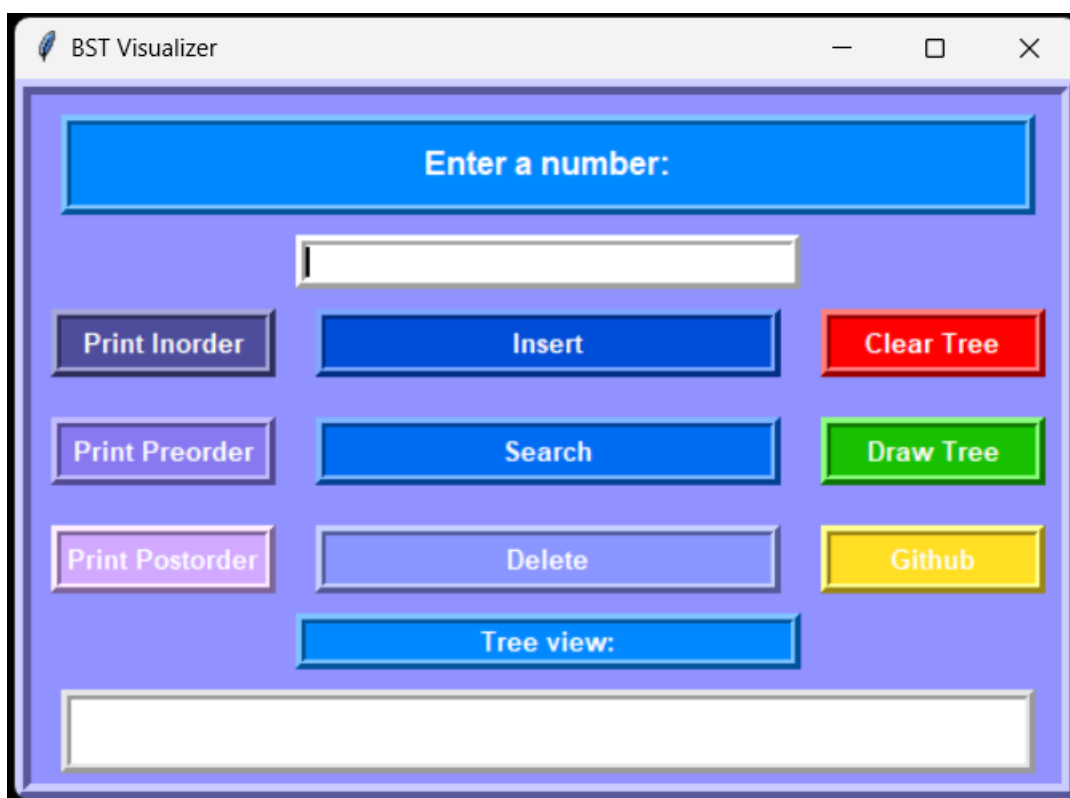


OPIS I DOKUMENTACJA PROJEKTU BST VISUALIZER



Autor: Agata Marczyk, ISI sem.2.

Github: <https://github.com/soberyoda>

Pliki z kodem źródłowym zostały umieszczone w folderze BST

Użyte języki programowania, technologie, biblioteki:

- Python 3.11.3 :
 - Tkinter – biblioteka języka Python umożliwiająca tworzenie interfejsu graficznego, w projekcie użyta do stworzenia mini aplikacji okienkowej. Dołączona do standardowych instalacji Pythona w systemach Linux, Microsoft Windows i MacOS X.
Strona z dokumentacją: <https://docs.python.org/3/library/tkinter.html>
 - webbrowser – moduł przeglądarki internetowej – w projekcie użyty do dodania linku przekierowującego użytkownika do profilu github autora projektu.
Strona z dokumentacją: <https://docs.python.org/3/library/webbrowser.html>
 - Graphviz – pakiet ułatwiający tworzenie i renderowanie opisów wykresów w języku DOT, w projekcie użyty do utworzenia pliku zawierającego wykres zdefiniowanego przez użytkownika grafu drzewa BST.
Strona pakietu: <https://pypi.org/project/graphviz/>
Instalacja: <https://graphviz.org/download/>

Przed rozpoczęciem korzystania należy dodać ścieżkę dostępu do katalogu Graphviz/bin do zmiennej Path, lub umieścić w programie:

```
import os
os.environ["PATH"] += os.pathsep + 'ścieżka do katalogu Graphviz/bin'
```

