**基于AVL树的控制台程序测试需求分析**

|  |  |
| --- | --- |
| **产品名称：** | 基于AVL树的控制台程序 |
| **撰写人：** | 09017447 蔡祺燊、09017443杨彦、09017423杨彬 |
| **完成日期：** | 2020.3.5 |
| **评审负责人:**  **评审日期：** | \_\_\_\_\_\_蔡祺燊\_\_\_\_\_\_\_  \_\_\_\_\_2020.3.6\_\_\_\_\_\_ |

## 1概述

### 1.1测试需求分析的目的

本测试需求分析的编写是为明确“基于AVL树的控制台程序测试“项目的所有测试需求，同时检验原先软件需求，并提前分析其测试规摸、复杂程度及可能存在的风险，确保产品质量符合用户需求、测试过程可行。

### 1.2测试需求分析的依据

1. 《基于AVL树的控制台程序软件需求规格说明》

### 1.3测试需求分析的方法

1. 列出软件开发需求中具有可测试性的开发需求；
2. 对1）中的每一条开发需求，形成可测试的分层描述的测试需求；
3. 对2）所确定的质量要求，分析测试执行时需要实施的测试类型；

### 定义

AVL树：是最先发明的自平衡二叉查找树。在AVL树中任何节点的两个子树的高度最大差别为1，所以它也被称为高度平衡树。增加和删除可能需要通过一次或多次树旋转来重新平衡这个树。

## 2 产品说明

### 2.1 项目说明

本项目首先实现了 AVL树数据结构，基于该数据结构实现了方便地进行建树、插入、删除、查找、打印等操作的控制台程序。

### 2.2 项目需求说明

* 实现AVL树数据结构及其建树、删除节点、查找节点、插入节点等操作
* 实现控制台程序，控制台能列出以上所有用户操作，实现用户交互，打印AVL树

## 3 测试需求分析

### 3.1 原始需求

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 系统名称 | 需求版本 | 业务需求编号 | 业务需求名 | 业务需求描述 | 评审状态 |
| AVLTree | 1.0 | 1 | 建立AVL树 | 输入：任意整数数组  输出：AVL树形 | 通过 |
| AVLTree | 1.0 | 2 | 删除节点 | 输入：想要删除的节点值  输出：是否删除成功，并打印树形。 | 通过 |
| AVLTree | 1.0 | 3 | 查找值 | 输入：一个待查找的整数值  输出：查找成功、查找失败 | 通过 |
| AVLTree | 1.0 | 4 | 插入节点 | 输入: 一个待插入的整数值  输出：是否插入成功，并打印树形 | 通过 |
| AVLTree | 1.0 | 5 | 打印树形 | 在屏幕上打印AVL树的树形(采用层级结构打印方式） | 通过 |
| AVLTree | 1.0 | 6 | 控制台菜单 | 能够在控制台中选择建树、查找结点、插入节点、删除节点等功能 | 通过 |

