

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ» ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет по лабораторной работе №12 по
дисциплине: основы программной
инженерии**

Выполнил:

студент группы ПИЖ-б-о-21-1

Прокопов Дмитрий Владиславович

Проверил:

доцент кафедры инфокоммуникаций

Воронкин Р.А.

Ставрополь, 2022 г.

Пример 1

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def recursion(n):
    if n == 1:
        return 1

    return n + recursion(n - 1)

if __name__ == '__main__':
    print(f"Recursion: {recursion(5)}")

    n = 0
    k = 5
    for i in range(1, k + 1):
        n += i

    print(f"'for': {n}")

C:\Users\dmidt\Pycha
Recursion: 15
'for': 15
```

Индивидуальное задание

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 $0 \leq m \leq n$.

```
1  ▶  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4
5  def cmn(n, m):
6      if m == n or m == 0:
7          return 1
8      if m != 1:
9          return cmn(n - 1, m) + cmn(n - 1, m - 1)
10     else:
11         return n
12
13
14  ▶  if __name__ == '__main__':
15      print(cmn(9, 3))
```

ind x

↑ C:\Users\dmidt\PycharmProjects\lb12\venv\Scripts\python.e

↓ 84

Контрольные вопросы:

1. Рекурсия существенно сокращает объем кода и входит во многие встроенные функции языков.
2. База рекурсии – это тривиальный случай, при котором решение задачи очевидно, то есть не требуется обращение функции к себе.
3. Компьютер использует стек вызовов — специальную область памяти, где хранит данные о точках перехода между фрагментами кода.
последовательность шагов, выполняемых при вызове функции: а. Программа сталкивается с вызовом функции. б. Создается фрейм стека, который помещается в стек. в. Процессор переходит к точке начала выполнения функции. г. Инструкции внутри функции начинают выполняться. После завершения функции, выполняются следующие шаги: д. Регистры восстанавливаются из стека вызовов. е. Фрейм стека вытягивается из стека. Освобождается память, которая была выделена для всех локальных переменных и аргументов. ж. Обрабатывается возвращаемое значение. з. ЦП возобновляет выполнение кода (исходя из обратного адреса).
4. Чтобы получить текущее значение максимальной глубины рекурсии следует вызвать функцию `sys.getrecursionlimit()`
5. Когда предел достигнут, возникает исключение: `RuntimeError: Maximum Recursion Depth Exceeded`
6. Можно изменить предел глубины рекурсии с помощью вызова: `sys.setrecursionlimit(limit)`

7. `lru_cache` можно использовать для уменьшения количества лишних вычислений.

8. Хвостовая рекурсия — частный случай рекурсии, при котором любой рекурсивный вызов является последней операцией перед возвратом из функции. Оптимизация происходит, вызывая исключение, если оно является его прародителем, и перехватывает исключения, чтобы подделать оптимизацию хвоста.