

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**КУРСОВАЯ РАБОТА**  
**по дисциплине «Алгоритмы и структуры данных»**  
**ТЕМА: ИДЕАЛЬНО СБАЛАНСИРОВАННЫЕ БДП – вставка и исключение.**

Студент(ка) гр. 9381

Шахин Н.С

Преподаватель

Фирсов М.А.

Санкт-Петербург

2020

## **ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

Студент Шахин Н.С.

Группа 9381

Тема работы:

Вариант 21. Идеально сбалансированные бдп – вставка и исключение.

Демонстрация.

Исходные данные:

Последовательность элементов, из которых нужно построить идеально сбалансированное бдп.

Содержание пояснительной записки:

«Содержание»

«Введение»

«Ход выполнения»

«Тестирование»

«Заключение»

«Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 31.10.2020

Дата сдачи реферата: 14.12.2020

Дата защиты реферата: 15.12.2020

Студент

Шахин Н.С.

Преподаватель

Фирсов М.А.

## **АННОТАЦИЯ**

На языке программирования C++ был написан класс идеально сбалансированного бинарного дерева поиска. Продемонстрирована вставка и удаление введённых элементов. Был написан gui для взаимодействия с программой.

## **SUMMARY**

A class for a perfectly balanced binary search tree was written in the C++ programming language. Inserting and deleting the entered elements is demonstrated. A gui was written to interact with the program.

## СОДЕРЖАНИЕ

	Введение	5
1.	Задание	6
2.	Описание алгоритма	6
2.1	Построение дерева	6
2.2	Поиск в дереве	6
2.3	Добавление элемента	6
2.4	Удаление элемента	6
3.	Описание структур данных и функций.	7
3.1	Класс Node	7
3.2	Класс BinaryTree	7
3.3	Класс InOut	9
3.4	Класс MyView	10
3.5	Класс MainWindow	10
4.	Описание интерфейса пользователя.	11
	Заключение	11
	Список источников	12
	Приложение А. Тестирование.	13
	Приложение Б. исходный код программы.	15

## ВВЕДЕНИЕ.

Целью работы является написать программу, реализующую идеально сбалансированное БДП. Для этого необходимо изучить соответствующую структуру данных, операции вставки, удаления и поиска в ней.

Бинарное дерево поиска - это двоичное дерево, для которого выполняются следующие дополнительные условия - свойства дерева поиска:

- Оба поддерева — левое и правое — являются двоичными деревьями поиска.

- У всех узлов левого поддерева произвольного узла  $X$  значения ключей данных меньше, нежели значение ключа данных самого узла  $X$ .

- У всех узлов правого поддерева произвольного узла  $X$  значения ключей данных больше, нежели значение ключа данных самого узла  $X$ .

Очевидно, данные в каждом узле должны обладать ключами, на которых определена операция сравнения меньше

Идеально сбалансированным назовем такое бинарное дерево  $T$ , что для каждого его узла  $x$  справедливо соотношение  $|n_L(x) - n_R(x)| \leq 1$ , где  $n_L(x)$  — количество узлов в левом поддереве узла  $x$ , а  $n_R(x)$  — количество узлов в правом поддереве узла  $x$ .

## **1. ЗАДАНИЕ.**

Реализовать структуру данных “Идеально сбалансированное БДП” и продемонстрировать операции вставки, удаления и поиска.

## **2. ОПИСАНИЕ АЛГОРИТМА.**

### **2.1 Построение дерева.**

Сначала данные сохраняются в массиве и сортируются в порядке возрастания, и считается длина массива без повторяющихся элементов, для того чтобы построить идеально сбалансированное дерево. Вызывается рекурсивный метод построения БДП. По формулам высчитывается сколько элементов должно быть в левом и правом поддереве корня ( $nL = n/2$ .  $nR = n - nL - 1$ ), функция вызывается для левого поддерева, потом записывается корень, затем функция вызывается для правого поддерева. Если элементы в массиве повторяются, то увеличивается поле amount, хранящее количество повторений.

### **2.2 Поиск в дереве.**

Поиск в дереве производится рекурсивно функцией find. Поиск производится по принципу: если элемент равен значению корня, то он находится в корне дерева. Если элемент меньше значения корня, то он находится в левом поддереве элемента. Если элемент больше корня, то он находится в правом поддереве. Если выбранное поддерево оказывается пустым, то поиск завершается «неудачно» – элемента x в дереве b нет.

### **2.3 Добавление элемента в дерево.**

Чтобы добавить элемент в дерево нужно перестроить его полностью, поэтому все элементы дерева сохраняются в порядке возрастания в массиве вместе с новым элементом, а затем строится новое дерево. Для того чтобы найти местоположение нового элемента в массиве, находим, где этот элемент должен быть в БДП, и затем сохраняем в массив элементы до нового и после, а затем удаляем дерево и функцией make tree строим новое.

### **2.4 Удаление элемента из дерева.**

Чтобы удалить элемент из дерева нужно полностью его перестроить, поэтому все элементы дерева сохраняются в массиве в порядке возрастания, параллельно сравниваясь с элементом, который нужно удалить. Если элемент дерева равен элементу для удаления, то в массив он не попадает. Затем удаляем старое дерево и строим новое.

### **3. ОПИСАНИЕ СТРУКТУР ДАННЫХ И ФУНКЦИЙ.**

#### **3.1 Класс Node.**

Для хранения одного элемента дерева написан класс Node.

У класса есть поля:

Node\* right – указатель на правое поддерево.

int data – данные в узле.

int amount – количество повторений элемента.

Node\* left – указатель на левое поддерево.

Для работы с элементом дерева реализованы методы:

Node\* getLeft() – возвращает указатель на левое поддерево.

Node\* getRight() - возвращает указатель правое поддерево.

int getData() const – возвращает значение поля data.

void setLeft(Node\* l) – присваивает значение полю left.

void setRight(Node\* r) - присваивает значение полю right.

void setData(int d) – присваивает значение полю data.

int getAmount() const – возвращает количество повторений.

void incAmount() – увеличивает количество повторений на 1.

#### **3.2 Класс BinaryTree.**

Для хранения дерева реализован класс BinaryTree.

Поля класса:

Node\* tree – указатель на корень дерева.

int maxDepth – высота дерева.

Методы класса:

`explicit BinaryTree(std::vector<int>& arr, string& str)` – конструктор. В конструктор передаётся `arr` – массив типа `int`, строка для записи промежуточных выводов по ссылке `string& str`.

`void printTree(Node* node)` – функция для вывода дерева. В функцию передаётся указатель на элемент дерева `node` типа `Node*`.

`Node* getTree()` – функция возвращает указатель на корень дерева.

`~BinaryTree()` – деструктор класса.

