项目说明文档

数据结构课程设计

——考试报名系统

作 者 姓 名： 李圣

学 号： 1952723

指 导 教 师： 张颖

学院、 专业： 软件学院 软件工程

同济大学

Tongji Universit

目录

[1 分析 1](#_Toc5959)

[1.1 背景分析 1](#_Toc25144)

[1.2 功能分析 1](#_Toc14293)

[2 设计 1](#_Toc8032)

[2.1 文件结构设计 1](#_Toc19593)

[2.2 数据结构设计 1](#_Toc739)

[2.3 类结构设计 1](#_Toc10946)

[2.4 成员与函数设计 1](#_Toc1295)

[2.5 系统设计 3](#_Toc10331)

[3 实现 3](#_Toc28683)

[3.1 插入功能的实现 3](#_Toc27570)

[3.1.1 插入功能流程图 3](#_Toc2634)

[3.1.2 插入功能核心代码 3](#_Toc7373)

[3.1.3 插入功能截屏示例 3](#_Toc29071)

[3.2 删除功能的实现 3](#_Toc2263)

[3.3 查找功能的实现 3](#_Toc4872)

[3.4 修改功能的实现 3](#_Toc4678)

[3.5 统计功能的实现 3](#_Toc20813)

[3.6 总体系统的实现 3](#_Toc23396)

[4 测试 3](#_Toc12101)

[4.1 鲁棒性测试 3](#_Toc17955)

[4.1.1 正常案例测试 3](#_Toc20283)

[4.1.2 删除测试 4](#_Toc31566)

[4.1.3 查找测试 4](#_Toc20402)

[4.1.4 修改测试 4](#_Toc31904)

[4.1.5 指令测试 4](#_Toc7092)

# 分析

## 背景分析

考试报名系统是一个学校不可缺少的部分，拥有着大量的需求。随着计算机科学技术的不断成熟，使用计算机对考试报名系统进行管理能够极大地提高学校和学生的效率。开发一套考试报名系统十分重要。

## 功能分析

考试报名系统至少应该具有的基本功能：

1. 输入
2. 输出
3. 插入
4. 删除
5. 修改
6. 退出

# 设计

## 文件结构设计

表达式求解的过程中需要使用Vector，因此该项目包含两个文件，如下：

main.cpp

MyVector.h

其中main.cpp是主程序，且main.cpp调用了MyVector.h

## 数据结构设计

## 类结构设计

考虑到功能需要，除了Vector相关的类以外，拟定设计以下类：

class Data

class Calculator

其中Data是数据类，Calculator是计算机类，用于计算相关的操作

## 成员与函数设计

**MyVectorIterator类：**

template<typename val\_type>

using iterator = MyVectorIterator;

using pointer = val\_type\*;

using reference = val\_type&;

**私有成员：**

pointer \_ptr;

int \_pos;

**公有成员：**

MyVectorIterator(pointer ptr = nullptr, int pos = -1)；

~MyVectorIterator()；

int getPos()；

bool operator==(const iterator& it)const；

bool operator!=(const iterator& it)const；

iterator& operator++()；

iterator& operator++(int)；

iterator& operator--()；

iterator& operator--(int)；

reference operator\*()；

pointer operator->()；

**MyVector类：**

template<typename val\_type>

using vector = MyVector<val\_type>;

using reference = val\_type&;

using iterator = MyVectorIterator<val\_type>;

**私有成员：**

int \_size;

int \_capacity;

val\_type\* \_data;

**公有成员：**

MyVector(int capacity = DEFAULT\_CAPACITY)；

MyVector(const vector& vec)；

~MyVector()；

bool empty()；

int size()；

int length()；

int capacity()；

void resize(int newcapacity)；

void insert(iterator pos, const val\_type& val)；

val\_type erase(iterator pos)；

void push\_back(val\_type val)；

val\_type pop\_back()；

vector& operator=(vector& vec)；

vector& operator=(vector&& vec)；

iterator find(val\_type val)；

reference at(int pos)；

reference operator[](int pos)；

reference front()；

reference back()；

iterator begin()；

iterator end()；

void clear()；

## 系统设计

# 实现

## 插入功能的实现

### 插入功能流程图

### 插入功能核心代码

### 插入功能截屏示例

## 删除功能的实现

## 查找功能的实现

## 修改功能的实现

## 统计功能的实现

## 总体系统的实现

# 测试

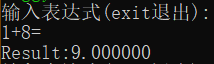
## 鲁棒性测试

### 正常案例测试

**测试用例：**一般情况

**预期结果：**对应的正确结果。

**实验结果：**



**测试用例：**含有括号的情况

**预期结果：**对应的正确结果。

**实验结果：**



**测试用例：**含有单目运算符的情况

**预期结果：**对应的正确结果。

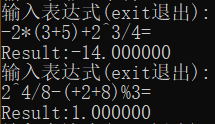
**实验结果：**



**测试用例：**复杂的情况

**预期结果：**对应的正确结果。

**实验结果：**



### 删除测试

### 查找测试

### 修改测试

### 指令测试