Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе «Деревья»**

Выполнил работу

студент группы ИВТ-22-2б

Коняев Александр Сергеевич

Проверила

Доцент кафедры ИТАС Полякова О.А.

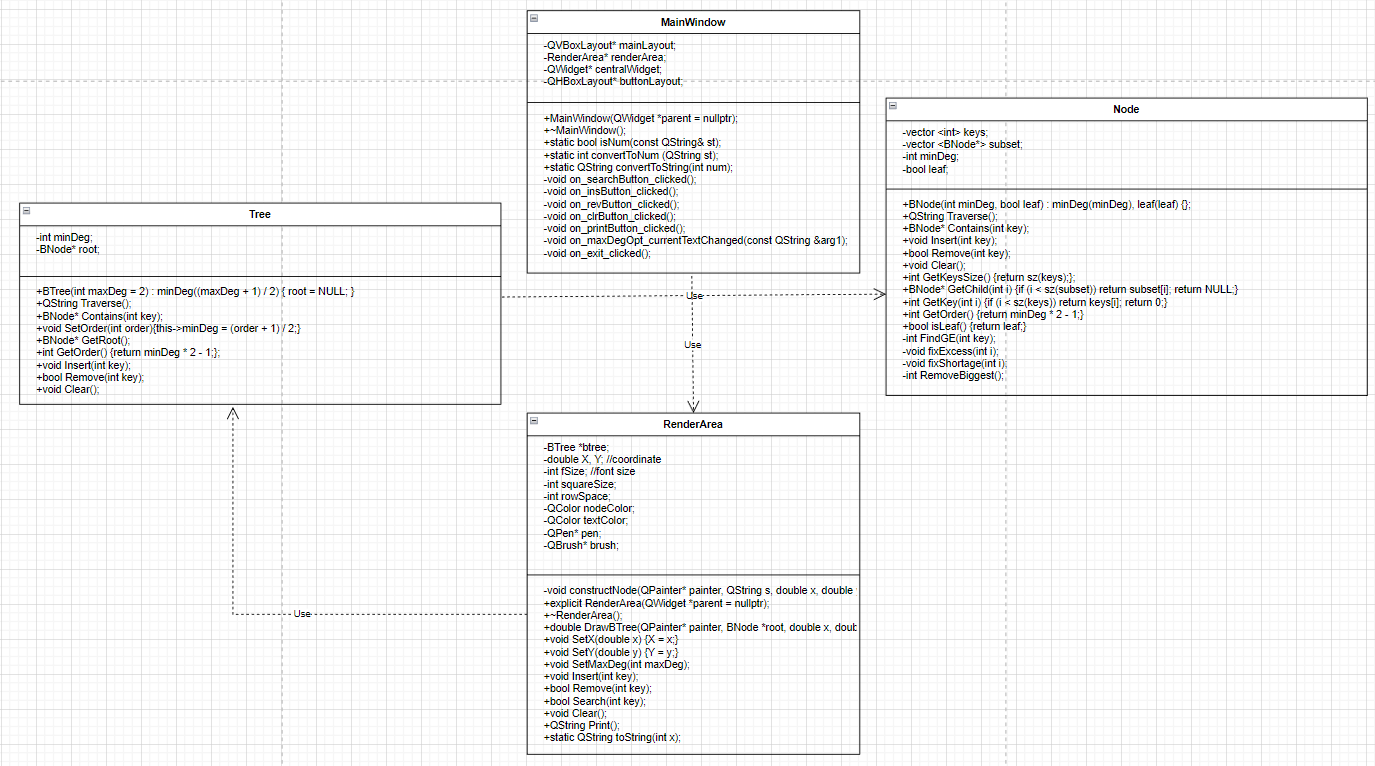
Пермь, 2023

**Постановка задачи**

## Постановка задачи:

Требуется реализовать алгоритмы для собственного варианта бинарного дерева поиска, имеющего не менее трёх уровней .  
Алгоритмы:  
1. Необходимо реализовать функции для редактирования дерева:  
- Вставка узла.  
- Удаление узла.  
- Поиск элемента по ключу.  
2. Реализовать алгоритмы обхода дерева:  
2.1 Прямой  
2.2 Симметричный  
2.3 Обратный  
3. Реализовать алгоритм балансировки дерева.  
4. Реализовать вертикальную и горизонтальную печать.  
5. Визуализация дерева должна быть выполнена с использованием любой доступной графической библиотеки – SFML, SDL, OpenGL и подобных.  
6. Пользовательский интерфейс по усмотрению разработчика - с условием кроссплатформенности (поощряется использование Qt или иных фреймворков).

**UML - диаграмма**

****

**Код**

QString BTree::Traverse()

{

if (root != NULL) return root->Traverse();

return "";

}

BNode\* BTree::Contains(int key)

{

if (!root) return NULL;

return root->Contains(key);

}

void BTree::Insert(int key)

{

if (!root)

{

root = new BNode(minDeg, 1);

root->keys.pb(key);

return;

}

root->Insert(key);

if (sz(root->keys) == 2 \* minDeg - 1)

{

BNode\* nroot = new BNode(minDeg, 0);

nroot->subset.pb(root);

nroot->fixExcess(0);

root = nroot;

}

}

bool BTree::Remove(int key)

{

if (!root)

{

return false;

}

bool removed = root->Remove(key);

if (!removed) return false;

if (!sz(root->keys))

{

BNode\* tmp = root;

root = (root->leaf ? NULL : root->subset[0]);

delete tmp;

}

return true;

}

void BTree::Clear()

{

if (root == NULL) return;

root->Clear();

root = NULL;

}

BNode\* BTree::GetRoot()

{

return root;

}