

数据结构大作业报告

红黑树

姚皓天 (2013011515)

2014 年 12 月

目录

1	基本原理	3
1.1	概述	3
1.2	原理	3
1.3	界面	3
1.4	操作说明	3
1.4.1	文件操作	3
1.4.2	添加删除	4
1.4.3	红黑树可视化	4
1.4.4	遍历	4
1.4.5	检索	4
2	程序设计	4
2.1	需求分析	4
2.2	概要设计	4
2.3	详细设计	5
2.3.1	Student 类	5
2.3.2	RBTree 类	6
2.3.3	HashTable 类	7
3	设计心得	7
3.1	收获	7
3.2	特色	7
4	文件清单	7
5	附：红黑树高度的证明	8

1 基本原理

1.1 概述

本程序是由 Microsoft Visual Studio 2012 创建,目标框架为 .NET Framework 4.5。程序实现了红黑树数据结构的可视化,实现了对学生成绩信息的输入输出以及检索的功能。此外,实现了 Hash 表,用于管理学生的姓名和学号数据。

1.2 原理

首先实现了将学生封装为 Student 类,接着封装为 Node 类,并在基础上创建红黑树, RBTree 类。接着创建了 HashTable<T> 模板,实现了 Hash 表的功能。最后,在 .NET 框架下,完成图形界面的部署。

