**东北大学计算机科学与工程学院**

**数据结构课程设计报告**

题目 学生成绩条形图统计问题

课题组长 戚子强

课题组成员 吕振峰 张卓

专业名称 计算机科学与技术

班级 计1606

指导教师 马安香

2017年12月

目录

**[题目二 学生成绩条形图统计问题](#_Toc503211164)** [1](#_Toc503211164)

**[1](#_Toc503211165)****[课题概述](#_Toc503211165)** [1](#_Toc503211165)

**[1.1 课题任务](#_Toc503211166)** [1](#_Toc503211166)

**[1.2 课题原理](#_Toc503211167)** [2](#_Toc503211167)

**[1.3 相关知识](#_Toc503211168)** [2](#_Toc503211168)

**[2](#_Toc503211169)****[需求分析](#_Toc503211169)** [2](#_Toc503211169)

**[2.1 课题调研](#_Toc503211170)** [2](#_Toc503211170)

**[2.2 用户需求分析](#_Toc503211171)** [3](#_Toc503211171)

**[3 方案设计](#_Toc503211172)** [3](#_Toc503211172)

**[3.1 总体功能设计](#_Toc503211173)** [3](#_Toc503211173)

**[3.2 数据结构设计](#_Toc503211174)** [4](#_Toc503211174)

**[3.3 函数原型设计](#_Toc503211175)** [5](#_Toc503211175)

**[3.4 主算法设计](#_Toc503211176)** [6](#_Toc503211176)

**[3.5 用户界面设计](#_Toc503211177)** [8](#_Toc503211177)

**[3.6 输入输出设计](#_Toc503211178)** [9](#_Toc503211178)

**[4 方案实现](#_Toc503211179)** [9](#_Toc503211179)

**[4.1 开发环境与工具](#_Toc503211180)** [9](#_Toc503211180)

**[4.2 程序设计关键技术](#_Toc503211181)** [9](#_Toc503211181)

**[4.3 个人设计实现（按组员分工）](#_Toc503211182)** [10](#_Toc503211182)

**[5 测试与调试](#_Toc503211183)** [11](#_Toc503211183)

**[5.1 个人测试（按组员分工）](#_Toc503211184)** [11](#_Toc503211184)

**[5.2 组装与系统测试](#_Toc503211185)** [11](#_Toc503211185)

**[5.3 系统运行](#_Toc503211186)** [11](#_Toc503211186)

**[6 课题总结](#_Toc503211187)** [12](#_Toc503211187)

**[6.1 课题评价](#_Toc503211188)** [12](#_Toc503211188)

**[6.2 团队协作](#_Toc503211189)** [12](#_Toc503211189)

**[6.3 个人设计小结（按组员分工）](#_Toc503211190)** [13](#_Toc503211190)

**[7 附录](#_Toc503211191)** [14](#_Toc503211191)

**[A 课题任务分工](#_Toc503211192)** [14](#_Toc503211192)

**[B 课题设计文档（光盘）](#_Toc503211193)** [18](#_Toc503211193)

**[C 使用手册（可选）](#_Toc503211194)** [19](#_Toc503211194)

**题目二 学生成绩条形图统计问题**

1. **课题概述**

**1.1 课题任务**

**课程设计任务书**

|  |
| --- |
| **题目：**  学生成绩条形图统计问题 |
| **问题描述：**  条形图问题描述：给定n个数据，绘出表示这n个数据的条形统计图。即统计出这n个数据中有多少个不同的值，以及每个值出现的频率是多少。条形图常用于表示数据分布情况。例如，学生考试成绩统计、居民收入分布情况等。假设输入数据为正整数，利用二叉排序树完成输入数据频率统计。 |
| **设计要求：**  设计基于二叉排序树的学生成绩条形图统计程序。  （1）采用STL的二叉排序树等数据结构。  （2）应用基本运算，设计算法求解。 |
| **指导教师签字：**  **年　　月　　日** |

**1.2 课题原理**

数据获取：将学生成绩保存在CSV文件中，每个学生成绩各占一行，共九列数据，分别为ID号码、六科成绩、总分、排名，在用户界面输入对应的列号，即可得到对应的数据分布条形图。读入数据时，先文件字符串格式进行切片和数据类型转换，保存进动态数组中。

