

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ

Институт Цифрового Развития

Кафедра инфокоммуникаций

ОТЧЁТ

по лабораторной работе №4

Дисциплина: «Языки программирования»

Выполнил: студент 2 курса

группы ИТС-б-о-20-1

Харченко Екатерина Владимировна

Проверил:

К.т.н., доцент кафедры инфокоммуникаций

Воронкин Роман Александрович

Работа защищена с оценкой: \_\_\_\_\_

Ставрополь, 2021 г.

## Рекурсия в языке Python

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

Порядок выполнения работы:

1. Создание общедоступного репозитория на GitHub, в котором использована лицензия MIT и язык программирования Python.
3. Выполняем клонирование созданного репозитория.
4. Самостоятельно изучим работу со стандартным пакетом Python `timeit`.

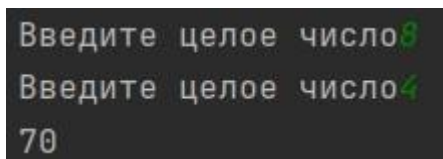
Оцените с помощью этого модуля скорость работы итеративной и рекурсивной версий функций `factorial` и `fib`. Во сколько раз изменится скорость работы рекурсивных версий функций `factorial` и `fib` при использовании декоратора `lru_cache`? Приведите в отчет и обоснуйте полученные результаты.

При использовании декоратор `lru_cache` на примере последовательности Фибоначчи, он имеет неэффективное время работы, из-за того, что выполняет 2 рекурсивных вызова.

5. Выполним индивидуальное задание вариант 9.

9. Даны целые числа  $m$  и  $n$ , где  $0 \leq m \leq n$ , вычислить, используя рекурсию, число сочетаний  $C_n^m$  по формуле:  $C_n^0 = C_n^n = 1$ ,  $C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$  при  $0 < m < n$ . Воспользовавшись формулой

$$C_n^m = \frac{n!}{m!(n-m)!} \quad (1)$$



```
Введите целое число8
Введите целое число4
70
```

Рис 1.1- пример работы программы

6. Зафиксируем сделанные изменения в репозитории.
7. Выполним слияние ветки для разработки с веткой `main`.
8. Отправим сделанные изменения на сервер GitHub.

Контрольные вопросы:

1. Для чего нужна рекурсия?

Функция может содержать вызов других функций. В том числе процедура может вызвать саму себя. Компьютер лишь последовательно выполняет встретившиеся ему в программе команды и, если встречается вызов процедуры, просто начинает выполнять эту функцию. Без разницы, какая функция дала команду это делать.

2. Что называется базой рекурсии?

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

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

Чтобы проверить текущие параметры лимита, нужно запустить:  
`sys.getrecursionlimit()`

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

Существует предел глубины возможной рекурсии, который зависит от реализации Python. Когда предел достигнут, возникает исключение `RuntimeError`.

5. Как изменить максимальную глубину рекурсии в языке Python?

Можно изменить предел глубины рекурсии с помощью вызова:  
`sys.setrecursionlimit(limit)`

6. Каково назначение декоратора `lru_cache` ?

Полезным инструментом является декоратор `lru_cache` , который можно использовать для уменьшения количества лишних вычислений.

7. Что такое хвостовая рекурсия?

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

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