Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

**ОТЧЕТ**

**Тема:** Лабораторная работа по бинарным деревьям

Семестр: 2

Выполнил студент ИВТ-22-2б:

Мельников Глеб Владимирович

(дата, подпись)

Проверила:

Полякова Ольга Андреевна

(дата, подпись)

Пермь 2023

ПО:

Для выполнение лабораторной работы выбран фреймворк Qt.

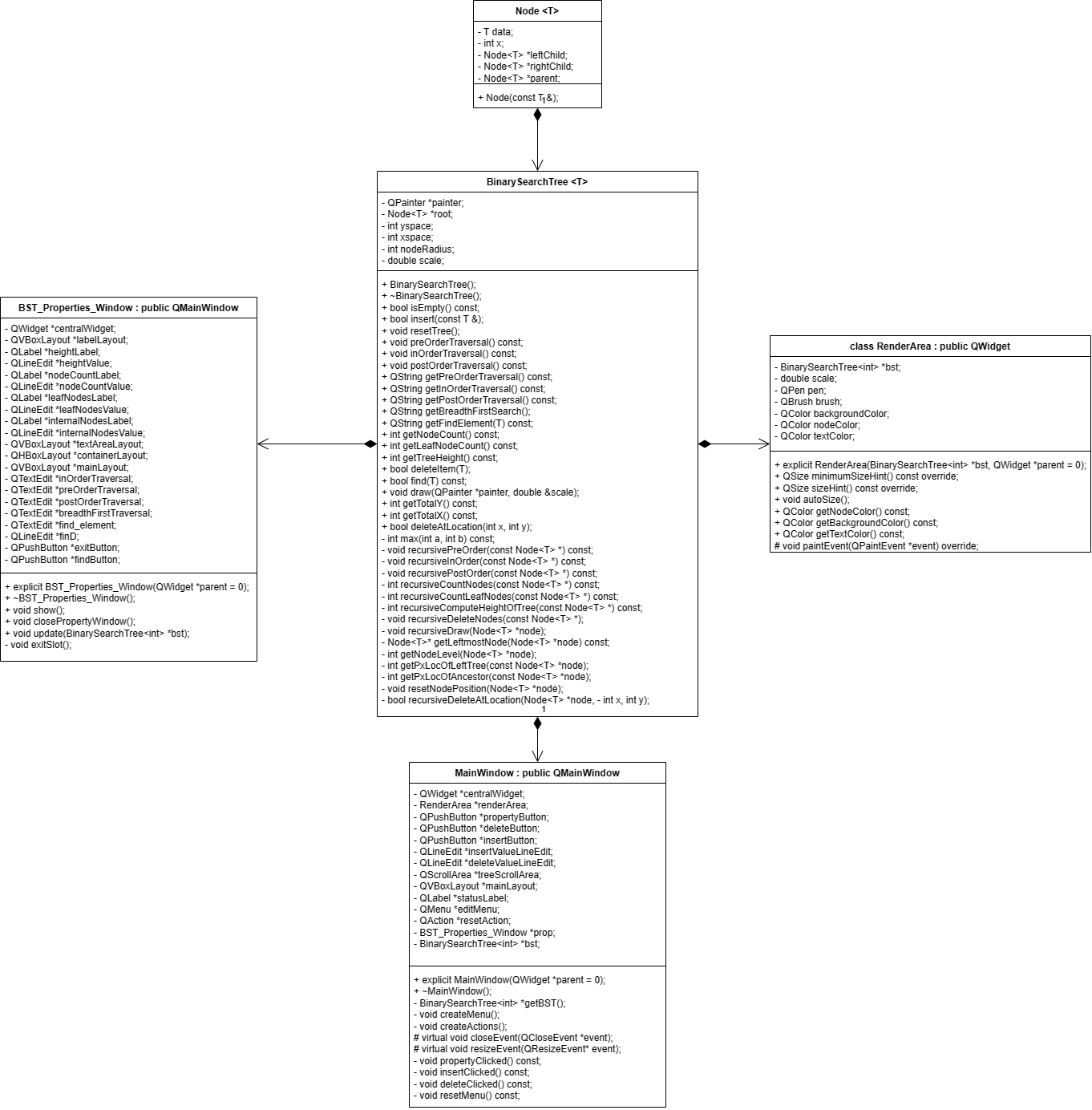
Классы:

Класс Node отвечает за объект узела. Хранение данных о правом ребёнке, левом и родителе.

