

## პროგრამირების აბსტრაქციები

### სემინარის ამოცანები #21

ორობითი ძეგნის ხე

ააგეთ ორობითი ძეგნის ხის კლასი. ამ კლასის ობიექტებში უნდა შეგვეძლოს ახალი **int**-ების ჩასმა და მოცემული **int**-ისთვის იმის შემოწმება ეს რიცხვი უკვე გვაქვს თუ არა.

interface

```
#ifndef BST_H_
#define BST_H_

class bst {
private:
    struct Node {
        int val;
        Node* left;
        Node* right;
    };
    Node* root;
    void deleteSubtree(Node* subtree);
    Node*& findValue(Node*& subtree, int val);
public:
    bst();
    ~bst();
    bool contains(int val);
    void insert(int val);
};

#endif
```

## implementation

```
#include <cstdlib>
#include "bst.h"

void bst::deleteSubtree(Node* subtree) {
    if (subtree != NULL) {
        deleteSubtree(subtree -> left);
        deleteSubtree(subtree -> right);
        delete subtree;
    }
}

bst::Node*& bst::findValue(Node*& subtree, int val) {
    if (subtree == NULL || subtree -> val == val)
        return subtree;

    if (val < subtree -> val) {
        return findValue(subtree -> left, val);
    } else {
        return findValue(subtree -> right, val);
    }
}

bst::bst() {
    root = NULL;
}

bst::~~bst() {
    deleteSubtree(root);
}

bool bst::contains(int val) {
    return findValue(root, val) != NULL;
}

void bst::insert(int val) {
    Node*& destination = findValue(root, val);
    if (destination == NULL) {
        destination = new Node;
        destination -> val = val;
        destination -> left = NULL;
        destination -> right = NULL;
    }
}
```