数据处理：调用二叉排序树数据输入函数（形参为一个动态数组），将数据依次存进二叉树。首先从二叉树根部开始（如果为空，则建立二叉树），比较待插入数字与结点值大小，若小于结点值，则继续与左子树的根结点比较；若大于结点值，则继续与右子树的根节点比较；出现重复的数字将使计数变量加一，而终止查找。在找到合适的位置后，将临时结点插入二叉树中，否则释放临时结点。最终，二叉树的结点中保存所有出现过的数字及对应的频数，而没有重复。

数据输出：通过二叉树的中序遍历，将二叉树中的有序数据及对应的频数分别存入动态数组中，此时，数组是有序且对应的。

析构函数：从根节点开始，利用递归，依次向左右孩子结点移动指针并释放结点占据的存储空间。

**1.3 相关知识**

* C++语言
* 数据结构STL框架
* 二叉排序树
* 树的遍历
* 动态数组

1. **需求分析**

**2.1 课题调研**

学生成绩即是同学们学习成果的体现，也是教学质量的反映。老师可以通过统计各分数段的频数获得学生成绩分布直方图，进而分析得出整体的学习情况，以期对下一阶段的教学提供参考。但当学生数量增多，数据量变得庞大时，人工统计和计算不再现实。本次课题就是希望通过程序处理数据，对数据进行排序和频数统计，并绘出“学生成绩分布直方图”，具体用到的数据结构为二叉排序树。

如今的学校对学生成绩的观察越来越细致，条形图统计能够清晰的反应出学生成绩的总体分布，实现对数据分区段处理的意义在于：能够将所获取的数据通过计算机自动的统计处理并输出条形图样展示学生成绩的分布及频率，实现高效的数据处理过程，展现出直观的数据处理结果。

**2.2 用户需求分析**

经过实际的调研，本程序需要满足拥护的以下需求：

输入整数数据：

1.可以根据数据情况自定义设置分段并展示各分段的统计结果。

2.程序可以调试

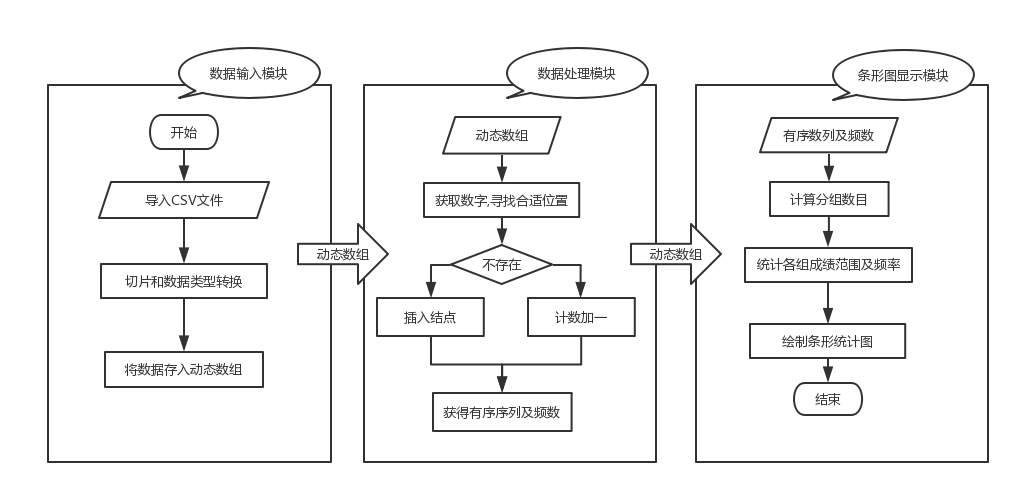
3.对错误数据具有甄别能力

4.输入输出数据显示在界面上

5.数据显示清晰，通过颜色区分

**3 方案设计**

**3.1 总体功能设计**



**3.2 数据结构设计**

class BSTNode

{

public:

BSTNode();

BSTNode(int val);

int value;

int number; //计数元素

BSTNode \*lchild;

BSTNode \*rchild;

};

class BST

{

//为了数据结构私有化，不为外部访问，这里提供一些私有内部函数实现真正的操作以"\_\_"开头。

//对于public的接口来说，只需要直接调用内部函数即可

private:

BSTNode \* bstroot;//二叉排序树数据结构

BSTNode \* \_\_search(BSTNode\* root,const int& key);//查找关键字

BSTNode \* \_\_treeMin(BSTNode\*const root,BSTNode \*&parent);//返回当前节点的最小孩子（一路向左）