- IDE : <https://www.jetbrains.com/pycharm/>

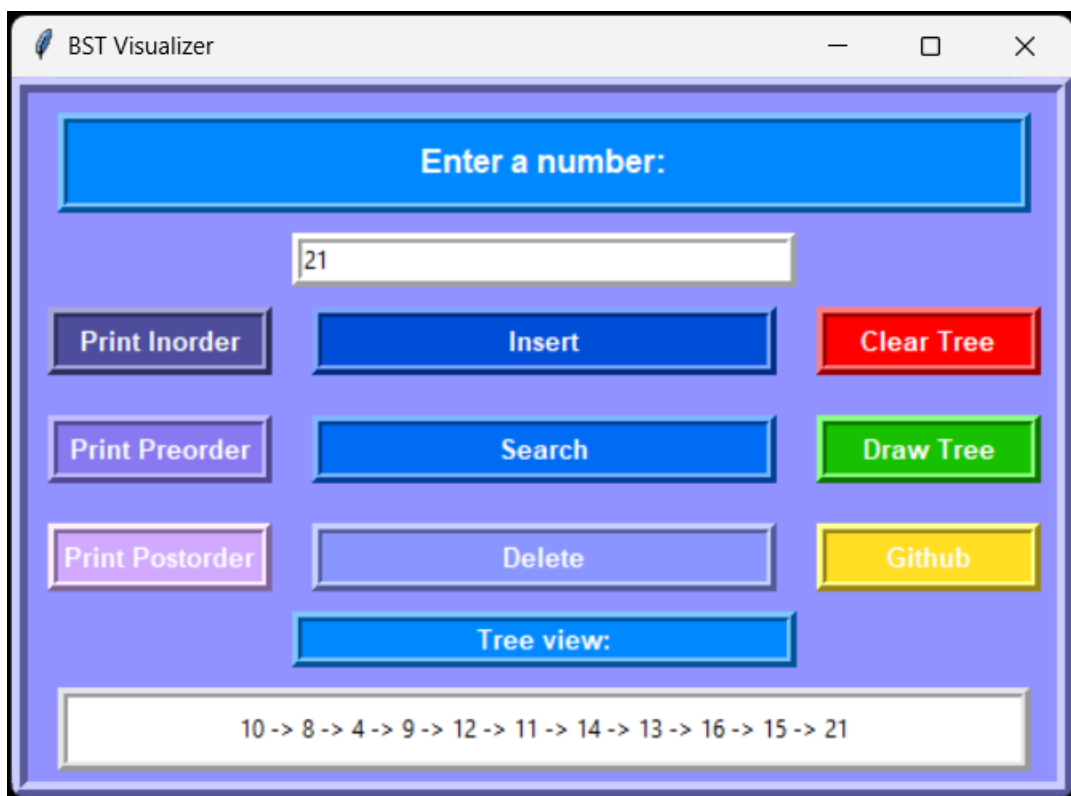
Opis i prezentacja działania funkcji aplikacji:

Po uruchomieniu programu przez użytkownika na ekranie pojawia się interaktywny interfejs graficzny umożliwiający wykonanie podstawowych operacji na drzewie BST – dodania, usunięcia oraz wyszukania elementu w drzewie (zwraca True jeśli element znajduje się w drzewie lub None w przeciwnym wypadku). W programie dodałam możliwość wyświetlenia lokalizacji w pamięci danego elementu – należy w tym celu obłżyć komentarzem lub usunąć odpowiednio zaznaczony w programie fragment kodu) a także przejścia drzewa metodami „w głąb” (inorder, preorder, postorder)- zwracają one ciąg liczb uporządkowany według założeń każdego przypadku.

Dodatkowo zdefiniowane zostały trzy operacje – Clear tree, która to usuwa całe wprowadzone przez użytkownika drzewo, Draw tree która generuje graf i zapisuje go w pliku pdf w katalogu źródłowym oraz Github – przekierowująca użytkownika do profilu Github autora projektu.

Poniżej umieszone są zdjęcia obrazujące działanie pojedynczych operacji (wstawiania do drzewa, usunięcia elementu, przejścia drzewa metodą inorder oraz procedury Draw tree)

DODANIE ELEMENTÓW DO DRZEWY BST



USUNIĘCIE ELEMENTU 13 Z DRZEWA

BST Visualizer

Enter a number:

