<https://www.cyberforum.ru/cpp-beginners/thread1888806.html>

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

struct ListViolations

{

char\* violat;

ListViolations\* next;

};

void Push(ListViolations\*& node, char\* newStr)

{

ListViolations\* tmp = new ListViolations;

tmp->violat = new char[strlen(newStr) + 1];

strcpy(tmp->violat, newStr);

tmp->next = NULL;

if (node == NULL)

node = tmp;

else

{

tmp->next = node;

node = tmp;

}

}

void ClearList(ListViolations\*& Head)

{

ListViolations\* curr = Head;

while (curr)

{

ListViolations\* temp = curr->next;

delete[] curr->violat;

delete curr;

curr = temp;

}

}

void PrintList(ListViolations\* Head)

{

ListViolations\* curr = Head;

while (curr)

{

std::cout << curr->violat << std::endl;

curr = curr->next;

}

}

////////////////

///////////////////////////

struct TreeNode

{

TreeNode\* parent;

TreeNode\* left;

TreeNode\* right;

int number;

ListViolations\* list;

};

class database

{

private:

int size;

TreeNode\* root;

public:

database();

~database();

void Add(int value, char\*);

void Add(TreeNode\*&, TreeNode\*, int, char\*);

void PrintAll(TreeNode\*);

// void PrintNumber(int);

//void PrintRangeNumber(TreeNode\*);

TreeNode\* ReturnRoot();

void DeleteAll(TreeNode\* temp);

};

database::database()

{

root = NULL;

size = 0;

}

//деструктор

database::~database()

{

DeleteAll(root);

}

//метод получения корня

TreeNode\* database::ReturnRoot()

{

return root;

}

//метод удаления всего

void database::DeleteAll(TreeNode\* temp)

{

if (temp)

{

DeleteAll(temp->left);

DeleteAll(temp->right);

ClearList(temp->list);

delete temp;

}

}

//метод добавления

void database::Add(TreeNode\*& head, TreeNode\* parent, int value, char\* str)

{

TreeNode\* temp = new TreeNode;

temp->number = value;

temp->list = NULL;

temp->left = temp->right = temp->parent = NULL;

if (!head)

{

head = temp;

Push(head->list, str);

}

else

{

if (value < head->number)

Add(head->left, head, value, str);

else if (value > head->number)

Add(head->left, head, value, str);

else

Push(head->list, str);

}

}

void database::Add(int value, char\* str)

{

Add(root, NULL, value, str);

}

//метод печати полной базы данных

void database::PrintAll(TreeNode\* temp)

{

if (temp)

{

PrintAll(temp->left);

cout << "Номер машины: " << temp->number << endl;

cout << "Нарушения: " << endl;

PrintList(temp->list);

PrintAll(temp->right);

}

}

int main(int argc, char\* argv[])

{

setlocale(LC\_ALL, "Russian");

database bd;

bd.Add(9999, "Нарушение А");

bd.Add(9999, "Нарушение В");

bd.Add(9999, "Нарушение С");

bd.Add(2099, "Нарушение А");

bd.Add(1000, "Нарушение С");

bd.Add(9333, "Нарушение С");

bd.Add(9055, "Нарушение А");

bd.Add(9055, "Нарушение В");

bd.PrintAll(bd.ReturnRoot());

}