**Практическая работа №5**

**Рекурсия, рекурсивные алгоритмы**

1. Изучите материал в презентации.
2. Дайте определение рекурсии (в широком смысле).
3. Приведите примеры рекурсии в жизни и опишите их (4-5 примеров, можно фото).
4. Дайте определение рекурсивного алгоритма.
5. Что такое рекурсивная триада?
6. Что такое полное дерево, глубина и объем рекурсии?
7. Как называется область памяти, выделяемая для хранения всех промежуточных значений локальных переменных? Опишите ее.
8. Приведите пример рекурсивной процедуры/функции. Постройте полное дерево рекурсии, определите глубину и объем рекурсии. Укажите рекурсивную триаду.
9. Приведите примеры мемчиков на рекурсии. Опишите их (желательно сделать свой).
10. Рекурсия в широком смысле – это определение объекта посредством ссылки на себя.
11. - Наведение веб-камеры на экран монитора компьютера. Камера будет записывать изображение экрана компьютера и выводить его же на этот экран, получится что-то вроде замкнутого цикла.

- Географические объекты. Например, острова посреди озёр, которые в свою очередь расположены на островах.

- Круговорот воды в природе.

- Два поставленных друг напротив друга зеркала. В них образуются два коридора из затухающих отражений зеркал.

1. Рекурсивный алгоритм – это алгоритм, в определении которого содержится прямой или косвенный вызов этого же алгоритма
2. Рекурсивная триада – это этапы для решения задач рекурсивными методами, включающие в себя
   1. Параметризацию – 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.
3. - Дерево рекурсии – схема для оценки трудоемкости рекурсивных алгоритмов

- Глубина рекурсии – количество одновременно выполняемых процедур.

- Объем рекурсии - количество вершин полного рекурсивного дерева без единицы

1. Рекурсивный стек:

- Для каждого текущего обращения формируется локальный

слой данных стека (при этом совпадающие идентификаторы

разных слоев стека независимы друг от друга и не

отождествляются).

- Завершение вычислений происходит посредством

восстановления значений данных каждого слоя в порядке,

обратном рекурсивным обращениям.

- Количество рекурсивных обращений ограничено размером

области памяти, выделяемой под программный код.

1. Числа Фибоначчи

public class Program

{

public static int Fibonacci(int n)

{

if (n <= 1) return n;

return Fibonacci(n - 1) + Fibonacci(n - 2);

}

public static void Main(string[] args)

{

Console.WriteLine(Fibonacci(3));

}

Дерево рекурсии:

Fibonacci(3)

├── Fibonacci(2)

│ ├── Fibonacci(1) -> 1

│ └── Fibonacci(0) -> 0

└── Fibonacci(1) -> 1

Глубина рекурсии равна 3.

Объем рекурсии: 5 вызовов.

Рекурсивная триада:

База рекурсии: if (n <= 1) return n;

Параметризация: public static int Fibonacci(int n)

Декомпозиция: return Fibonacci(n - 1) + Fibonacci(n - 2);

1. 

