

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

Отчет по лабораторной работе №2.9

Рекурсия в языке Python

по дисциплине «Технологии программирования и алгоритмизация»

Выполнил

студент группы ИВТ-б-о-20-1

Дыбов Д.В. « » _____ 20__ г.

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)

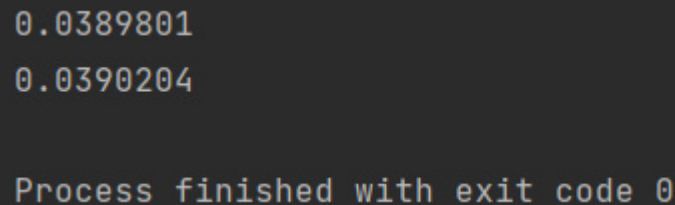
Ставрополь 2021

Цель работы: приобретение навыков по работе с рекурсивными функциями при написании программ с помощью языка программирования Python версии 3.x.

Ход работы

Ссылка на репозиторий:

1. Создал новый репозиторий для лабораторной работы №2.9;
2. Клонировал созданный репозиторий на компьютер;
3. Создал новый PyCharm проект в папке репозитория;
4. С помощью модуля timeit проверил скорость работы функций factorial и fib;

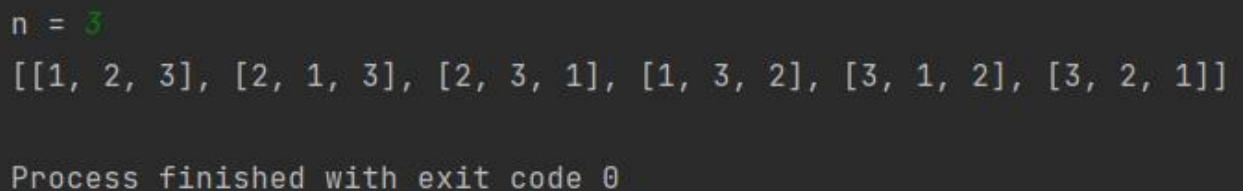


```
0.0389801
0.0390204
Process finished with exit code 0
```

Рисунок 1 – Результат проверки скорости работы функций

5. Приступил к выполнению индивидуального задания согласно своему варианту;

6. Проверил работоспособность кода:



```
n = 3
[[1, 2, 3], [2, 1, 3], [2, 3, 1], [1, 3, 2], [3, 1, 2], [3, 2, 1]]
Process finished with exit code 0
```

Рисунок 2 – Результат выполнения индивидуального задания

7. Перешёл на сайт для проверки кода на наличие ошибок и проверил задачу;

8. Результат не выдал ошибок:

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import timeit
5 from functools import lru_cache
6
7
8 @lru_cache
9 def factorial(number=1):
10     for i in range(1, number+1):
11         number += i
12
13

```

No syntax errors detected :)

Рисунок 3 – Результат проверки задачи на наличие ошибок

9. Проверил наличие ошибок в индивидуальном задании;

10. Результат не выдал ошибок;

```

1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4
5 def numbers(arr):
6     if len(arr) == 1:
7         return [arr]
8     else:
9         first = arr[0]
10        perm = numbers(arr[1:])
11        a = []
12        for perms in perm:
13            for i in range(len(perms)):

```

No syntax errors detected :)

Рисунок 4 – Проверка индивидуального задания на наличие ошибок

11. Отправил все изменения в репозитории;

```
C:\Program Files\Git>cd laba2.9

C:\Program Files\Git\laba2.9>git add .

C:\Program Files\Git\laba2.9>git commit -m "Add python files"
[main 23fca58] Add python files
 3 files changed, 82 insertions(+)
 create mode 100644 Individual.py
 create mode 100644 Primer.py
 create mode 100644 Zada4a.py

C:\Program Files\Git\laba2.9>git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.21 KiB | 1.21 MiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/dibovdmity/laba2.9.git
   49f7fdb..23fca58  main -> main

C:\Program Files\Git\laba2.9>
```

Рисунок 5 – Сохранение изменений

Контрольные вопросы

1. Для чего нужна рекурсия?

Рекурсия – это способ организации циклического процесса путем вызова рекурсивной функции.

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 программы – это абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO.

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

Чтобы узнать текущее значение максимальной глубины рекурсии необходимо написать `sys.getrecursionlimit()`

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

Если число рекурсивных вызовов превысит максимальную глубину рекурсии, то возникнет исключение `RuntimeError: Maximum Recursion Depth Exceeded`.

6. Как изменить максимальную глубину рекурсии в языке Python?

Чтобы изменить предел глубины рекурсии в языке python необходимо использовать `sys.setrecursionlimit(limit)`.

7. Каково назначение декоратора `lru_cache`?

Декоратор `lru_cache` оборачивает функцию с переданными в нее аргументами и запоминает возвращаемый результат соответствующий этим аргументам.

8. Что такое хвостовая рекурсия? Как проводится оптимизация хвостовых вызовов— конвертировать часовые пояса.

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

Оптимизация хвостовой рекурсии путём преобразования её в плоскую итерацию реализована во многих оптимизирующих компиляторах. В некоторых функциональных языках программирования спецификация гарантирует обязательную оптимизацию хвостовой функции.

Вывод: в ходе выполнения лабораторной работы были приобретены навыки по работе с рекурсивными функциями при написании программ с помощью языка программирования Python версии 3.x.