`void findElem(int e, string& str)` – функция для поиска элемента, вызываемая пользователем. В функцию передаётся элемент для поиска `int e`, строка для записи промежуточных выводов по ссылке `string& str`.

`void addElem(int e, string& str)` – функция для добавления элемента, вызываемая пользователем. В функцию передаётся элемент для добавления `int e`, строка для записи промежуточных выводов по ссылке `string& str`.

`void deleteElem(int e, string& str)` – Функция для удаления элемента, вызываемая пользователем. В Функцию передаётся элемент для удаления `int e`, строка для записи промежуточных выводов по ссылке `string& str`.

`int getMaxDepth()` – функция возвращает высоту дерева.

`Node* makeTree(int n, std::vector<int>& arr, int& pos, int indent, string& str)` – функция для создания дерева. В функцию передаются: `int n` количество элементов, `std::vector<int>& arr` массив элементов, `int& pos` – текущая позиция в массиве, `int indent` – сдвиг для вывода промежуточных результатов, `string& str` - строка для записи промежуточных выводов по ссылке.

`void destroy(Node*& buf)` – функция для удаления дерева и отчищения памяти. В функцию передаётся `Node*& buf` – указатель на элемент дерева по ссылке, `string& str` - строка для записи промежуточных выводов по ссылке.

`void treeToArrAdd(Node* node, std::vector<int>& arr, int e, string& str)` – функция для выгрузки элементов из части дерева с новым элементом в массив. В функцию передаются: `Node* node` – указатель на элемент дерева, `std::vector<int>& arr` - массив для сохранения элементов, `int e` – элемент для добавления, `string& str` - строка для записи промежуточных выводов по ссылке.



`void treeToArrDelete(Node* node, std::vector<int>& arr, int e, string& str)` - функция для выгрузки элементов из части дерева с удаляемым элементом в массив. В функцию передаются: `Node* node` – указатель на элемент дерева, `std::vector<int>& arr` - массив для сохранения элементов, `int e` – элемент для добавления, `string& str` - строка для записи промежуточных выводов по ссылке.

`void treeToArrHelp(Node* node, std::vector<int>& arr, string& str)` - функция для выгрузки элементов из части дерева без нового элемента в массив. В функцию передаются: `Node* node` – указатель на элемент дерева, `std::vector<int>& arr` - массив для сохранения элементов, `string& str` - строка для записи промежуточных выводов по ссылке.

`bool find(Node* node, int x, string& str)` – функция для поиска элемента в массиве. В функцию передаются: `Node* node` – указатель на корень дерева, `int x` – элемент для поиска, `string& str` - строка для записи промежуточных выводов по ссылке.

### **3.3 Класс InOut.**

Класс, содержащий методы для считывания данных из файла и консоли.

Методы класса:

`int makeArr(std::vector<int>& arr, string res)`– функция для считывания массива. Функция возвращает код ошибки или 0, если считывание прошло успешно. В функцию передаются `std::vector<int>& arr` - массив целых чисел по ссылке, `string res` – строка с данными.

`int makeArrFile(std::vector<int>& arr, string filename)` – функция для считывания массива из файла. Функция возвращает код ошибки или 0, если считывание прошло успешно. В функцию передаются `std::vector<int>& arr` - массив целых чисел по ссылке, `string filename` – имя файла.

`int getNum(string& input)` – функция для преобразования строки в число. Функция возвращает число. Функция принимает: `string& input` – строку по ссылке.

`void skip (string& str, int n = 1)` – функция для удаления первых `n` элементов строки. В функцию передаются: `string& str` – строка по ссылке, `int n` – количество символов, которые надо удалить.

### **3.4 Класс MyView.**

Класс `MyView` создан для отрисовки дерева в окне приложения.

Методы класса:

`void updateGraphics(Node* tree, int maxdepth)` – функция для обновления картинки. В функцию передаются: `Node* tree` – указатель на корень дерева, `int maxdepth` – высота дерева.

`void drawNode(Node* node, int maxdepth, int depth = 0, int x = 0, int y = 0)` – функция для отрисовки дерева. В Функцию передаются: `Node* tree` – указатель на корень дерева, `int maxdepth` – высота дерева, `int depth` – текущий уровень в дереве, `int x` – положение в картинке по оси `x`, `int y` – положение на картинке по оси `y`.

### **3.5 Класс MainWindow.**

Класс, контролирующий окно приложения, нажатия на кнопки, и отрисовку дерева, и промежуточных выводов.

Поля класса – указатели на экземпляры классов, описанных выше.

`Ui::MainWindow *ui`

`MyView* view`

`INOut* scan`

`BinaryTree* tree`

Методы класса – обработка слотов кнопок.

`void on_open_clicked()` – обработка нажатия на кнопку открыть файл.

`void on_ok_clicked()` – обработка нажатия на кнопку “ok”.

`void on_delete_2_clicked()` – обработка нажатия на кнопку delete.

`void on_add_clicked()` – обработка нажатия на кнопку add.

`void on_find_clicked()` - обработка нажатия на кнопку find.

#### **4. ОПИСАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.**

Для взаимодействия пользователя и программы был разработан GUI.

У пользователя есть выбор: считать данные из файла, нажав кнопку “open”, и выбрав файл, или ввести данные в специальное поле и нажать кнопку “ok”. Затем пользователь выбирает действие, которое он может совершить с деревом, нажав одну из трёх кнопок: add, find, delete. И введя нужный ему элемент.

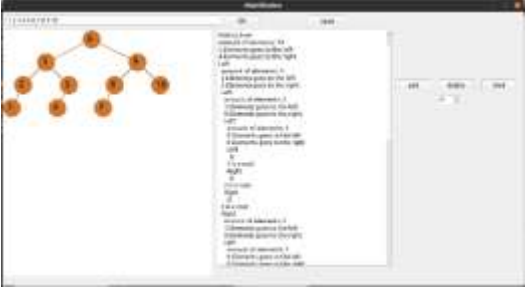
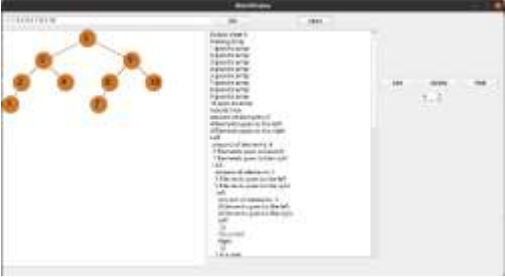
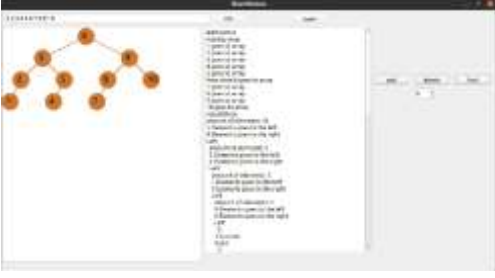