BSTNode \* \_\_treeMax(BSTNode\*const root);//查找最大值（未实现）

bool \_\_Insert( const int &key);//插入节点

bool \_\_Delete(const int &key);//删除删除

bool \_\_isLeaf(BSTNode\* const &);//判断是否是叶子节点

bool \_\_isNodeWithTwoChild(BSTNode \* const &);//判断是否有两个孩子

void \_\_InorderTraversal(BSTNode \*root,std::vector<int>&result,std::vector<int>&Num);//中序遍历

void \_\_DeleteAllNodes(BSTNode \*root);//删除所有节点

public:

//构造函数

BST();//默认构造函数

BST(std::vector<int>arr);

BST(int \*arr, int len);

//析构函数

~BST();

bool isEmpty() const;//判断树空

bool search(const int &key);//查找关键字是否存在的对外接口

bool Insert(const int &key);//插入节点的外部接口

bool Delete(const int &key);//删除节点的外部接口

void InorderTraversal(std::vector<int>&,std::vector<int>&Num);//中序遍历的外部接口

};

**3.3 函数原型设计**

class BSTNode

{

public:

BSTNode();

BSTNode(int val);

int value;

int number; //计数元素

BSTNode \*lchild;

BSTNode \*rchild;

};

class BST

{

//为了数据结构私有化，不为外部访问，这里提供一些私有内部函数实现真正的操作以"\_\_"开头。

//对于public的接口来说，只需要直接调用内部函数即可

private:

BSTNode \* bstroot;//二叉排序树数据结构

BSTNode \* \_\_search(BSTNode\* root,const int& key);//查找关键字

BSTNode \* \_\_treeMin(BSTNode\*const root,BSTNode \*&parent);//返回当前节点的最小孩子（一路向左）

BSTNode \* \_\_treeMax(BSTNode\*const root);//查找最大值（未实现）

bool \_\_Insert( const int &key);//插入节点

bool \_\_Delete(const int &key);//删除删除

bool \_\_isLeaf(BSTNode\* const &);//判断是否是叶子节点

bool \_\_isNodeWithTwoChild(BSTNode \* const &);//判断是否有两个孩子

void \_\_InorderTraversal(BSTNode \*root,std::vector<int>&result,std::vector<int>&Num);//中序遍历

void \_\_DeleteAllNodes(BSTNode \*root);//删除所有节点

public:

//构造函数

BST();//默认构造函数

BST(std::vector<int>arr);

BST(int \*arr, int len);

//析构函数

~BST();

bool isEmpty() const;//判断树空

bool search(const int &key);//查找关键字是否存在的对外接口

bool Insert(const int &key);//插入节点的外部接口

bool Delete(const int &key);//删除节点的外部接口

void InorderTraversal(std::vector<int>&,std::vector<int>&Num);//中序遍历的外部接口

};

**3.4 主算法设计**

/\*插入函数\*/

//插入节点的内部实现，插入操作一定都在叶子节点处。

bool BST::\_\_Insert(const int & key)

{

int a;

BSTNode\* t = new BSTNode(key);//临时节点

t->number = 1;

BSTNode\*parent = NULL;

if (isEmpty())//新树

{

bstroot = t;

return true;

}

else

{

BSTNode\* curr;

curr = bstroot;

while (curr)

{

//插入位置都位于叶子节点处

parent = curr;

if (t->value > curr->value)

curr = curr->rchild;

else if (t->value < curr->value)

curr = curr->lchild;

else

{

curr->number++;//curr与待插入的节点相同,计数加一

break;

}

}

if (t->value < parent->value)

{

parent->lchild = t;

return true;

}

else if(t->value > parent->value)

{

parent->rchild = t;

return true;

}

//else ;

}

return false;

}

//中序遍历内部实现

void BST::\_\_InorderTraversal(BSTNode \*root,std::vector<int>&result,std::vector<int>&Num)

{

if (NULL == root)

return;

\_\_InorderTraversal(root->lchild, result, Num);

std::cout << root->value <<"->"<<root->number<< " ";

result.push\_back(root->value);

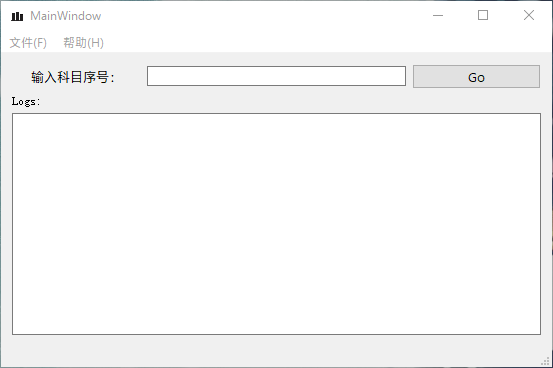
Num.push\_back(root->number);

