

Лабораторная работа №6st - Tree (BST).

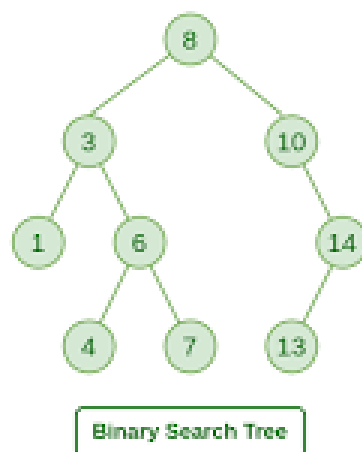
Основные требования:

- каждая функция должна иметь docstring - множ-й комментарий (по типу что делает данная функция)
- написание функций должно быть компактным (не в 100 строчек кода)
- именованя функций должны быть нормальными и четко отражать смысл самой функции
- код должен соответствовать стандарту языка Python (PEP).
- классы и объекты должны соответствовать стандартам языка программирования Python.

Общее задание к Л/Р:

Цель - реализация бинарного дерева поиска (BST) для управления списком контактов.

Вы работаете над приложением для управления списком контактов. Вам необходимо создать структуру данных и методы для добавления, поиска и удаления контактов. Контакты будут храниться в алфавитном порядке по именам.



Структура узла дерева:

```
class TreeNode:
    def __init__(self, key):
        self.left = None
        self.right = None
        self.val = key
```

Структура класса дерева:

```
class BinarySearchTree:
    def __init__(self):
        self.root = None

    def insert(self, name, phone):
        pass

    def search(self, name):
        pass

    def _search_recursively(self, root, name):
        pass

# остальные методы класса
```

Основной файл main.py:

- Реализуйте класс для бинарного дерева поиска (**BST**), который будет использоваться для хранения контактов.
- Каждый узел дерева будет представлять один контакт и содержать следующую информацию:
 - Имя контакта (ключ).
 - Номер телефона (значение).
- Реализуйте следующие методы в классе **BST**:
 - **insert(name, phone)**: Добавление нового контакта.
 - **search(name)**: Поиск контакта по имени и возврат номера телефона.
 - **delete(name)**: Удаление контакта по имени.
- Реализуйте метод **display_contacts()**, который будет выводить список контактов в алфавитном порядке.
- проверить код на соот-е **PEP**
- продемонстрируйте работу кода