1.3 界面

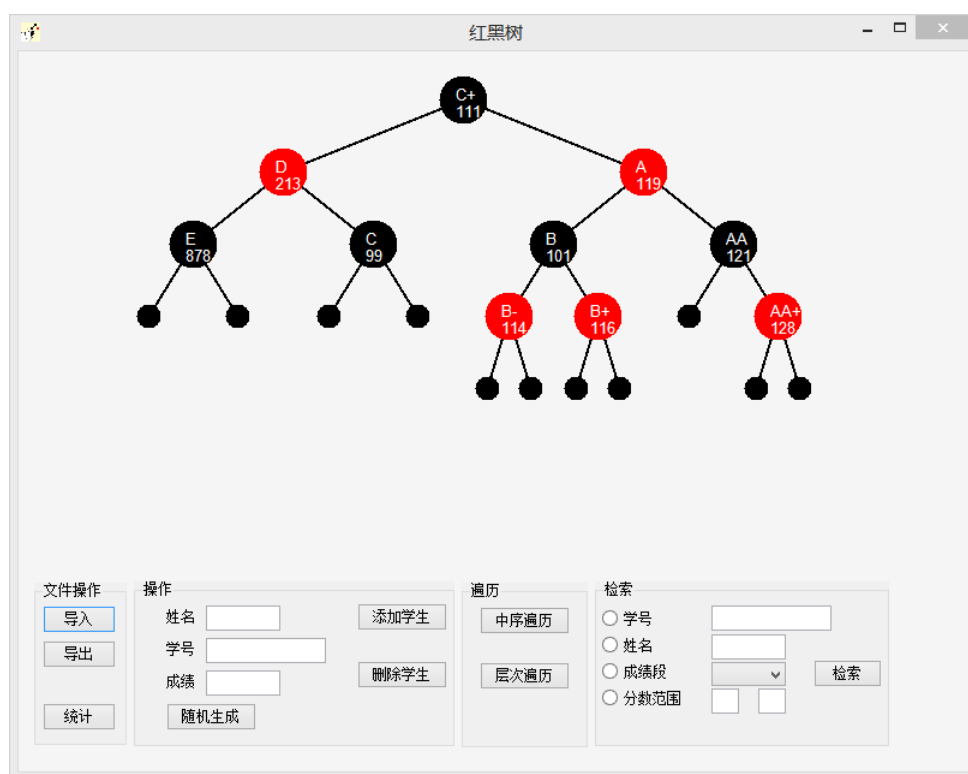


Figure 1: 界面

1.4 操作说明

1.4.1 文件操作

导入, 导出按钮, 可以从文本文件导入数据, 或者导出到文本文件。统计信息按钮可以显示实时的统计信息。

1.4.2 添加删除

输入学生的信息，就可以添加学生的信息，同时，上方的界面将同步绘制响应的红黑树可视化界面。输入学生的学号，就可以删除对应的学生。

1.4.3 红黑树可视化

点击节点，可以在弹出的对话框中查看节点的信息，同时在对话框中选中的条目信息可以回填到主界面中。

1.4.4 遍历

可以实现中序遍历和层次遍历两种方式的遍历。

1.4.5 检索

可以选择通过，姓名，学号，成绩段和分数区间四种不同的方式来检索学生成绩信息。

2 程序设计

2.1 需求分析

程序需要实现红黑树及其可视化，以及 Hash 表的功能。

2.2 概要设计

首先实现了将学生封装为 Student 类，接着封装为 Node 类，并在基础上创建红黑树，RBTree 类。接着创建了 HashTable<T> 模板，实现了 Hash 表的功能。最后，在 .NET 框架下，完成图形界面的部署。