\_\_InorderTraversal(root->rchild, result, Num);

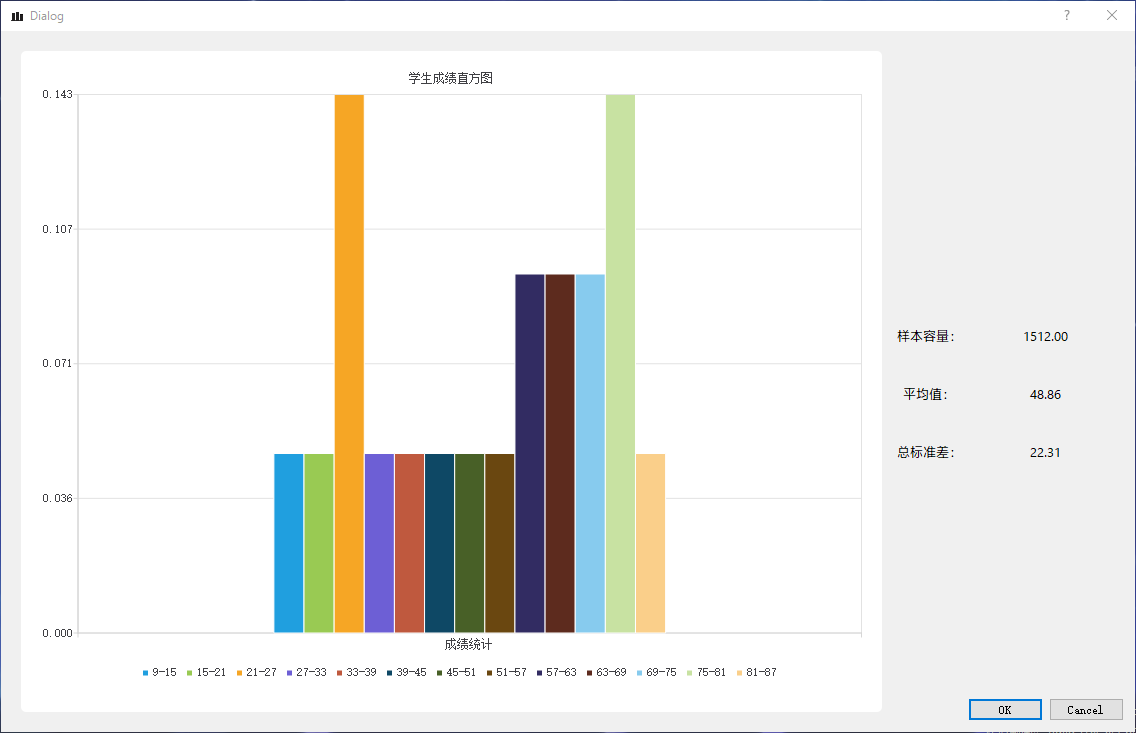
}

**3.5 用户界面设计**

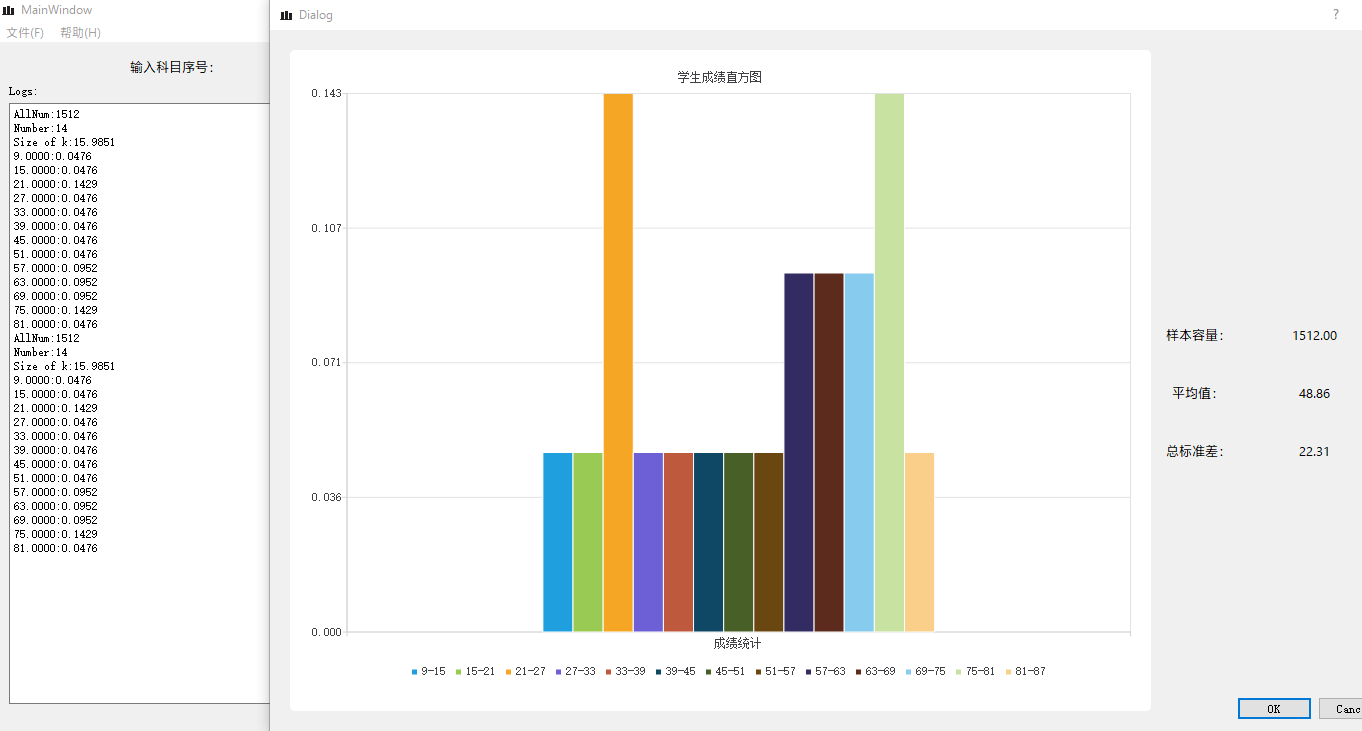
主界面如下：



导入文件后选择科目，点击GO，弹出子窗口并在子窗口绘图：



子窗口显示直方图，父窗口显示Logs：



**3.6 输入输出设计**

输入：将学生成绩保存在CSV文件中，每个学生成绩各占一行，共九列数据，分别为ID号码、六科成绩、总分、排名，在用户界面输入对应的列号，即可得到对应的数据分布条形图。读入数据时，先文件字符串格式进行切片和数据类型转换，保存进动态数组中。

输出：通过二叉树的中序遍历，将二叉树中的有序数据及对应的频数分别存入动态数组中，此时，数组是有序且对应的。

**4 方案实现**

**4.1 开发环境与工具**

开发环境：Windows 10 + Qt 5.10 + MinGw 5.3

开发工具：Qt Creator 4.4 + NotePad++ 7.2

**4.2 程序设计关键技术**

* **数据结构中的二叉树**
* **QSS**
* **C++ STL**

