



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ

Направление 02.03.03 «Математическое обеспечение и
администрирование информационных систем»

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ «В-ДЕРЕВО ПОРЯДКА 2»
ПО ДИСЦИПЛИНЕ «СТРУКТУРЫ И АЛГОРИТМЫ КОМПЬЮТЕРНОЙ
ОБРАБОТКИ ДАННЫХ»

Выполнил студент группы Б9119-02.03.03техпро
Марков А.В. _____

Проверил к.т.н., доцент кафедры ПММУ и ПО
Остроухова С.Н. _____

Содержание

1	Описание	2
2	Спецификация	3
2.1	Поля класса BTree	3
2.2	Методы класса BTree	3
3	Тестирование	4

1 Описание

Цель лабораторной работы: разработать собственный класс **BTree** для работы с B-деревьями порядка 2 со следующим необходимым пакетом методов:

1. конструктор;
2. деструктор;
3. добавление;
4. удаление;
5. поиск;
6. печать.

2 Спецификация

2.1 Поля класса BTree

1. `root` – ссылка на корень дерева типа `Node<data>*`, где:

- `Node` – структура узла дерева;
- `data` – тип данных шаблона дерева.

2.2 Методы класса BTree

1. Конструктор

`BTree::BTree()` – инициализирует `root` значением `nullptr`.

2. Деструктор

`BTree::~~BTree()` – рекурсивно очищает выделенную память, начиная с `root`.

3. Добавление

`bool BTree::add(data key)` – добавляет элемент шаблонного типа в структуру дерева. Если такой же элемент уже находится в дереве, то элемент не добавляется, и в таком случае возвращается `false`. В ином случае возвращается `true`.

4. Удаление

`bool BTree::remove(data key)` – удаляет элемент шаблонного типа из структуры дерева. Если такого элемента не существует в дереве, то элемент не удаляется, и в таком случае возвращается `false`. В ином случае возвращается `true`.

5. Поиск

`bool BTree::find(data key)` – рекурсивно ищет по всем узлам дерева элемент с таким же значением. Если такой элемент не существует в дереве, то возвращается `false`. В ином случае возвращается `true`.

6. Печать

`void BTree::print()` – выводит в консоль визуальное представление дерева (корень отрисовывается слева, а листья дерева – справа). Если дерево пусто, то в консоль выводится следующее сообщение: «B-Tree is empty».

3 Тестирование

Таблица 1: Тестирование конструктора

Тестовая ситуация	Входные данные	Выходные данные
Создание пустого дерева	BTree btree	<pre>btree: { root = nullptr; }</pre>

Таблица 2: Тестирование деструктора

Тестовая ситуация	Входные данные	Выходные данные
Удаление пустого дерева	<pre>btree: { root = nullptr; }</pre>	<pre>btree: { root = nullptr; }</pre>
Удаление непустого дерева	<pre>btree: { root = 0x00000001; }</pre>	<pre>btree: { root = nullptr; }</pre>

Таблица 3: Тестирование добавления

Тестовая ситуация	Входные данные	Выходные данные
Добавление в пустое дерево	<pre>btree: B-tree is empty key: 1</pre>	<pre>btree: 1 return: true</pre>
Добавление в непустое дерево	<pre>btree: 1 key: 2</pre>	<pre>btree: 2 1 return: true</pre>

Таблица 3: Тестирование добавления

Тестовая ситуация	Входные данные	Выходные данные
Перепополнение в корневом узле	<pre> btree: 3 2 1 key: 4 </pre>	<pre> btree: 3 4 2 1 return: true </pre>
Перепополнение на 2-м уровне	<pre> btree: 7 8 6 5 4 3 2 1 key: 9 </pre>	<pre> btree: 7 8 9 6 5 4 3 2 1 return: true </pre>
Добавление существующего ключа	<pre> btree: 1 key: 1 </pre>	<pre> btree: 1 return: false </pre>

Таблица 4: Тестирование удаления

Тестовая ситуация	Входные данные	Выходные данные
Удаление, не приводящее к перебалансировке	<pre> btree: 16 19 15 11 10 3 5 2 0 1 key: 5 </pre>	<pre> btree: 16 19 15 11 10 3 2 0 1 return: true </pre>

Таблица 4: Тестирование удаления

Тестовая ситуация	Входные данные	Выходные данные
Удаление, приводящее к ребалансировке	<pre> btree: 16 24 15 11 14 10 3 7 2 1 key: 14 </pre>	<pre> btree: 16 24 15 11 10 3 7 2 1 return: true </pre>
Удаление последнего элемента	<pre> btree: 10 key: 10 </pre>	<pre> btree: B-tree is empty return: true </pre>
Удаление несуществующего элемента	<pre> btree: 16 24 15 11 14 10 3 7 2 1 key: 8 </pre>	<pre> btree: 16 24 15 11 14 10 3 7 2 1 return: false </pre>

Таблица 5: Тестирование поиска

Тестовая ситуация	Входные данные	Выходные данные
Поиск минимального элемента в дереве	<pre> btree: 16 19 15 11 10 3 5 2 0 1 key: 0 </pre>	<pre> return: true </pre>

Таблица 5: Тестирование поиска

Тестовая ситуация	Входные данные	Выходные данные
Поиск максимального элемента в дереве	<pre> btree: 16 19 15 11 10 3 5 2 0 1 key: 19 </pre>	<pre> return: true </pre>
Поиск не минимального и не максимального элемента в дереве	<pre> btree: 16 19 15 11 10 3 5 2 0 1 key: 15 </pre>	<pre> return: true </pre>
Поиск несуществующего элемента в дереве	<pre> btree: 16 19 15 11 10 3 5 2 0 1 key: 53 </pre>	<pre> return: false </pre>

Таблица 6: Тестирование печати

Тестовая ситуация	Входные данные	Выходные данные
Печать пустого дерева	<pre> btree: { root = nullptr; } </pre>	<pre> out: B-tree is empty </pre>

Таблица 6: Тестирование печати

Тестовая ситуация	Входные данные	Выходные данные
Печать одного элемента	<pre>btree: 10</pre>	<pre>out: 10</pre>
Печать нескольких элементов в корневом узле	<pre>btree: 3 2 1</pre>	<pre>out: 1 2 3</pre>
Печать 2-го уровня	<pre>btree: 7 8 6 5 4 3 2 1</pre>	<pre>out: 2 4 6 1 3 5 7 8</pre>
Печать 3-го уровня	<pre>btree: 16 24 15 11 14 10 3 7 2 1</pre>	<pre>out: 10 2 1 3 7 15 11 14 16 24</pre>
Печать пустого дерева	<pre>btree: { root = nullptr; }</pre>	<pre>out: B-tree is empty</pre>