

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

ШАБАНОВ ИВАН ВЛАДИСЛАВОВИЧ

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДОВ ПОСТРОЕНИЯ СПРАВОЧНИКОВ ДЛЯ АВТОМОБИЛЕЙ

**КУРСОВОЙ ПРОЕКТ**

по дисциплине «Фундаментальные структуры данных и алгоритмы»

по образовательной программе подготовки бакалавров по направлению

09.03.04 - Программная инженерия

Студент группы Б8217 \_\_\_\_\_\_\_\_\_ Шабанов И.В.

(подпись)

Руководитель

ст. преподаватель \_\_\_\_\_\_\_\_\_ Крестникова О.А.

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

(подпись)

Защищен оценкой \_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) Фамилия И.О.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

г. Владивосток

2019

Оглавление

Оглавление

[Введение 3](#_Toc11617131)

[1 Анализ предметной области 4](#_Toc11617132)

[1.1 Модель предметной области 4](#_Toc11617133)

[2 Теоретическая часть 5](#_Toc11617134)

[2.1 АВЛ дерево 5](#_Toc11617135)

[2.1.1 Добавление, поиск и удаление 5](#_Toc11617136)

[2.1.2 Перебалансировка 6](#_Toc11617137)

# Введение

На сегодняшний день, в мире существует огромное количество автомобилей. И, также, существуют множество задач, где надо хранить большие объемы данных о них. Например, надо создать справочник угнанных автомобилей, чтобы определенные органы могли быстро узнать находится ли какой-либо автомобиль в угоне или нет. Для этого нам надо определить, какую структуру данных лучше всего использовать, для того чтобы поиск, удаление и добавление в базу данных были максимально эффективны.

Целью курсового проекта является: разработка информационной системы для работы со справочником автомобилей и экспериментальное исследование методов построения справочников.

Исходя из поставленной цели, в работе были определены следующие задачи:

* Провести анализ предметной области автомобилей и построить модель.
* Изучить теоретические основы методов построения справочников.
* Разработать проект информационной системы автомобилей.
* Реализовать и провести тестирование информационной системы автомобилей.
* Провести эксперименты и проанализировать их результаты.

# 1 Анализ предметной области

## 1.1 Модель предметной области

Предметная область – справочник автомобилей.

Профессионал предметной области – сотрудники внутренних органов.

Профессиональная задача – поиск, добавление и удаление угнанных автомобилей в справочнике автомобилей.

Объект **автомобиль** – структура, которая состоит из атрибутов: **номер кузова,** **год выпуска**, **марка**.

Атрибут **номер кузова** – произвольная последовательность символов, количество которых равно 17, первые 13 символов могут состоять из чисел и заглавных букв английского алфавита кроме {I, O, Q}, последние 4 символа могут состоять только из чисел.

Атрибут **год выпуска** – число, которое находится в интервале от 1850 до 2019.

Атрибут **марка** – произвольная последовательность символов, количество которых не превышает 50.

Пример входных данных:

V67XHPRW77F6D1272 1924 Mercury

MKRLV6B08BV3U2807 1985 Ravon

4JDZSYWE6G8C34237 1918 MG

2BCVYWCFCFE2S9008 1916 GMC

ETUL7ZRN5LKC36903 1999 Iveco

GNGFK9T1P6HUG3071 1980 Dacia

YR1C15KUZ9U3B3537 1921 Daewoo

UK1KBSTWEGE2R4542 1996 Bugatti

J496XM4XZLXY07212 1996 Genesis

9U95DXSN4DRNE3335 1962 Lotus

# 2 Теоретическая часть

Для хранения данных об автомобилях нужна динамическая структура, т.к. у нас должны быть реализованы функции добавления удаления того или иного автомобиля в справочнике. Для этих целей я выбрал две структуры: АВЛ дерево и Хеш-таблицу.

## 2.1 АВЛ дерево

АВЛ дерево – это двоичное дерево поиска, которое сбалансировано таким образом, что у каждой вершины высота ее двух сыновей не отличается более чем на 1.

Двоичное дерево поиска – это дерево поиска, у которого помимо того, что каждый узел может иметь не больше двух сыновей (поддеревьев), выполняются следующие свойства:

* Поддеревья всякого узла так же являются двоичным деревом поиска.
* У всякого узла любое значение ключа из его левого поддерева меньше значения ключа его самого, а любое значение правого поддерева всегда больше (либо равно, но в нашем случае такого не будет) его самого.

Всякий узел АВЛ дерева содержит ключ, значение и высоту поддерева начиная с этого узла.

Сложность операций удаления, добавления и поиска всегда равняются 0(log n), где n – это количество элементов.

### 2.1.1 Операции с деревом

Для того чтобы добавить новый узел в АВЛ дерево нужно найти для него место и вставить его туда. Место ищется путем сравнивания ключей, начиная с корня: если ключ добавляемого узла больше сравниваемого, то идем в правое поддерево, иначе в левое. Затем нужно заново посчитать высоту узлов выше добавленного. Высота дерева равняется наибольшей высоте одного из его поддеревьев +1. Если дерево стало несбалансированным, выполнить перебалансировку.

При поиске узла мы так же сравниваем ключи до тех пор, пока не наткнемся на искомый узел.

При удалении сначала происходит рекурсивный поиск узла, который следует удалить, для того чтобы вы время выхода из рекурсии мы сразу выполняли перебалансировку если она требуется.

### 2.1.2 Балансировка

Как уже говорилось ранее, при добавлении или удалении узла у нас может нарушится свойство АВЛ дерева, что у вершины высота ее сыновей не должна отличаться более чем на 1, в противном случае нам нужно сделать балансировку.

Балансировка осуществляется путем вращения поддеревьев. Существует всего 4 случая, когда нам может потребоваться балансировка и соответственно 4 вида вращения.

1 случай: высота (B) = высоте (A) + 2, и высота (С) ≤ высоте (B) (рис. 1).  
У узла (x) кидаем ссылку его правого поддерева на узел (C), а у узла (y) кидаем ссылку его левого поддерева на узел (x), таким образом высота (A) и (B) сравнится, а высота (C) останется прежней, то есть либо будет равняться высоте (A) и (B), либо на 1 единицу будет выше.

y

А

C

B

x

y

А

C

B

x

Рис.1 – Малое левое вращение

2 случай: высота (A) = высота (B) + 2, и высота (С) ≤ высоты (A) (рис. 2).  
Аналогично, у узла (y) кидаем ссылку его левого поддерева на узел (C), а у узла (x) кидаем ссылку его правого поддерева на узел (y). Малое правое вращение полностью симметрично малому левому.

y

А

C

B

x

y

А

C

B

x

Рис.2 – Малое правое вращение