#### **ЗАКЛЮЧЕНИЕ.**

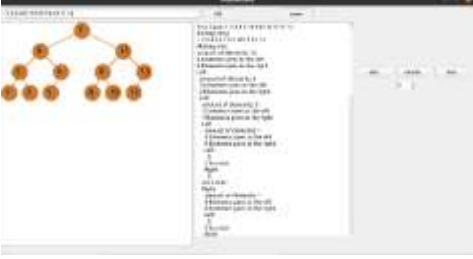
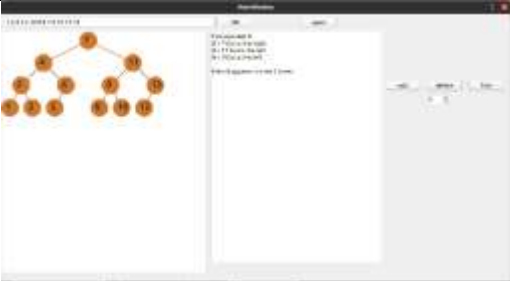
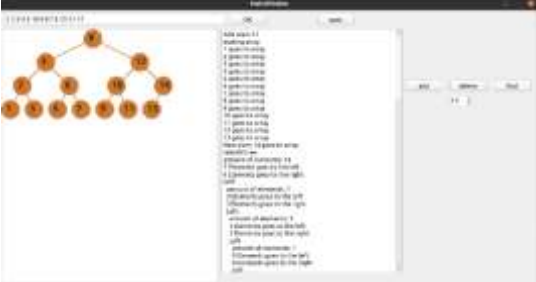
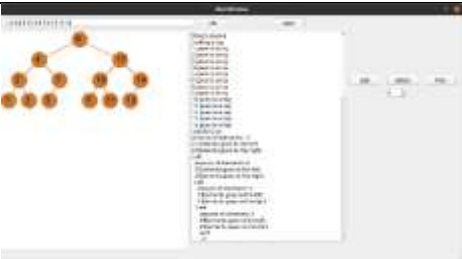
В результате выполнения курсовой работы была написана программа на языке C++, реализующая идеально сбалансированное бинарное дерево поиска. Были продемонстрированы операции вставки, удаления и поиска элементов. Был написан GUI для работы с программой.

## **СПИСОК ЛИТЕРАТУРЫ.**

1. Керниган Б. И Ритчи Д. Язык программирования Си М.: Вильямс, 1978  
288 с.

## ПРИЛОЖЕНИЕ А ТЕСТИРОВАНИЕ.

№	Входные данные	Результат	Комментарии
1	1 2 3 4 5 6 7 8 9 10		На скриншоте продемонстрирован ввод через консоль и построение дерева.
			На скриншоте продемонстрировано удаление элемента 6.
			На скриншоте продемонстрировано добавление элемента 6.
			На скриншоте продемонстрирован поиск элемента 4.
2	test.txt		На скриншоте продемонстрировано открытие файла test.txt

	<p>Содержимое файла: 1 2 3 4 5 10 9 8 7 6 13 11 12</p>		<p>На скриншоте продемонстрировано считывание из файла.</p>
			<p>На скриншоте продемонстрирован поиск элемента 8</p>
			<p>На скриншоте продемонстрировано добавление элемента 14.</p>
			<p>На скриншоте продемонстрировано удаление элемента 6.</p>

## ПРИЛОЖЕНИЕ Б

### ИСХОДНЫЙ КОД ПРОГРАММЫ

#### Файл main.cpp

```
#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

#### Файл node.h

```
#ifndef NODE_H
#define NODE_H

class Node {
private:
    Node* right;
    int data;
    int amount;
    Node* left;
public:
    Node(): right(nullptr), data(0), amount(0), left(nullptr) {}

    Node* getLeft();

    Node* getRight();

    int getData() const;

    int getAmount() const;

    void incAmount();

    void setAmount(int am);

    void setLeft(Node* l);

    void setRight(Node* r);

    void setData(int d);
};

#endif // NODE_H
```

#### Файл node.cpp

```
#include "node.h"

int Node::getData() const {
    return data;
}

Node* Node::getLeft() {
```

```

        return left;
    }

Node* Node::getRight() {
    return right;
}

int Node::getAmount() const {
    return amount;
}

void Node::setData(int d) {
    data = d;
}

void Node::setLeft(Node *l) {
    left = l;
}

void Node::setRight(Node *r) {
    right = r;
}

void Node::setAmount(int am) {
    amount = am;
}

void Node::incAmount() {
    amount++;
}

```

## Файл tree.h

```

#ifndef BINARYTREE_H
#define BINARYTREE_H

#include "node.h"
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <cmath>
using namespace std;

class BinaryTree{
public:
    explicit BinaryTree(std::vector<int>& arr, string& str);
    void printTree(Node* node);
    Node* getTree();
    ~BinaryTree();
    void findElem(int e, string& str);
    void addElem(int e, string& str);
    void deleteElem(int e, string& str);
    int getMaxDepth();
private:
    int maxDepth;
    Node* tree;
    Node* makeTree(int n, std::vector<int>& arr, int& pos, int indent, string&
str);
    void destroy(Node*& buf);
    void treeToArrAdd(Node* node, std::vector<int>& arr, int e, string& str);
    void treeToArrDelete(Node* node, std::vector<int>& arr, int e, string&
str);

```



```

        void treeToArrHelp(Node* node, std::vector<int>& arr, string& str);
        bool find(Node* node, int x, string& str);

};

#endif // BINARYTREE_H

```

## Файл tree.cpp

```

#include "tree.h"

BinaryTree::BinaryTree(std::vector<int> &arr, string& str) {
    str += "Your input: ";
    for(int i : arr){
        str+= std::to_string(i) + ' ';
    }
    str+= '\n';
    str+="Sorting array\n";
    sort(arr.begin(), arr.end());
    for(int i : arr){
        str+= std::to_string(i) + ' ';
    }
    str+= '\n';
    int countUnique = 1;
    int pos = 0;
    for(int i = 0; i < arr.size()-1; i++){
        if(arr[i]<arr[i+1]){
            countUnique++;
        }
    }
    str+="Making tree\n";
    maxDepth = ceil(log2(countUnique+1));
    tree = makeTree(countUnique, arr, pos, 0, str);
}