13

Print Inorder

Print Preorder

Print Postorder

Insert

Search

Delete

Clear Tree

Draw Tree

Github

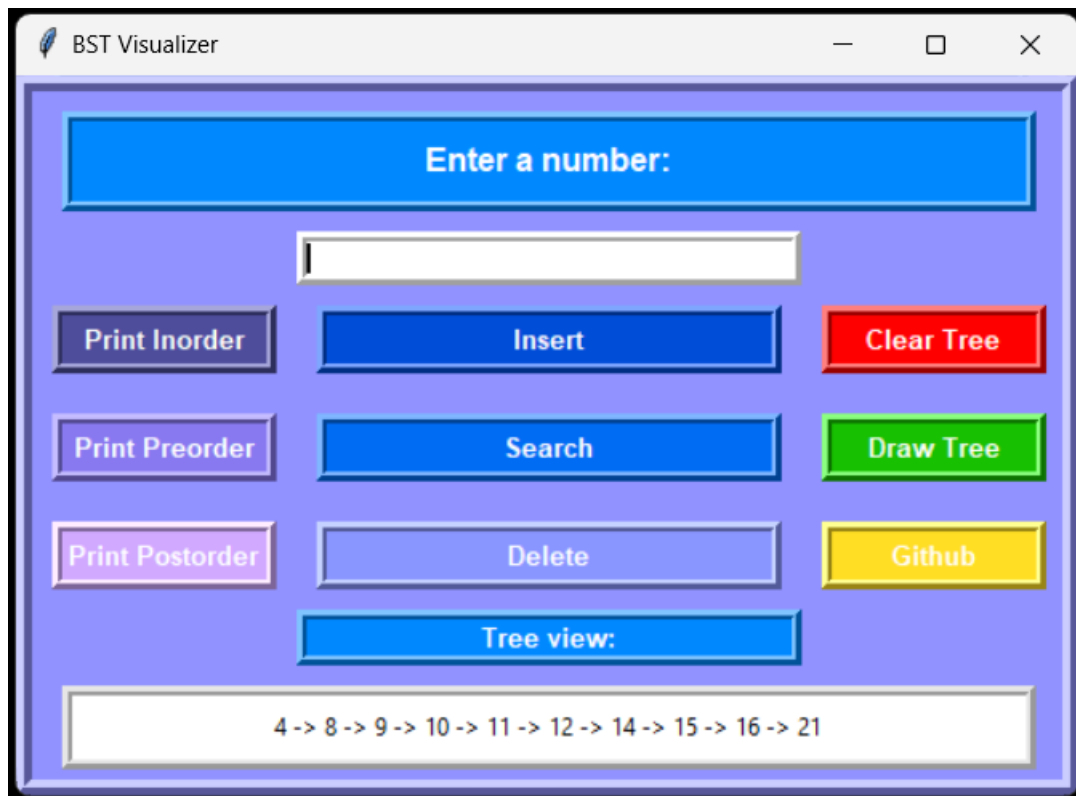
Tree view:

10 -> 8 -> 4 -> 9 -> 12 -> 11 -> 14 -> 16 -> 15 -> 21

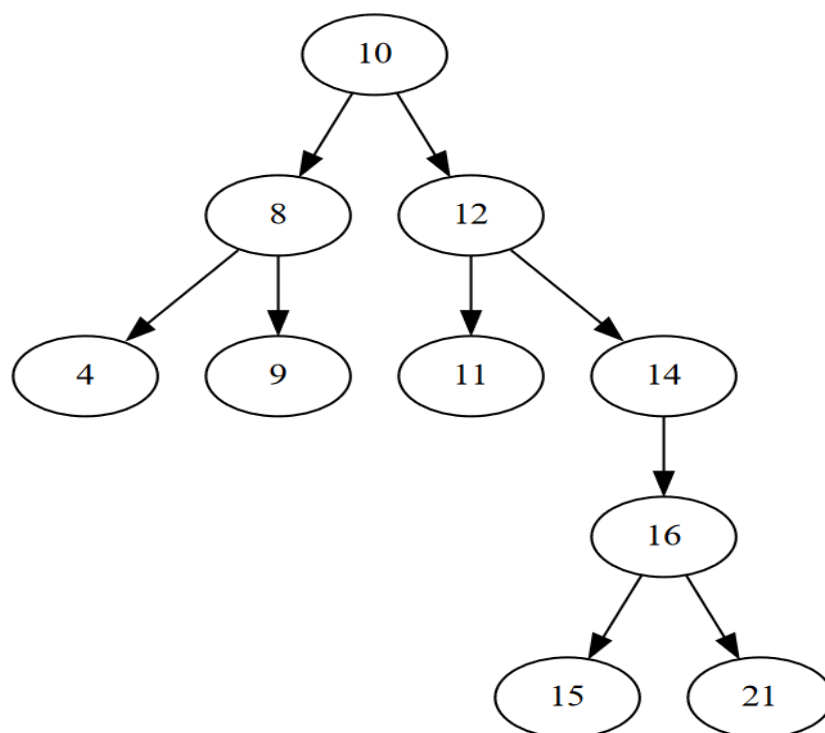
WYSZUKANIE ELEMENTU 4 W DRZEWIE

The screenshot shows a window titled "BST Visualizer" with standard window controls. The interface has a light blue background. At the top, there is a blue button labeled "Enter a number:". Below it, a text input field contains the number "4". To the left of the input field are three buttons: "Print Inorder" (dark blue), "Print Preorder" (light blue), and "Print Postorder" (very light blue). To the right of the input field are three buttons: "Insert" (blue), "Search" (blue), and "Delete" (light blue). Further to the right are two buttons: "Clear Tree" (red) and "Draw Tree" (green). Below the "Delete" button is a yellow button labeled "Github". At the bottom of the interface, there is a blue button labeled "Tree view:" and a large white text box containing the word "True".

PRZEJŚCIE PRZEZ DRZEWO METODĄ INORDER



PREZENTACJA DZIAŁANIA PROCEDURY DRAW TREE



```
1 digraph {
2     10
3     10 -> 8
4     8
5     8 -> 4
6     4
7     8 -> 9
8     9
9     10 -> 12
10    12
11    12 -> 11
12    11
13    12 -> 14
14    14
15    14 -> 16
16    16
17    16 -> 15
18    15
19    16 -> 21
20    21
21 }
22 |
```

```
tree.gv
tree.gv
```