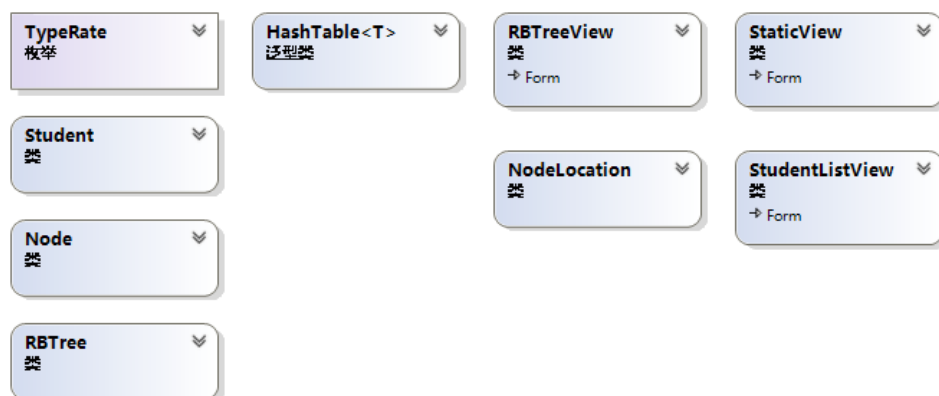


Figure 2: 类图

2.3 详细设计

2.3.1 Student 类

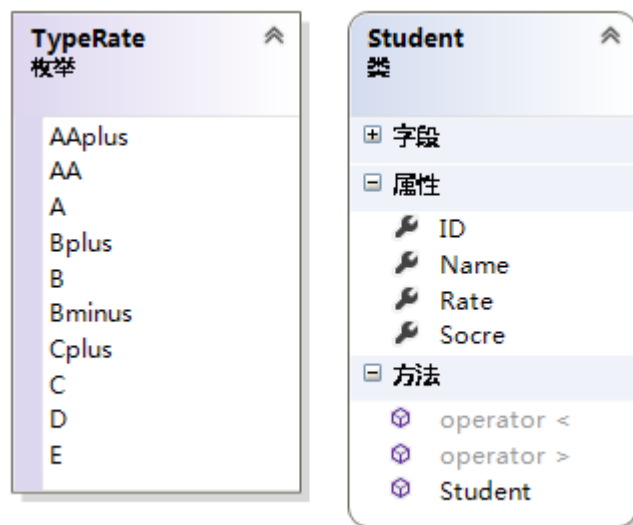


Figure 3: Student 类

枚举型 TypeRate 描述成绩段。

Student 类封装了学生的属性，并重载了 operator< 和 opeartor > 通过分数段来实现对学生的比较，便于在红黑树中进行操作。

2.3.2 RBTREE 类

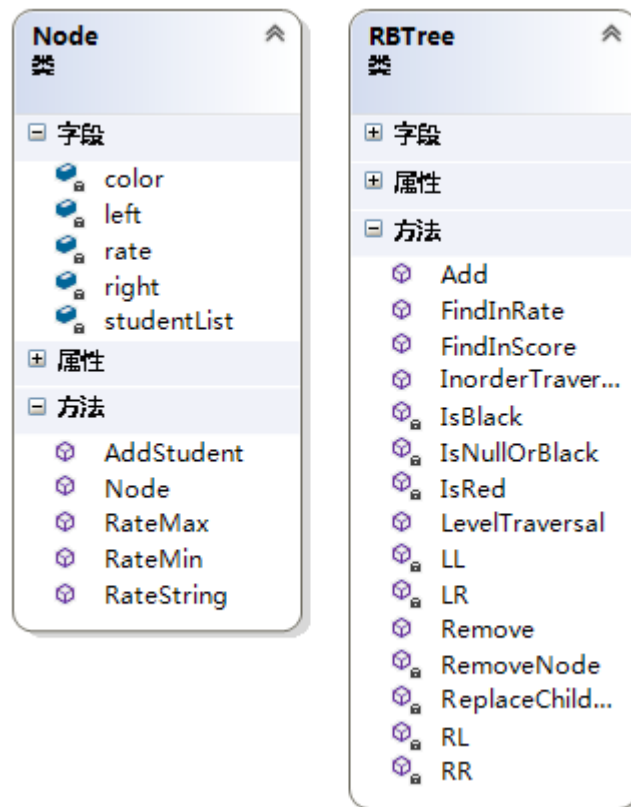


Figure 4: 红黑树的实现

`Node` 为红黑树中的每一个节点，包含的字段有颜色，关键码，左右子树，以及一个线性表用于保存数据。

`RBTREE` 实现了红黑树的插入，删除，查找，遍历等方法。其中树的旋转，换底作为私有方法，用于红黑树的维护。

2.3.3 HashTable 类



Figure 5: HashTable 的实现

HashTable 用泛型编写，实现添加，删除以及查找的功能。Hash 函数的实现，查找匹配的方法使用委托编写，在实例化时实现。

3 设计心得

3.1 收获

这次大作业抛弃了年代久远的 MFC 框架，在全新的 .NET Framework 4.5 框架下完成了此处程序的编写。主要收获是熟悉了 .NET Framework 以及编程语言 C#。

3.2 特色

程序界面简洁。

4 文件清单

- src\ 工程文件
 - Student.cs Student 类，对学生各种属性进行封装
 - RBTree.cs RBTree 类，实现红黑树

- HashTable.cs HashTable<T> 模板，实现哈希表功能
 - RBTreeView.cs 程序主界面
 - StudentListView.cs 列出学生信息的对话框
 - StaticView.cs 统计信息对话框
- bin\ 可执行文件
 - doc\ 文档

程序版本库: <https://github.com/yht1995/RBTree.git>

5 附：红黑树高度的证明

求证：包含 n 个内部节点的红黑树的高度是 $O(\log(n))$ 。

定义： $h(v)$ = 以节点 v 为根的子树的高度。 $bh(v)$ = 从 v 到子树中任何叶子的黑色节点的数目（如果 v 是黑色则不计数它）（也叫做黑色高度）。

引理：以节点 v 为根的子树有至少 $2^{bh(v)} - 1$ 个内部节点。

归纳基础： $h(v) = 0$ 时， $bh(v) = 0$ ，没有内部节点， $2^0 - 1 = 0$ ，结论成立。

归纳假设：若 $h(v) = k$ 的 v 有 $2^{bh(v)} - 1$ 个内部节点，则 $h(v') = k+1$ 的 v' 有 $2^{bh(v')} - 1$ 个内部节点。

因为 v' 有 $h(v') > 0$ 所以它是个内部节点。同样的它有黑色高度要么是 $bh(v')$ 要么是 $bh(v') - 1$ 的两个儿子。它的内部节点数为

$$2^{bh(v')-1} - 1 + 2^{bh(v')-1} - 1 + 1 = 2^{bh(v')} - 1$$

结论证明：因为在从根到叶子的任何路径上至少有一半的节点是黑色（根据红黑树属性 4），根的黑色高度至少是 $h(\text{root})/2$ 。通过引理我们得到：

$$n \geq 2^{\frac{h(\text{root})}{2}} - 1 \Leftrightarrow \log(n+1) \geq \frac{h(\text{root})}{2} \Leftrightarrow h(\text{root}) \leq 2\log(n+1)$$