**4.3 个人设计实现（按组员分工）**

**4.3.1 吕振峰设计实现**

框架设计：采用C++数据结构STL框架，根据需求列出所需实现的程序；比如为了数据结构私有化，不为外部访问，这里提供一些私有内部函数实现真正的操作以"\_\_"开头；而对于public的接口来说，只需要直接调用内部函数即可。

建立二叉树：采取两种建立方法，利用形参的不同实现函数重载；其一形参为空；其一形参为保存成绩的动态数组。通过依次插入动态数组中的元素，逐步建立二叉树（每次建立的新结点均为叶子）。

二叉树中序遍历：根据二叉排序树的建立原理，采用中序遍历的方法可得到依次增大的有序结点。采用递归算法，函数形参包括子树的根结点，两个保存结果的动态数组的指针，每遍历到一个结点，就将数值及计数元素放在数组的结尾，最终得到有序数列及对应的频数。

**4.3.2戚子强设计实现**

这里我主要做的工作是把C++原生代码移植到Qt中，将所有I/O操作替换并做一些界面UI和信号槽以及利用返回数据进行绘图的工作。

由于在移植代码之前，队友已经将C++原生代码做得比较完善了，所以在整个过程中并没有太大的障碍。主要难点在绘图实现上，刚开始使用的是QWT，过程比较繁琐并出现了很多问题，之后得知QtCharts已经开源，所以使用了QtCharts进行绘图工作。

