии. Однако, если использовать их с декоратором `lru_cache`, то время сокращается практически в 3 раза. Что касается итеративного способа, он является самым быстрым в данном случае, однако не всегда.

Решения предоставлены в файлах *Iterative.py*, *Recurs.py*, *lru.py*.

Индивидуальное задание 1.

В6: Создайте процедуру, печатающую все возможные представления натурального числа N в виде суммы других натуральных чисел.

```

1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def vec(arr):
5      if len(arr) != 1:
6          for i in arr:
7              print(i, end = " ")
8          print()
9
10
11 def find(arr, i, n):
12     if n == 0:
13         vec(arr)
14     for j in range(i, n + 1):
15         arr.append(j)
16         find(arr, j, n - j)
17         del arr[-1]
18
19
20 if __name__ == '__main__':
21     n = 4
22     arr = []
23     find(arr, 1, n)
24

```

Рисунок 2 – код индивидуального задания 1

```

1 1 1 1
1 1 2
1 3
2 2

```

Рисунок 3 – вывод индивидуального задания 1

Решение предоставлено в файле *ind1.py*.

Контрольные вопросы

1 Для чего нужна рекурсия?

Рекурсия функции нужна, когда требуется выполнить последовательность из одинаковых действий.

2 Что называется базой рекурсии?

База рекурсии – это такие аргументы функции, которые делают задачу настолько простой, что решение не требует дальнейших вложенных вызовов.

3 Самостоятельно изучите что является стеком программы. Как используется стек программы при вызове функций?

Стек — это вид структуры данных, в котором элементы упорядочены и добавление или удаление элементов происходит с верхней части стека — по принципу «Последним пришел — первым ушел».

4 Как получить текущее значение максимальной глубины рекурсии в языке Python?

Чтобы проверить текущие параметры лимита, нужно запустить:

```
sys.getrecursionlimit()
```

5 Что произойдет если число рекурсивных вызовов превысит максимальную глубину рекурсии в языке Python?

Когда предел достигнут, возникает исключение `RuntimeError` :

```
RuntimeError: Maximum Recursion Depth Exceeded
```

6 Как изменить максимальную глубину рекурсии в языке Python?

Можно изменить предел глубины рекурсии с помощью вызова:

```
sys.setrecursionlimit(limit)
```

7 Каково назначение декоратора `lru_cache` ?

Используется для уменьшения количества лишних вычислений, ускоряя работу программы.

8 Что такое хвостовая рекурсия? Как проводится оптимизация хвостовых вызовов?

Хвостовая рекурсия — частный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с рекурсией.