Node * BinaryTree::getTree() {
    return tree;
}

/*Функция для создания дерева из массива.*/
Node * BinaryTree::makeTree(int n, std::vector<int>& arr, int& pos, int indent,
string& str) {
    if (n == 0) {
        for(int i = 0; i < indent; i++){str+=" ";}
        str+="()\n";
        return nullptr;
    }
    for(int i = 0; i < indent; i++){str+=" ";}
    str+="amount of elements: "+ std::to_string(n)+'\n';
    Node *buf = new Node();
    int nL, nR;
    nL = n / 2;
    for(int i = 0; i < indent; i++){str+=" ";}
    str+= std::to_string(nL) + " Elements goes to the left\n";
    nR = n - nL - 1;
    for(int i = 0; i < indent; i++){str+=" ";}
    str+= std::to_string(nR) + " Elements goes to the right\n";
    for(int i = 0; i < indent; i++){str+=" ";}
    str+="Left\n";
    buf->setLeft(makeTree(nL, arr, pos, indent+1, str));
    for(int i = 0; i < indent; i++){str+=" ";}
    str += std::to_string(arr[pos]) + " is a root\n";
    buf->setData(arr[pos]);
}

```

```

        buf->incAmount();
        while (arr[pos] == arr[pos + 1]) {
            pos++;
            buf->incAmount();
        }
        pos++;
        for(int i = 0; i < indent; i++){str+=" ";}
        str+="Right\n";
        buf->setRight(makeTree(nR, arr, pos, indent+1, str));
        return buf;
    }

void BinaryTree::printTree(Node* node) {
    if(node == nullptr)
        return;
    cout<<node->getData();
    cout<<'(';
    printTree(node->getLeft());
    cout<<')';
    cout<<'(';
    printTree(node->getRight());
    cout<<')';
}

void BinaryTree::destroy(Node*& buf){
    if(buf != nullptr) {
        Node* left = buf->getLeft();
        Node* right = buf->getRight();
        destroy(left);
        destroy(right);
        delete buf;
    }
}

BinaryTree::~BinaryTree() {
    destroy(tree);
}

void BinaryTree::findElem(int e, string& str) {
    str+="find element " + std::to_string(e) + '\n';
    find(tree, e, str);
}

/*Рекурсивная функция для поиска элемента в дереве.
* Если Элемент меньше узла, то она вызывается для левого поддерева. Если
больше, то для правого поддерева
Поиск завершается, когда указатель на дерево равен nullptr или корень дерева
равен элементу.*/
bool BinaryTree::find(Node *node, int x, string& str) {
    if(node == nullptr) {
        str+="\nNo element "+ std::to_string(x) + " in bst\n";
        return false;
    }
    if(x == node->getData()){
        str+="\nElem "+ std::to_string(x)+" appears in tree " +
std::to_string(node->getAmount())+" times\n";
        return true;
    }else if(x < node->getData()){
        str+= std::to_string(x)+" < "+ std::to_string(node->getData())+" Go to
the left\n";
        if(find(node->getLeft(), x, str))
            return true;
    } else if(x > node->getData()) {
        str+= std::to_string(x)+" > "+ std::to_string(node->getData()) + " Go
to the right\n";
    }
}

```

```

        if(find(node->getRight(), x, str))
            return true;
    }
    return false;
}

/*функция для сохранения дерева не содержащего нужный элемент в массив*/
void BinaryTree::treeToArrHelp(Node *node, std::vector<int>& arr, string& str)
{
    if(node != nullptr){
        treeToArrHelp(node->getLeft(), arr, str);
        str+= std::to_string(node->getData())+" goes to array\n";
        for(int i = 0; i<node->getAmount(); i++){
            arr.push_back(node->getData());
        }
        treeToArrHelp(node->getRight(), arr, str);
    }
}

/*функция для сохранения дерева в массив при добавлении элемента*/
void BinaryTree::treeToArrAdd(Node *node, std::vector<int> &arr, int e, string& str) {
    if(node == nullptr){
        arr.push_back(e);
        str+="New elem "+ std::to_string(e)+" goes to array\n";

    } else if(e < node->getData()){
        treeToArrAdd(node->getLeft(), arr, e, str);
        str+= std::to_string(node->getData())+" goes to array\n";
        for(int i = 0; i<node->getAmount(); i++){
            arr.push_back(node->getData());
        }
        treeToArrHelp(node->getRight(), arr, str);
    } else if(e > node->getData()){
        treeToArrHelp(node->getLeft(), arr, str);
        str+= std::to_string(node->getData())+" goes to array\n";
        for(int i = 0; i<node->getAmount(); i++){
            arr.push_back(node->getData());
        }
        treeToArrAdd(node->getRight(), arr, e, str);
    } else if(e == node->getData()){
        treeToArrHelp(node->getLeft(), arr, str);
        node->incAmount();
        str+= std::to_string(node->getData())+" goes to array\n";
        for(int i = 0; i<node->getAmount(); i++){
            arr.push_back(node->getData());
        }
        treeToArrHelp(node->getRight(), arr, str);
    }
}

/*Функция для сохранения дерева в массив при удалении элемента*/
void BinaryTree::treeToArrDelete(Node *node, std::vector<int> &arr, int e, string& str) {
    if(node == nullptr){
        return;
    } else if(e < node->getData()){
        treeToArrDelete(node->getLeft(), arr, e, str);
        str+= std::to_string(node->getData())+" goes to array\n";
        for(int i = 0; i<node->getAmount(); i++){

```

```

        arr.push_back(node->getData());
    }
    treeToArrHelp(node->getRight(), arr, str);

} else if(e > node->getData()){
    treeToArrHelp(node->getLeft(), arr, str);
    str+= std::to_string(node->getData())+" goes to array\n";
    for(int i = 0; i<node->getAmount(); i++){
        arr.push_back(node->getData());
    }
    treeToArrDelete(node->getRight(), arr, e, str);

} else if(e == node->getData()){
    treeToArrHelp(node->getLeft(), arr, str);

    treeToArrHelp(node->getRight(), arr, str);
}
}

/*Удаление элемента и перестройка дерева*/
void BinaryTree::deleteElem(int e, string& str) {
    str+="Delete elem "+ std::to_string(e)+'\n';
    str+="making array\n";
    std::vector<int> arr;
    treeToArrDelete(tree, arr, e, str);
    destroy(tree);
    int countUnique = 1;
    int pos = 0;
    for(int i = 0; i < arr.size()-1; i++){
        if(arr[i]<arr[i+1]){
            countUnique++;
        }
    }
    maxDepth = ceil(log2(countUnique+1));
    str+="rebuild tree\n";
    tree = makeTree(countUnique, arr, pos, 0, str);
}

/*Добавление элемента и престройка дерева*/
void BinaryTree::addElem(int e, string& str) {
    str+="Add elem "+ std::to_string(e)+'\n';
    std::vector<int> arr;
    str+="making array\n";
    treeToArrAdd(tree, arr, e, str);
    destroy(tree);
    int countUnique = 1;
    int pos = 0;
    for(int i = 0; i < arr.size()-1; i++){
        if(arr[i]<arr[i+1]){
            countUnique++;
        }
    }
    maxDepth = ceil(log2(countUnique+1));
    str+="rebuild tree\n";
    tree = makeTree(countUnique, arr, pos, 0, str);
}

int BinaryTree::getMaxDepth(){
    return maxDepth;
}

```

Файл inout.h

```

#ifndef INOUT_H
#define INOUT_H

#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <cstring>
#include <unistd.h>
#include <fstream>

using namespace std;

class INOut
{
public:
    INOut();
    int makeArr(std::vector<int>& arr, string res);
    int makeArrFile(std::vector<int>& arr, string filename);
private:
    int getNum(string& input);
    void skip (string& str, int n = 1);
};