**4.3.3张卓设计实现**

数据获取：将CSV文件读出，利用CSV存储特点（通过“，”和换行符分割元素）进行切片，得到独立的元素；再通过元素类型转化得到整形数字，并存入动态数组中。

数据统计处理：对获取的有序数列及对应的频数进行分组、统计、计算；组数通过公式“k = 1 + 3.322\*log(double(result.back()-result[0]))”获得，其中k为组数，log处理的对象为数据集合的极差。最后将成绩段划分及对应的频率通过动态数组的方式传给直方图显示模块。

**5 测试与调试**

**5.1 个人测试（按组员分工）**

**5.1.1 吕振峰测试**

首先对类中程序的调用进行检查，使程序调用不出现错误。再手动输入结点值，着重测试数据出现重复时的处理情况。接着测试附加程序，即在函数调用时使用较少，较隐藏的函数，比如删除函数、查找函数和修改函数。

**5.1.2戚子强测试**

这里我所做的主要是文件操作方面的测试，主要是查看大量数据写入的时候程序处理时间是否在可接受范围之内，最终的情况是令人欣喜的。程序处理成绩的时间在可接受时间范围之内，没有出现异常崩溃等情况。

**5.1.3张卓测试**

建立CSV文件，并编写、录入模拟数据。调用程序，输出切片和数据转化结果，观察是否一致。设置极端情况，观察程序反映是否与预期一致。

**5.2 组装与系统测试**

**组装：**由三位成员完成各自负责的C++原生代码设计后，由戚子强将原生代码合并并移植到Qt平台，期间主要是替换所有I/O操作和解决一些逻辑兼容问题。

**系统测试：**组装完毕之后，在Qt平台编译DEBUG版本并分发到各位成员手中进行各种可能的数据测试，一旦出现不符合实际情况的部分立即进行分析并进行记录，找到目标问题所在的函数位置；如此进行反复测试。将所在函数位置修改完成后在Qt中进行局部替换，经过多次测试后生成RELEASE版本并进行程序打包成成安装包，请各位同学安装并进行了多次测试。期间出现的问题也会及时记录并修改。

**5.3 系统运行**

最终版本是安装包，内置了Qt各个组件以保证在各种平台正常运行而不会出现缺少组件的情况。用户可以像平常安装软件一样进行操作。

**6 课题总结**

**6.1 课题评价**

学生成绩条形图统计问题可以看作是一类统计问题，即得到大量的数据，需要对这些数据进行统计处理得出他们的分布趋势，也就是说看看这些数据分别落在哪些区间内，并且众数平均数等数据是多少的问题。

利用计算机代替人工处理大量数据，自然是有非常大的优势，除了能够节约时间和节约劳动力外，程序的可重复使用性能够保证只要有数据，就可以在短时间内得到统计结果。

对于本次研究课题设计，我们通过学习STL标准类库并且使用STL类才进行编程，设计过程中不仅要考虑实现哪些功能，同样还要考虑如何布局，其实每个程序都有一些共同点，通用的结构，相似的格式，使得自己的编程能力得到了提高，解决问题的方式更加多元化，团队合作意识得到加强，我们收获良多。

**6.2 团队协作**

组长提出了该系统应该具有的功能并做记录，与组员们进行讨论，综合建议和信息，制定了总体的设计框架和系统具备的功能。

大家一起进行了详细的分工明确，确定了我们的设计最终要实现什么样的功能，并进一步细致化，分类成一个个的函数，将函数的设计分配给每一个人。吕振峰主要负责整个程序的走向、二叉排序树的设计、二叉排序树的插入和中序遍历这一部分的设计;戚子强负责，函数设计包括界面的设计，以及可视化操作设计等等;张卓主要负责主函数中数据输入、数据再处理工作。任务分工完成后，大家都回去查找相关资料，潜心研究，独立完成各自的设计任务。

每次上机课的时候，我们都会在一起讨论研究，进一步明确设计所要实现的具体目标，解决代码编写过程中的编写错误或者是当时设计时出现的纰漏。

经过组内人员的共同努力，最后我们顺利地完成了此次程序设计任务，实现了学生成绩条形图统计课题要求。更为珍贵的是，通过团队合作，我们学会了团队开发模式，注重交流，分享经验，从彼此身上学到了很多，也为自己能在团队中贡献一份力量感到由衷的开心。