Класс BinarySearchTree отвечает за функции работы с деревьями. И отрисовки данных.

Класс BST\_Properties\_Window отвечает за свойства окна класса. BinarySearchTree.

Класс RenderArea отвечает за задание параметров отрисовки.

UML-диаграмма: 

Kод программы:

BST\_Properties\_Window::**BST\_Properties\_Window**(QWidget \*parent) : QMainWindow(*parent*)

{

this->setWindowTitle("Свойства");

this->setFixedSize(QSize(400, 650));

labelLayout = new QVBoxLayout;

heightLabel = new QLabel("Высота:");

nodeCountLabel = new QLabel("Количество узлов:");

leafNodesLabel = new QLabel("Количество \"листьев\":");

internalNodesLabel = new QLabel("Внутренние узлы:");

heightLabel->setStyleSheet("font-size: 14px;");

nodeCountLabel->setStyleSheet("font-size: 14px;");

leafNodesLabel->setStyleSheet("font-size: 14px;");

internalNodesLabel->setStyleSheet("font-size: 14px;");

labelLayout->addWidget(*heightLabel*);

labelLayout->addWidget(*nodeCountLabel*);

labelLayout->addWidget(*leafNodesLabel*);

labelLayout->addWidget(*internalNodesLabel*);

textAreaLayout = new QVBoxLayout;

heightValue = new QLineEdit("");

nodeCountValue = new QLineEdit("");

leafNodesValue = new QLineEdit("");

internalNodesValue = new QLineEdit("");

heightValue->setReadOnly(true);

nodeCountValue->setReadOnly(true);

leafNodesValue->setReadOnly(true);

internalNodesValue->setReadOnly(true);

heightValue->setStyleSheet("color: black; width: 100px;"

"padding-left: 10px; font-size: 14px;");

nodeCountValue->setStyleSheet("color: black; width: 100px;"

"padding-left: 10px; font-size: 14px;");

leafNodesValue->setStyleSheet("color: black; width: 100px;"

"padding-left: 10px; font-size: 14px;");

internalNodesValue->setStyleSheet("color: black; width: 100px;"

"padding-left: 10px; font-size: 14px;");

textAreaLayout->addWidget(*heightValue*);

textAreaLayout->addWidget(*nodeCountValue*);

textAreaLayout->addWidget(*leafNodesValue*);

textAreaLayout->addWidget(*internalNodesValue*);

containerLayout = new QHBoxLayout;

containerLayout->addLayout(*labelLayout*);

containerLayout->addLayout(*textAreaLayout*);

preOrderTraversal = new QTextEdit;

inOrderTraversal = new QTextEdit;

postOrderTraversal = new QTextEdit;

breadthFirstTraversal = new QTextEdit;

find\_element=new QTextEdit;

finD=new QLineEdit;

preOrderTraversal->setReadOnly(true);

inOrderTraversal->setReadOnly(true);

postOrderTraversal->setReadOnly(true);

breadthFirstTraversal->setReadOnly(true);

find\_element->setReadOnly(true);

exitButton = new QPushButton("В&ыход", this);

connect(exitButton, SIGNAL(clicked()), this, SLOT(exitSlot()));

findButton=new QPushButton("П&оиск", this);

QHBoxLayout \*buttonLayout = new QHBoxLayout;

buttonLayout->addStretch(0);

buttonLayout->addWidget(*exitButton*);

buttonLayout->addStretch(3);

buttonLayout->addWidget(*findButton*);

buttonLayout->addWidget(*finD*);

mainLayout = new QVBoxLayout;

mainLayout->addLayout(*containerLayout*);

mainLayout->addWidget(new QLabel*(*"Прямой обход:"*)*);

mainLayout->addWidget(*preOrderTraversal*);

mainLayout->addWidget(new QLabel*(*"Симметричный обход:"*)*);

mainLayout->addWidget(*inOrderTraversal*);

mainLayout->addWidget(new QLabel*(*"Обратный обход:"*)*);

mainLayout->addWidget(*postOrderTraversal*);

mainLayout->addWidget(new QLabel*(*"Обход в ширину"*)*);

mainLayout->addWidget(*breadthFirstTraversal*);

mainLayout->addWidget(new QLabel*(*"Поиск элемента по ключу"*)*);

mainLayout->addWidget(*find\_element*);

mainLayout->addLayout(*buttonLayout*);

centralWidget = new QWidget(this);

centralWidget->setLayout(*mainLayout*);

this->setCentralWidget(*centralWidget*);

}