#endif // INOUT_H

```

## Файл inout.cpp

```

#include "inout.h"

INOut::INOut()
{
}

int INOut::makeArr(std::vector<int>& arr, string res){
    while (!res.empty()){
        if(!(isdigit(res[0]) || res[0] == '-')){
            cout<<" error not integer in array\n";
            arr.clear();
            return 1;
        }
        int n = getNum(res);
        arr.push_back(n);
        skip(res,1);
    }
    return 0;
}

int INOut::makeArrFile(std::vector<int>& arr, string filename){
    ifstream infile(filename);
    if (!infile) {
        cout << "> File can't be open!" << endl;
        return 1;
    }
    string str;
    string num;
    getline(infile, str);
    while (!str.empty()) {
        if (!(isdigit(str[0]) || str[0] == '-')) {
            cout << "not integer in array\n";
            arr.clear();
            return 1;
        }
    }
}

```

```

        }
        int n = getNum(str);
        arr.push_back(n);
        skip(str, 1);
    }
    return 0;
}

int INOut::getNum(string& str){
    string strNum;
    while (isdigit(str[0]) || (strNum.length() == 0 && str[0] == '-')) {
        strNum += str[0];
        skip(str, 1);
    }
    return stoi(strNum);
}

void INOut::skip(string& str, int n){
    if (str.length() >= n) {
        str = str.substr(n);
    }
}

```

## Файл myview.h

```

#ifndef MYVIEW_H
#define MYVIEW_H
#include "QGraphicsScene"
#include "QGraphicsView"
#include "tree.h"
#include <cmath>
#include <QGraphicsTextItem>

class MyView: public QGraphicsView
{
    Q_OBJECT
public:
    explicit MyView(QWidget* parent = nullptr);
    void updateGraphics(Node* tree, int maxdepth);
    QGraphicsScene* scene;
private:
    void drawNode(Node* node, int maxdepth, int depth = 0, int x = 0, int y =
0);

    QPen pen;
    QColor color;
    QBrush brush;
    QFont font;
};

#endif // MYVIEW_H

```

## Файл myview.cpp

```

#include "myview.h"

MyView::MyView(QWidget* parent):QGraphicsView (parent)
{
    this->setAlignment(Qt::AlignLeft | Qt::AlignTop);
    scene = new QGraphicsScene();
    QColor color = QColor(203, 119, 47);
    pen.setColor(color);
}

```

```

        brush.setColor(color);
        font.setFamily("Roboto");
        pen.setWidth(3);
    }

void MyView::updateGraphics(Node *tree, int maxdepth){
    scene->clear();
    if(!tree) return;
    drawNode(tree, maxdepth);
    this->setScene(scene);
}
/*Рекурсивная функция для отрисовки дерева*/
void MyView::drawNode(Node* node, int maxdepth, int depth, int x, int y){
    if (node == nullptr) return;
    int offset = pow(2, maxdepth + 3) / pow(2, depth);
    if (node->getLeft()) scene->addLine(x + 32, y + 32, x - offset + 32, y
+ 64 + 32, pen);
    if (node->getRight()) scene->addLine(x + 32, y + 32, x + offset + 32,
y + 64 + 32, pen);
    color.setRgb(203, 119, 47);
    QBrush brush(color);
    QPen pen(color, 3);
    scene->addEllipse(x+8, y+8, 48, 48, pen, brush);
    QGraphicsTextItem *numb = new QGraphicsTextItem();
    numb->setPlainText(QString::number(node->getData()));
    numb->setDefaultTextColor(Qt::black);
    numb->setScale(1.5);
    numb->setPos(x + 14, y + 8);
    scene->addItem(numb);
    drawNode(node->getLeft(), maxdepth, depth + 1, x - offset, y + 64);
    drawNode(node->getRight(), maxdepth, depth + 1, x + offset, y + 64);
}

```

## Файл mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QFileDialog>
#include "tree.h"
#include "inout.h"
#include "myview.h"
#include <vector>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void on_open_clicked();

    void on_ok_clicked();

```

```

        void on_delete_2_clicked();

        void on_add_clicked();

        void on_find_clicked();

private:
    Ui::MainWindow *ui;
    MyView* view;
    INOut* scan;
    BinaryTree* tree;
};
#endif // MAINWINDOW_H

```

## Файл mainwindow.cpp

```

#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    scan = new INOut();
    view = new MyView();
    tree = nullptr;
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_open_clicked()
{
    QString path = QFileDialog::getOpenFileName(nullptr, "Выберете файл",
"/home/user", "*.txt");
    if(path == nullptr)
        return;
    std::vector<int> arr;
    scan->makeArrFile(arr, path.toLocal8Bit().constData());
    if(tree) delete tree;
    std::string str;
    tree = new BinaryTree(arr, str);
    ui->textBrowser->setText(QString::fromStdString(str));
    view->updateGraphics(tree->getTree(), tree->getMaxDepth());
    ui->gridLayout->addWidget(view);
}

void MainWindow::on_ok_clicked()
{
    QString path = ui->lineEdit->text();
    std::vector<int> arr;
    scan->makeArr(arr, path.toLocal8Bit().constData());
    if(tree) delete tree;
    std::string str;
    tree = new BinaryTree(arr, str);
    ui->textBrowser->setText(QString::fromStdString(str));
    view->updateGraphics(tree->getTree(), tree->getMaxDepth());
    ui->gridLayout->addWidget(view);
}

```



```

}

void MainWindow::on_delete_2_clicked()
{
    int el = ui->elem->value();
    std::string str;
    tree->deleteElem(el, str);
    ui->textBrowser->setText(QString::fromStdString(str));
    view->updateGraphics(tree->getTree(), tree->getMaxDepth());
}

void MainWindow::on_add_clicked()
{
    int el = ui->elem->value();
    std::string str;
    tree->addElem(el, str);
    ui->textBrowser->setText(QString::fromStdString(str));
    view->updateGraphics(tree->getTree(), tree->getMaxDepth());
}

void MainWindow::on_find_clicked()
{
    int el = ui->elem->value();
    std::string str;
    tree->findElem(el, str);
    ui->textBrowser->setText(QString::fromStdString(str));
    view->updateGraphics(tree->getTree(), tree->getMaxDepth());
}