**6.3 个人设计小结（按组员分工）**

**6.3.1 吕振峰设计小结**

通过这次课程设计，使我更充分认识了团队合作的重要性。由于这次课设是以小组为单位的，还要查找资料，每个人都有分工。所以在进行分析的过程中每个成员都要保证查找资料时都要有耐心，这样才能确保小组顺利完成任务。在这次课设过程中，我们小组成员之间都互帮互助、共同思考，相互查漏补缺，互相给予信心，这样得以保证高速、高效率的完成任务，充分体现了团队精神。

课程设计就是简单的毕业设计，通过这次课设。给予我们实践的机会使我积累了一定的经验，为以后的学习打下了基础。

对二叉树的概念有了一个新的认识，在学习离散数学的时候，总觉得二叉树是很抽象的东西，但是在学习了《数据结构》这门课程之后，我慢慢地体会到了其中的奥妙，二叉树能够在计算机中存在，首先要捕捉他有哪些具体化、数字化的信息，比如说权值、叶子个数等，这也就说明了想要把生活中的信息转化到计算机中必须用数字来完整的构成一个信息库，而二叉排序树的更是一件很巧妙的事情。在计算机中实现一个很简单的想法就需要涉及到很多专业知识，为了完成设计，在前期工作中，基本都是以学习二叉排序树的基本操作和STL类库为主。有了这次课程设计的经验和教训，我能够很清楚的对自己定一个合适的水平，而且在这次课程设计中我学会了运用二叉排序树对数据进行存储和处理。通过这段时间的课程设计，我认识到数据结构是一门比较难的课程。需要多花时间上机练习。这次的程序训练培养了我实际分析问题、编程和动手能力，使我掌握了程序设计的基本技能，提高了我适应实际，实践编程的能力。

总的来说，这次课程设计让我获益匪浅，对数据结构也有了进一步的理解和认识。

**6.3.2 戚子强设计小结**

通过课程设计，使我加深了对所学理论知识的理解与巩固，并能将课本上的纯理论应用到实践中，进一步加深了对知识的认识。同时，也有助于对其他知识的理解。我深刻地体会到了自学能力的重要性。着手程序的总体设计时，我发现课堂上学习的知识还远远不够，很多东西都需要自己临时查阅并学以致用，例如需要使用QT可视化编程，需要使用STL类库，需要学习如何通过程序调用Excel表格等等。靠自己的不断尝试，凭借老师和同学的帮助，以及查阅相关资料，在网络上寻找答案，再自己一点点调试，才最终完成了这个实验设计。

通过对STL的学习，我较为全面的了解到了STL类库的实现方式，并且感受到了它给编程所带来的方便，收获良多。

可视化编程的实现因为有了A题的学习基础，在B题上并没有遇到太大的难题，只是需要不断的调试，来使程序达到最好的运行效果，减少错误的出现。

最棘手的问题莫过于使用程序调用Excel操作了，为了实现这个功能，我还是花了一天的时间去尝试，最后事实证明这时间花的是值得的。我通过学习尝试最终实现了调运Excel的功能。

最后感谢在程序设计中给予我帮助的同学们和老师们，有了你们的建议和指导，我们这个团队才能不断完善设计，找到程序的Bug，并最终设计出一个相对完备的学生成绩条形图统计程序。

**6.3.3 张卓设计小结**

通过这次课程设计，我充分认识到数据结构这门课的重要性,在本课程设计中，我明白了理论与实际应用相结合的重要性，并提高了自己组织数据及编写大型程序的能力。培养了基本的、良好的程序设计技能以及合作能力。这次课程设计同样提高了我的综合运用所学知识的能力。并对VC有了更深入的了解。《数据结构》是一门实践性很强的课程，上机实习是对学生全面综合素质进行训练的一种最基本的方法，是与课堂听讲、自学和练习相辅相成的、必不可少的一个教学环节。上机实习一方面能使书本上的知识变“活”，起到深化理解和灵活掌握教学内容的目的;另一方面，上机实习是对学生软件设计的综合能力的训练，包括问题分析，总体结构设计，程序设计基本技能和技巧的训练。此外，还有更重要的一点是:机器是比任何教师更严厉的检查者。因此，在“数据结构”的学习过程中，必须严格按照老师的要求，主动地、积极地、认真地做好每一个实验，以不断提高自己的编程能力与专业素质。这次实验培养了我耐心、仔细、谨慎的工作态度。这次课程设计的内容最多的是绘画机构简图。在绘图过程中，对比例尺的选定以及合理的布图都要有细致的规划，这是最考验一个人的耐心与严谨态度的过程，只有这样才能使绘制的机构简图更准确、更美观。课程设计诚然是一门专业课，给我很多专业知识以及专业技能上的提升，同时，设计让我感触很深。使我对抽象的理论有了具体的认识。