### 3.2 产品测试需求列表

#### 3.2.1 功能测试需求列表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 业务需求编号 | 测试需求编号 | 测试需求名称 | 测试需求描述 | 前置条件 | 预期结果 | 类型 | 优先级 | 作者 | 评审状态 |
| 1 | 1 | 建立AVL树 | 输入数组成功建立AVL树 | 输入一个整形的数组 | 根据输入数组建立一棵AVL树并打印树形。 | 功能测试 | 高 | 杨彦 | 通过 |
| 1 | 2 | 空输入建树 | 输入空数组返回null | 输入空数组 | 输出null | 功能测试 | 高 | 杨彦 | 通过 |
| 1 | 3 | 重复输入建树 | 输入的数组里包含重复值，输出没有重复节点的AVL树 | 输入的数组里含有重复值 | 程序只保留一个重复值，建立节点都不相同的AVL树并打印树形 | 功能测试 | 高 | 杨彦 | 通过 |
| 2 | 4 | 删除存在的节 | 输入的待删除值存在于树中时能够正确删除 | 已建立AVL树且树中存在待删除节点 | 删除AVL树对应的节点，删除后仍然是一颗AVL树，并打印删除节点后的树形 | 功能测试 | 高 | 杨彬 | 通过 |
| 2 | 5 | 删除不存在的节点 | 输入的待删除值不存在于树种 | 已建立AVL树且树中不存在待删除节点 | 不进行任何操作，输出null | 功能测试 | 高 | 杨彬 | 通过 |
| 3 | 6 | 查找存在的节点 | 输入已存在的节点值，正确查找出结果 | 已建立AVL树且树中存在待查找节点 | 输出True | 功能测试 | 高 | 蔡祺燊 | 通过 |
| 3 | 7 | 查找不存在的节点 | 输入一个不存在的节点，返回查找的值不存在 | 已建立AVL树且树中不存在待查找节点 | 输出False | 功能测试 | 高 | 蔡祺燊 | 通过 |
| 4 | 8 | 插入不存在的值 | 输入一个不存在树中的值，能够正确插入并返回树形 | 未建立AVL树或已建立的树中不存在插入的值 | 将值插入到AVL树对应的位置，插入之后仍然是一棵AVL树，输出True,并打印树形 | 功能测试 | 高 | 蔡祺燊 | 通过 |
| 4 | 9 | 插入已存在的值 | 插入已存在的值，树不发生变化，并返回值已存在 | 已建树且树中存在插入的值 | 不做任何操作，输出False | 功能测试 | 高 | 蔡祺燊 | 通过 |
| 5 | 10 | 正确打印树形 | 在树不为空的情况下，以层次结构的方式打印出树形 | 已建树且树不为空 | 按层次关系打印出正确的树形 | 功能测试 | 中 | 杨彬 | 通过 |

#### 3.2.2 性能测试需求列表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 业务需求编号 | 测试需求编号 | 测试需求名称 | 测试需求描述 | 前置条件 | 预期结果 | 类型 | 优先级 | 作者 | 评审状态 |
| 2 | 11 | 删除节点 | 在一定高度下，在较小时间内完成删除操作 | 树高为50 | 5ms以内完成操作 | 性能测试 | 中 | 杨彦 | 通过 |
| 3 | 12 | 查找节点 | 在一定高度下，在一定时间内完成查找操作 | 树高为50 | 5ms以内完成操作 | 性能测试 | 高 | 蔡祺燊 | 通过 |
| 4 | 13 | 插入节点 | 在一定高度下，在一定时间内完成插入操作 | 树高为50 | 5ms以内完成操作 | 性能测试 | 高 | 杨彬 | 通过 |

#### 3.2.3 用户界面测试需求列表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 业务需求编号 | 测试需求编号 | 测试需求名称 | 测试需求描述 | 前置条件 | 预期结果 | 类型 | 优先级 | 作者 | 评审状态 |
| 5 | 14 | 控制台树形打印 | 能够清晰地以层次结构打印AVL树 | 树不为空 | 数据打印完全，节点关系明确，子树不重叠，间隔固定 | 易用性测试 | 低 | 杨彬 | 通过 |
| 6 | 15 | 控制台菜单测试需求 | 能够在控制台中选择建树、查找结点、插入节点、删除节点等功能 | 打开程序 | 打开程序以及每次操作结束后均能显示主菜单 | 易用性测试 | 低 | 杨彦 | 通过 |

### 3.3 测试类型确定

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 系统名称 | 功能测试 | 性能测试 | 易用性 | 可靠性 | 安全性 | 可移植性 | 备注 |
| AVLTree | √ | √ | √ |  |  |  |  |

### 3.4 测试环境要求

#### 3.4.1 软件环境

OS： Windows 10

jdk: java 13.0.1 2019-10-15

Java(TM) SE Runtime Environment (build 13.0.1+9)

Java HotSpot(TM) 64-Bit Server VM (build 13.0.1+9, mixed mode, sharing)

ide: Intellij

#### 3.4.2 硬件环境

CPU：Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.80 GH

RAM：8G

## 4 测试规格评估

### 4.1 测试类型评估

|  |  |  |
| --- | --- | --- |
| 序号 | 测试类型 | 比率 |
| 1 | 功能测试 | 70% |
| 2 | 性能测试 | 20% |
| 3 | 易用性测试 | 10% |

### 4.2 需求覆盖率：100%