```

## Файл mainvindow.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>MainWindow</class>
    <widget class="QMainWindow" name="MainWindow">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>1460</width>
                <height>768</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>MainWindow</string>
        </property>
        <widget class="QWidget" name="centralwidget">
            <widget class="QWidget" name="gridLayoutWidget">
                <property name="geometry">
                    <rect>
                        <x>0</x>
                        <y>50</y>
                        <width>591</width>
                        <height>691</height>
                    </rect>
                </property>
                <layout class="QGridLayout" name="gridLayout"/>
            </widget>
            <widget class="QPushButton" name="open">
                <property name="geometry">
                    <rect>
                        <x>860</x>
                        <y>10</y>

```

```

        <width>101</width>
        <height>29</height>
    </rect>
</property>
<property name="text">
    <string>open</string>
</property>
</widget>
<widget class="QLineEdit" name="lineEdit">
    <property name="geometry">
        <rect>
            <x>12</x>
            <y>10</y>
            <width>601</width>
            <height>28</height>
        </rect>
    </property>
</widget>
<widget class="QPushButton" name="ok">
    <property name="geometry">
        <rect>
            <x>620</x>
            <y>10</y>
            <width>101</width>
            <height>29</height>
        </rect>
    </property>
    <property name="text">
        <string>OK</string>
    </property>
</widget>
<widget class="QPushButton" name="add">
    <property name="geometry">
        <rect>
            <x>1100</x>
            <y>190</y>
            <width>101</width>
            <height>29</height>
        </rect>
    </property>
    <property name="text">
        <string>add</string>
    </property>
</widget>
<widget class="QPushButton" name="delete_2">
    <property name="geometry">
        <rect>
            <x>1220</x>
            <y>190</y>
            <width>101</width>
            <height>29</height>
        </rect>
    </property>
    <property name="text">
        <string>delete</string>
    </property>
</widget>
<widget class="QSpinBox" name="elem">
    <property name="geometry">
        <rect>
            <x>1220</x>
            <y>230</y>
            <width>61</width>

```

```

        <height>29</height>
    </rect>
</property>
<property name="minimum">
    <number>-100</number>
</property>
<property name="maximum">
    <number>100</number>
</property>
</widget>
<widget class="QTextBrowser" name="textBrowser">
    <property name="geometry">
        <rect>
            <x>600</x>
            <y>50</y>
            <width>491</width>
            <height>661</height>
        </rect>
    </property>
</widget>
<widget class="QPushButton" name="find">
    <property name="geometry">
        <rect>
            <x>1340</x>
            <y>190</y>
            <width>101</width>
            <height>29</height>
        </rect>
    </property>
    <property name="text">
        <string>find</string>
    </property>
</widget>
</widget>
<widget class="QMenuBar" name="menubar">
    <property name="geometry">
        <rect>
            <x>0</x>
            <y>0</y>
            <width>1460</width>
            <height>25</height>
        </rect>
    </property>
</widget>
<widget class="QStatusBar" name="statusbar"/>
</widget>
<resources/>
<connections/>
</ui>

```

## Файл aisd\_cw.pro

```

QT      += core gui

greaterThan(QT_MAJOR_VERSION, 4): QT += widgets

CONFIG += c++11

# You can make your code fail to compile if it uses deprecated APIs.
# In order to do so, uncomment the following line.
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all the APIs
deprecated before Qt 6.0.0

```

```

SOURCES += \
    inout.cpp \
    main.cpp \
    mainwindow.cpp \
    myview.cpp \
    node.cpp \
    tree.cpp

HEADERS += \
    inout.h \
    mainwindow.h \
    myview.h \
    node.h \
    tree.h

FORMS += \
    mainwindow.ui

# Default rules for deployment.
qnx: target.path = /tmp/${TARGET}/bin
else: unix:!android: target.path = /opt/${TARGET}/bin
!isEmpty(target.path): INSTALLS += target

```

## Файл aisd\_cw.pro.user

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE QtCreatorProject>
<!-- Written by QtCreator 4.13.2, 2020-12-14T01:00:14. -->
<qtcreator>
  <data>
    <variable>EnvironmentId</variable>
    <value type="QByteArray">{b326802d-e95d-4824-84db-98b52acee183}</value>
  </data>
  <data>
    <variable>ProjectExplorer.Project.ActiveTarget</variable>
    <value type="int">0</value>
  </data>
  <data>
    <variable>ProjectExplorer.Project.EditorSettings</variable>
    <valuemap type="QVariantMap">
      <value type="bool" key="EditorConfiguration.AutoIndent">true</value>
      <value type="bool"
key="EditorConfiguration.AutoSpacesForTabs">false</value>
      <value type="bool"
key="EditorConfiguration.CamelCaseNavigation">true</value>
      <valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.0">
        <value type="QString" key="language">Cpp</value>
        <valuemap type="QVariantMap" key="value">

```

```

    <value type="QByteArray" key="CurrentPreferences">CppGlobal</value>
  </valuemap>
</valuemap>
<valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.1">
  <value type="QString" key="language">QmlJS</value>
  <valuemap type="QVariantMap" key="value">
    <value type="QByteArray" key="CurrentPreferences">QmlJSGlobal</value>
  </valuemap>
</valuemap>
<value type="int" key="EditorConfiguration.CodeStyle.Count">2</value>
<value type="QByteArray" key="EditorConfiguration.Codec">UTF-8</value>
<value type="bool"
key="EditorConfiguration.ConstrainTooltips">>false</value>
<value type="int" key="EditorConfiguration.IndentSize">4</value>
<value type="bool"
key="EditorConfiguration.KeyboardTooltips">>false</value>
<value type="int" key="EditorConfiguration.MarginColumn">80</value>
<value type="bool" key="EditorConfiguration.MouseHiding">true</value>
<value type="bool" key="EditorConfiguration.MouseNavigation">true</value>
<value type="int" key="EditorConfiguration.PaddingMode">1</value>
<value type="bool"
key="EditorConfiguration.ScrollWheelZooming">true</value>
<value type="bool" key="EditorConfiguration.ShowMargin">>false</value>
<value type="int"
key="EditorConfiguration.SmartBackspaceBehavior">0</value>
<value type="bool"
key="EditorConfiguration.SmartSelectionChanging">true</value>
<value type="bool" key="EditorConfiguration.SpacesForTabs">true</value>
<value type="int" key="EditorConfiguration.TabKeyBehavior">0</value>
<value type="int" key="EditorConfiguration.TabSize">8</value>
<value type="bool" key="EditorConfiguration.UseGlobal">true</value>
<value type="int" key="EditorConfiguration.Utf8BomBehavior">1</value>
<value type="bool" key="EditorConfiguration.addFinalNewLine">true</value>
<value type="bool" key="EditorConfiguration.cleanIndentation">true</value>
<value type="bool" key="EditorConfiguration.cleanWhitespace">true</value>
<value type="QString" key="EditorConfiguration.ignoreFileTypes">*.md,
*.MD, Makefile</value>
<value type="bool"
key="EditorConfiguration.inEntireDocument">>false</value>
<value type="bool"
key="EditorConfiguration.skipTrailingWhitespace">true</value>