**7 附录**

**A 课题任务分工**

**A-1 课题程序设计分工**

**课题程序设计分工**

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | 姓名 | 程序设计函数原型、类 | 功能说明 |
|  | 吕振峰 | class BSTNode  {  public:  BSTNode();  BSTNode(int val);  int value;  int number;  BSTNode \*lchild;  BSTNode \*rchild;  };  class BST  {  //为了数据结构私有化，不为外部访问，这里提供一些私有内部函数实现真正的操作以"\_\_"开头。  //对于public的接口来说，只需要直接调用内部函数即可  private:  BSTNode \* bstroot;//二叉排序树数据结构  BSTNode \* **\_\_search**(BSTNode\* root,const int& key);//查找关键字  BSTNode \* **\_\_treeMin**(BSTNode\*const root,BSTNode \*&parent);//返回当前节点的最小孩子（一路向左）  BSTNode \* **\_\_treeMax**(BSTNode\*const root);//查找最大值（未实现）  bool **\_\_Insert**( const int &key);//插入节点  bool **\_\_Delete**(const int &key);//删除删除  bool **\_\_isLeaf**(BSTNode\* const &);//判断是否是叶子节点  bool **\_\_isNodeWithTwoChild**(BSTNode \* const &);//判断是否有两个孩子  void **\_\_InorderTraversal**(BSTNode \*root,std::vector<int>&result,std::vector<int>&Num);//中序遍历  void **\_\_DeleteAllNodes**(BSTNode \*root);//删除所有节点  public:  //构造函数  BST();//默认构造函数  BST(std::vector<int>arr);  BST(int \*arr, int len);  //析构函数  ~BST();  bool **isEmpty**() const;//判断树空  bool **search**(const int &key);//查找关键字是否存在的对外接口  bool **Insert**(const int &key);//插入节点的外部接口  bool **Delete**(const int &key);//删除节点的外部接口  void **InorderTraversal**(std::vector<int>&,std::vector<int>&Num);//中序遍历的外部接口  }; |  |
|  | 戚子强 | void **getWidget**(QChartView \*chartView);  void **getNum**(int numbers,double aver, double sd);  void **AddChartView**(QChartView \*chartview);  void **sendNum**(int numbers,double aver, double sd); | 界面UI操作和窗口间通信等 |
|  | 张卓 | void **Output**();  void **openfile**(); | 输入输出、调用库以及文件操作等 |

**A-2 课题报告分工**

**课题报告分工**

|  |  |  |
| --- | --- | --- |
| 章节 | 内容 | 完成人 |
| 1 课题概述 | 1.1 课题任务  1.2 课题原理  1.3 相关知识 | 吕振峰 |
| 2 需求分析 | 2.1 课题调研  2.2 用户需求分析 | 张卓 |
| 3 方案设计 | 3.1 总体功能设计  3.2 数据结构设计  3.3 函数原型设计  3.4 输入输出设计  3.5 主算法设计  3.6 用户界面设计 | 吕振峰 |
| 4 方案实现 | 4.1 开发环境与工具  4.2 程序设计关键技术  4.3 个人设计实现（按组员分工）  4.3.1  4.3.2  4.3.3 | 吕振峰 |
| 5 测试与调试 | 5.1 个人测试（按组员分工）  5.1.1  5.1.2  5.1.3  5.2 组装与系统测试  5.3 系统运行 | 戚子强 |
| 6 课题总结 | 6.1 课题评价  6.2 团队协作  6.3 下一步工作  6.4 个人设计心得（按组员分工）  6.4.1  6.4.2  6.4.3 | 张卓 |

**B 课题设计文档（光盘）**

**B-1课程设计报告（电子版）**

即本文。

**B-2源程序代码（\*.H，\*.CPP）**

BST.h

dialog.h

mainwindow.h

BST.cpp

dialog.cpp

mainwindow.cpp

main.cpp

**B-3工程与可执行文件**

工程请见本目录下Grades文件夹

**B-4屏幕演示录像文件（可选）**

<Grades.mp4>

**C 使用手册（可选）**

**C.1 运行环境说明**

操作系统: Win XP+ X86 & X64

配置： 内存 2GB+

硬盘 40GB+

**C.2 操作说明**

打开主界面，先导入CSV表格文件，再输入科目，科目一输入1，以此类推，点击GO，即可看到子窗口的统计图和父窗口的Logs记录。