Binary Tree

- Introduction

Binary Tree에 저장되어 있는 요소들은 트리 형태로 표현되며 요소 간에 부모, 자식 관계가 존재하고 트리의 형태는 complete binary tree입니다.

- Declaration

|  |  |
| --- | --- |
| **NO** | **Constructor** |
| 1 | * var bt = new BinaryTree(); |

- Methods

|  |  |
| --- | --- |
| **No** | **Method & Description** |
| 1 | binary\_tree::isEmpty()  • This method returns if binary tree is empty. |
| 2 | binary\_tree::size()  • This method returns the number of elements in this binary tree. |
| 3 | binary\_tree::push(value)  • This method inserts the specified element into this binary tree. |
| 4 | binary\_tree::pop()  • This method removes the very first element from this binary tree. |
| 5 | binary\_tree::postOrder(value)  • This method is used for tree traversal. Traversal order is left -> right -> parent. |
| 6 | binary\_tree::inOrder(value)  • This method is used for tree traversal. Traversal order is left -> parent -> right. |
| 7 | binary\_tree::preOrder(value)  • This method is used for tree traversal. Traversal order is parent -> left -> right. |
| 8 | binary\_tree::state()  • This method shows state of this binary tree. |
| 9 | binary\_tree::clear()  • This method nullifies this binary tree and make all variables initial. |

1. binary\_tree::isEmpty()

* Description

This isEmpty() method is used to check if this binary tree is empty.

* Declare

Following is the declaration binary\_tree.isEmpty() method.

|  |
| --- |
| var bt = new BinaryTree();  var ret = bt.isEmpty(); |

* Parameter

NA

* Return Value

The method returns ‘true’ if this binary tree is empty, or ‘false’ if this binary tree is not empty.

* Exception

NA

* Example

|  |
| --- |
|  |

1. binary\_tree::size()

* Description

This size() method is used to get the number of elements in this binary tree.

* Declare

Following is the declaration binary\_tree.size() method.

|  |
| --- |
| var bt = new BinaryTree();  var size = bt.size(); |

* Parameter

NA

* Return Value

The method returns the number of elements in this binary tree.

* Exception

NA

* Example

|  |
| --- |
|  |

1. binary\_tree::push(value)

* Description

This push(value) method is used to insert the specified element into this binary tree.

* Declare

Following is the declaration binary\_tree.push(value) method.

|  |
| --- |
| var bt = new BinaryTree();  bt.push(1);  bt.push(2); |

* Parameter

value – The element to be inserted to this binary tree.

* Return Value

NA

* Exception

NA

* Example

|  |
| --- |
|  |

1. binary\_tree::pop()

* Description

This pop() method is used to remove the very first element from this binary tree.

* Declare

Following is the declaration binary\_tree.pop() method.

|  |
| --- |
| var bt = new BinaryTree();  bt.pop(); |

* Parameter

NA

* Return Value

The method returns the first element from this binary tree.

* Exception

The method returns null if binary tree is empty.

* Example

|  |
| --- |
|  |

1. binary\_tree::postOrder(value)

* Description

This postOrder(value) method is used for tree traversal. Traversal order is left -> right -> parent.

* Declare

Following is the declaration binary\_tree.postOrder(value) method.

|  |
| --- |
| var bt = new BinaryTree();  bt.postOrder(this.root); |

* Parameter

value – The node is used to tree traversal recursively.

* Return Value

The method returns null if node has no data.

* Exception

The method returns null if node has no data.

* Example

|  |
| --- |
|  |

1. binary\_tree::inOrder(value)

* Description

This inOrder(value) method is used for tree traversal. Traversal order is left -> parent -> right.

* Declare

Following is the declaration binary\_tree.inOrder(value) method.

|  |
| --- |
| var bt = new BinaryTree();  bt.inOrder(this.root); |

* Parameter

value – The node is used to tree traversal recursively.

* Return Value

The method returns null if node has no data.

* Exception

The method returns null if node has no data.

* Example

|  |
| --- |
|  |

1. binary\_tree::preOrder(value)

* Description

This preOrder(value) method is used for tree traversal. Traversal order is parent -> left -> right.

* Declare

Following is the declaration binary\_tree.preOrder(value) method.

|  |
| --- |
| var bt = new BinaryTree();  bt.preOrder(this.root); |

* Parameter

value – The node is used to tree traversal recursively.

* Return Value

The method returns null if node has no data.

* Exception

The method returns null if node has no data.

* Example

|  |
| --- |
|  |

1. binary\_tree::state()

* Description

This state() method shows state of the binary\_tree.

* Declare

Following is the declaration binary\_tree.state() method.

|  |
| --- |
| var bt = new BinaryTree();  bt.state(); |

* Parameter

NA

* Return Value

NA

* Exception

NA

* Example

|  |
| --- |
|  |

1. binary\_tree::clear()

* Description

This clear() method is used to nullify this binary tree and make all variables initial.

* Declare

Following is the declaration binary\_tree.clear() method.

|  |
| --- |
| var bt = new BinaryTree();  bt.clear(); |

* Parameter

NA

* Return Value

NA

* Exception

NA

* Example

|  |
| --- |
|  |