Практическая работа №1

Рекурсия, рекурсивные алгоритмы

1. Рекурсия в широком смысле – это описание объекта, ссылающегося на самого себя
2. 1) Матрешка достается из другой матрешки, которая достается из другой матрешки
   2) Деление картошки на кусочки, деление кусочков на кусочки
   3) Веточка дерева делится на другие веточки, которые делятся на другие веточки
   4) Зеркало в зеркале – одно зеркало повторяет отражение другого, которое отражает отражение первого
   5) Демонстрация экрана в демонстрации экрана – одна демонстрация экрана показывает изображение второй демонстрации экрана, в которой находится первая
3. Рекурсивный алгоритм – это алгоритм, в определении которого содержится прямой или косвенный вызов этого же алгоритма.
4. Рекурсивная триада:
   1) Параметризация – выделяют параметры, которые используются для описания условия задачи, а затем в решении.
   2) База рекурсии – определяют тривиальный случай, при котором решение очевидно, то есть не требуется обращение функции к себе.
   3) Декомпозиция – выражают общий случай через более простые подзадачи с измененными параметрами.
5. Полное дерево – это дерево, в котором каждый узел имеет либо нуль, либо два потомка.
   Глубина рекурсии – это количество уровней дерева, то есть количество шагов от корня до самого глубокого узла.
   Объем рекурсии – это количество раз, которое функция вызывает саму себя в процессе выполнения. В рекурсивной функции объем рекурсии описывает, насколько глубоко вложены вызовы функции друг в друга перед их завершением.
6. Область памяти, предназначенная для хранения всех промежуточных значений локальных переменных при каждом следующем рекурсивном обращении, образует рекурсивный стек.
   1) Для каждого текущего обращения формируется локальный слой данных стека

2) Завершение вычислений происходит посредством восстановления значений данных каждого слоя в порядке, обратном рекурсивным обращениям.

7. Пример рекурсивной функции на Python.

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1) – пример декомпозиции
```

Построим полное дерево рекурсии для вызова функции factorial(3). Определим его глубину и объем. Это является примером параметризации.

```
factorial(3)
    |
    |--> factorial(2)
            |
            |--> factorial(1)
                    |
                    |--> factorial(0)
```

Это полное дерево рекурсии для вызова функции факториала при n=3. Глубина этого дерева составляет 3, а объем - 4 (т.е. общее количество вызовов функции, включая начальный вызов). Это является примером базы рекурсии

factorial(3) = 3 * factorial(2)
factorial(2) = 2 * factorial(1)
factorial(1) = 1 * factorial(0)
factorial(0) = 1

Таким образом, глубина - это количество уровней в дереве рекурсии, а объем - это общее количество вызовов функции в этом дереве.

8. Мемчик