```

```

    </valuemap>
</data>
<data>
  <variable>ProjectExplorer.Project.PluginSettings</variable>
  <valuemap type="QVariantMap">
    <valuemap type="QVariantMap" key="AutoTest.ActiveFrameworks">
      <value type="bool" key="AutoTest.Framework.Boost">true</value>
      <value type="bool" key="AutoTest.Framework.Catch">true</value>
      <value type="bool" key="AutoTest.Framework.GTest">true</value>
      <value type="bool" key="AutoTest.Framework.QtQuickTest">true</value>
      <value type="bool" key="AutoTest.Framework.QtTest">true</value>
    </valuemap>
    <valuemap type="QVariantMap" key="AutoTest.CheckStates"/>
    <value type="int" key="AutoTest.RunAfterBuild">0</value>
    <value type="bool" key="AutoTest.UseGlobal">true</value>
    <valuelist type="QVariantList" key="ClangCodeModel.CustomCommandLineKey"/>
    <value type="bool" key="ClangCodeModel.UseGlobalConfig">true</value>
    <value type="QString"
key="ClangCodeModel.WarningConfigId">Builtin.Questionable</value>
    <valuemap type="QVariantMap" key="ClangTools">
      <value type="bool" key="ClangTools.BuildBeforeAnalysis">true</value>
      <value type="QString"
key="ClangTools.DiagnosticConfig">Builtin.DefaultTidyAndClazy</value>
      <value type="int" key="ClangTools.ParallelJobs">4</value>
      <valuelist type="QVariantList" key="ClangTools.SelectedDirs"/>
      <valuelist type="QVariantList" key="ClangTools.SelectedFiles"/>
      <valuelist type="QVariantList" key="ClangTools.SuppressedDiagnostics"/>
      <value type="bool" key="ClangTools.UseGlobalSettings">true</value>
    </valuemap>
  </valuemap>
</data>
<data>
  <variable>ProjectExplorer.Project.Target.0</variable>
  <valuemap type="QVariantMap">
    <value type="QString" key="DeviceType">Desktop</value>
    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Desktop Qt
5.15.1 GCC 64bit</value>
  </valuemap>
</data>

```

```

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Desktop Qt 5.15.1 GCC
64bit</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">qt.qt5.5151.gcc_64_kit</value>

    <value type="int"
key="ProjectExplorer.Target.ActiveBuildConfiguration">0</value>

    <value type="int"
key="ProjectExplorer.Target.ActiveDeployConfiguration">0</value>

    <value type="int"
key="ProjectExplorer.Target.ActiveRunConfiguration">0</value>

    <valuemap type="QVariantMap"
key="ProjectExplorer.Target.BuildConfiguration.0">

        <value type="bool">true</value>

        <value type="int" key="EnableQmlDebugging">0</value>

        <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/shakhinn/qt
projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Debug</value>

        <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/shakh
inn/qt projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Debug</value>

        <valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

            <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

                <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

                <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep
</value>

                <value type="QString"
key="QtProjectManager.QMakeBuildStep.QMakeArguments"></value>

                <value type="bool"
key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

                <valuelist type="QVariantList"
key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

            </valuemap>

            <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

                <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

                <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

                <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

                <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">false</value>

                <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments"></value>

```

```

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">2</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Buil
d</value>

</valuemap>

<valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

        <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

        <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

        <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">true</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">1</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clea
n</value>

</valuemap>

    <value type="int"
key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

    <value type="bool"
key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

```



```

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Отладка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfi
guration</value>

    <value type="int"
key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">2</value>

    <value type="int" key="QtQuickCompiler">2</value>

    <value type="int" key="SeparateDebugInfo">2</value>

</valuemap>

<valuemap type="QVariantMap"
key="ProjectExplorer.Target.BuildConfiguration.1">

    <value type="bool">true</value>

    <value type="int" key="EnableQmlDebugging">2</value>

    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/shakhinn/qt
projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Release</value>

    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/shakh
inn/qt projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Release</value>

    <valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

            <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

            <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep
</value>

            <value type="QString"
key="QtProjectManager.QMakeBuildStep.QMakeArguments"></value>

            <value type="bool"
key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

            <valuelist type="QVariantList"
key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

        </valuemap>

        <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

            <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

            <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

            <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

            <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">false</value>

```

```

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments"></value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">2</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Buil
d</value>

</valuemap>

<valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

        <value type="bool" key="ProjectExplorer.BuildStep.Enabled">>true</value>

        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

    <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

        <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">true</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">1</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clea
n</value>

</valuemap>

    <value type="int"
key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

```

```

    <value type="bool"
key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Выпуск</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConf
iguration</value>

    <value type="int"
key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>

    <value type="int" key="QtQuickCompiler">0</value>

    <value type="int" key="SeparateDebugInfo">2</value>

</valuemap>

    <valuemap type="QVariantMap"
key="ProjectExplorer.Target.BuildConfiguration.2">

    <value type="bool">true</value>

    <value type="int" key="EnableQmlDebugging">0</value>

    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/shakhinn/qt
projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Profile</value>

    <value type="QString"
key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/shakh
inn/qt projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Profile</value>

    <valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep
</value>

    <value type="QString"
key="QtProjectManager.QMakeBuildStep.QMakeArguments"></value>

    <value type="bool"
key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

    <valuelist type="QVariantList"
key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

</valuemap>

    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

    <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

```

```

        <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

        <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">false</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments"></value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">2</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Сборка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Buil
d</value>

</valuemap>

<valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

    <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

        <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

        <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</val
ue>

        <valuelist type="QVariantList"
key="Qt4ProjectManager.MakeStep.BuildTargets"/>

        <value type="bool" key="Qt4ProjectManager.MakeStep.Clean">true</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

        <value type="QString"
key="Qt4ProjectManager.MakeStep.MakeCommand"></value>

        <value type="bool"
key="Qt4ProjectManager.MakeStep.OverrideMakeflags">false</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">1</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Очистка</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clea
n</value>

```

```

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

    <value type="bool"
key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

    <valuelist type="QVariantList"
key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Профилирование</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfi
guration</value>

    <value type="int"
key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>

    <value type="int" key="QtQuickCompiler">0</value>

    <value type="int" key="SeparateDebugInfo">0</value>

</valuemap>

    <value type="int"
key="ProjectExplorer.Target.BuildConfigurationCount">3</value>

    <valuemap type="QVariantMap"
key="ProjectExplorer.Target.DeployConfiguration.0">

    <valuemap type="QVariantMap"
key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

    <value type="int"
key="ProjectExplorer.BuildStepList.StepsCount">0</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Развёртывание</
value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.DisplayName">Развёртывание</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Deplo
y</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.BuildConfiguration.BuildStepListCount">1</value>

    <valuemap type="QVariantMap"
key="ProjectExplorer.DeployConfiguration.CustomData"/>

    <value type="bool"
key="ProjectExplorer.DeployConfiguration.CustomDataEnabled">false</value>

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.DefaultDeployCo
nfiguration</value>

</valuemap>

```

```

    <value type="int"
key="ProjectExplorer.Target.DeployConfigurationCount">1</value>

    <valuemap type="QVariantMap" key="ProjectExplorer.Target.PluginSettings"/>

    <valuemap type="QVariantMap"
key="ProjectExplorer.Target.RunConfiguration.0">
    <value type="QString" key="Analyzer.Perf.CallgraphMode">dwarf</value>
    <valuelist type="QVariantList" key="Analyzer.Perf.Events">
        <value type="QString">cpu-cycles</value>
    </valuelist>
    <valuelist type="QVariantList" key="Analyzer.Perf.ExtraArguments"/>
    <value type="int" key="Analyzer.Perf.Frequency">250</value>
    <valuelist type="QVariantList" key="Analyzer.Perf.RecordArguments">
        <value type="QString">-e</value>
        <value type="QString">cpu-cycles</value>
        <value type="QString">--call-graph</value>
        <value type="QString">dwarf,4096</value>
        <value type="QString">-F</value>
        <value type="QString">250</value>
    </valuelist>
    <value type="QString" key="Analyzer.Perf.SampleMode">-F</value>
    <value type="bool"
key="Analyzer.Perf.Settings.UseGlobalSettings">true</value>
    <value type="int" key="Analyzer.Perf.StackSize">4096</value>
    <value type="bool"
key="Analyzer.QmlProfiler.AggregateTraces">false</value>
    <value type="bool" key="Analyzer.QmlProfiler.FlushEnabled">false</value>
    <value type="uint" key="Analyzer.QmlProfiler.FlushInterval">1000</value>
    <value type="QString" key="Analyzer.QmlProfiler.LastTraceFile"></value>
    <value type="bool"
key="Analyzer.QmlProfiler.Settings.UseGlobalSettings">true</value>
    <valuelist type="QVariantList"
key="Analyzer.Valgrind.AddedSuppressionFiles"/>
    <value type="bool"
key="Analyzer.Valgrind.Callgrind.CollectBusEvents">false</value>
    <value type="bool"
key="Analyzer.Valgrind.Callgrind.CollectSystime">false</value>
    <value type="bool"
key="Analyzer.Valgrind.Callgrind.EnableBranchSim">false</value>
    <value type="bool"
key="Analyzer.Valgrind.Callgrind.EnableCacheSim">false</value>
    <value type="bool"
key="Analyzer.Valgrind.Callgrind.EnableEventToolTips">true</value>

```

```

    <value type="double"
key="Analyzer.Valgrind.Callgrind.MinimumCostRatio">0.01</value>

    <value type="double"
key="Analyzer.Valgrind.Callgrind.VisualisationMinimumCostRatio">10</value>

    <value type="bool"
key="Analyzer.Valgrind.FilterExternalIssues">true</value>

    <value type="QString"
key="Analyzer.Valgrind.KCachegrindExecutable">kcachegrind</value>

    <value type="int" key="Analyzer.Valgrind.LeakCheckOnFinish">1</value>

    <value type="int" key="Analyzer.Valgrind.NumCallers">25</value>

    <valuelist type="QVariantList"
key="Analyzer.Valgrind.RemovedSuppressionFiles"/>

    <value type="int"
key="Analyzer.Valgrind.SelfModifyingCodeDetection">1</value>

    <value type="bool"
key="Analyzer.Valgrind.Settings.UseGlobalSettings">true</value>

    <value type="bool" key="Analyzer.Valgrind.ShowReachable">false</value>

    <value type="bool" key="Analyzer.Valgrind.TrackOrigins">true</value>

    <value type="QString"
key="Analyzer.Valgrind.ValgrindExecutable">valgrind</value>

    <valuelist type="QVariantList" key="Analyzer.Valgrind.VisibleErrorKinds">
        <value type="int">0</value>
        <value type="int">1</value>
        <value type="int">2</value>
        <value type="int">3</value>
        <value type="int">4</value>
        <value type="int">5</value>
        <value type="int">6</value>
        <value type="int">7</value>
        <value type="int">8</value>
        <value type="int">9</value>
        <value type="int">10</value>
        <value type="int">11</value>
        <value type="int">12</value>
        <value type="int">13</value>
        <value type="int">14</value>
    </valuelist>

    <valuelist type="QVariantList" key="CustomOutputParsers"/>

    <value type="int" key="PE.EnvironmentAspect.Base">2</value>

    <valuelist type="QVariantList" key="PE.EnvironmentAspect.Changes"/>

```

```

    <value type="QString"
key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4RunConfigu
ration:/home/shakhinn/qt_projects/aisd_cw/aisd_cw.pro</value>

    <value type="QString"
key="ProjectExplorer.RunConfiguration.BuildKey">/home/shakhinn/qt
projects/aisd_cw/aisd_cw.pro</value>

    <value type="QString" key="RunConfiguration.Arguments"></value>

    <value type="bool" key="RunConfiguration.Arguments.multi">false</value>

    <value type="QString"
key="RunConfiguration.OverrideDebuggerStartup"></value>

    <value type="bool" key="RunConfiguration.UseCppDebugger">false</value>

    <value type="bool" key="RunConfiguration.UseCppDebuggerAuto">true</value>

    <value type="bool"
key="RunConfiguration.UseLibrarySearchPath">true</value>

    <value type="bool" key="RunConfiguration.UseMultiProcess">false</value>

    <value type="bool" key="RunConfiguration.UseQmlDebugger">false</value>

    <value type="bool" key="RunConfiguration.UseQmlDebuggerAuto">true</value>

    <value type="QString" key="RunConfiguration.WorkingDirectory"></value>

    <value type="QString"
key="RunConfiguration.WorkingDirectory.default">/home/shakhinn/qt
projects/build-aisd_cw-Desktop_Qt_5_15_1_GCC_64bit-Debug</value>

    </valuemap>

    <value type="int"
key="ProjectExplorer.Target.RunConfigurationCount">1</value>

    </valuemap>

</data>

<data>

    <variable>ProjectExplorer.Project.TargetCount</variable>

    <value type="int">1</value>

</data>

<data>

    <variable>ProjectExplorer.Project.Updater.FileVersion</variable>

    <value type="int">22</value>

</data>

<data>

    <variable>Version</variable>

    <value type="int">22</value>

</data>

</qtcreeator>

```