**《数据结构》课程设计**

**学号： 1453381**

**姓名： 曾 鸣**

**专业： 计算机科学与技术**

2017年7月

目录

[**第一部分 算法实现设计说明** 1](#_Toc491183069)

[**1.1** **题目** 1](#_Toc491183070)

[**1.2** **软件功能** 1](#_Toc491183071)

[**1.3** **设计思想** 1](#_Toc491183072)

[**1.4** **逻辑结构与物理结构** 2](#_Toc491183073)

[1.4.1 单链表 2](#_Toc491183074)

[1.4.2 循环链表 4](#_Toc491183075)

[1.4.3 双向（循环）链表 6](#_Toc491183076)

[**1.5** **开发平台** 7](#_Toc491183077)

[**1.6** **系统的运行结果分析说明** 8](#_Toc491183078)

[**1.6.1** **调试及开发过程** 8](#_Toc491183079)

[**1.6.1.1** **调试** 8](#_Toc491183080)

[**1.6.1.2** **开发** 8](#_Toc491183081)

[**1.6.2.1** **正确性** 9](#_Toc491183082)

[**1.6.2.2** **稳定性** 15](#_Toc491183083)

[**1.6.2.3** **容错能力** 16](#_Toc491183084)

[**1.7** **操作说明** 18](#_Toc491183085)

[1.7.1 主界面操作 18](#_Toc491183086)

[**1.7.1.1** **运行** 18](#_Toc491183087)

[**1.7.1.2** **主界面** 19](#_Toc491183088)

[1.7.2 单链表 20](#_Toc491183089)

[**1.7.2.1** **进入功能区** 20](#_Toc491183090)

[**1.7.2.2** **调节演示速度** 21](#_Toc491183091)

[**1.7.2.3** **创建单链表** 22](#_Toc491183092)

[**1.7.2.4** **插入节点** 23](#_Toc491183093)

[**1.7.2.5** **删除节点** 25](#_Toc491183094)

[**1.7.2.5** **查找节点** 26](#_Toc491183095)

[**1.7.2.6** **销毁链表** 28](#_Toc491183096)

[1.7.3 循环链表 28](#_Toc491183097)

[1.7.4 双向（循环）链表 28](#_Toc491183098)

[**第二部分 综合应用设计说明** 29](#_Toc491183099)

[**2.1** **题目** 29](#_Toc491183101)

[**2.2** **软件功能** 29](#_Toc491183102)

[**2.3** **设计思想** 29](#_Toc491183103)

[**2.4** **逻辑结构与物理结构** 31](#_Toc491183104)

[2.4.1 站点 32](#_Toc491183105)

[2.4.2 线路 32](#_Toc491183106)

[2.4.3 后端实现类 33](#_Toc491183107)

[**2.5** **开发平台** 35](#_Toc491183108)

[**2.6** **系统的运行结果分析说明** 35](#_Toc491183109)

[**2.6.1** **调试及开发过程** 35](#_Toc491183110)

[**2.6.1.1** **调试** 36](#_Toc491183111)

[**2.6.1.2** **开发** 36](#_Toc491183112)

[**2.6.2.1** **正确性** 37](#_Toc491183113)

[**2.6.2.2** **稳定性** 43](#_Toc491183114)

[**2.6.2.3** **容错能力** 43](#_Toc491183115)

[**2.7** **操作说明** 45](#_Toc491183116)

[2.7.1 视图查看 45](#_Toc491183117)

[2.7.2 换乘查询 47](#_Toc491183118)

[2.7.3 动态添加 50](#_Toc491183119)

[2.7.4 工具栏和状态栏 54](#_Toc491183120)

[2.7.5 帮助 54](#_Toc491183121)

[2.7.6 关于 55](#_Toc491183122)

[2.7.7 退出 55](#_Toc491183123)

[**第三部分 实践总结** 56](#_Toc491183124)

[**3.1** **所做的工作** 56](#_Toc491183126)

[**3.2** **总结和收获** 56](#_Toc491183127)

[**第四部分 参考文献** 56](#_Toc491183128)

**第一部分 算法实现设计说明**

* 1. **题目**

题号1：分别以单链表、循环链表、双向链表为例，实现线性表的建立、插入、删除、查找等基本操作。

要求：能够把建立、插入、删除等基本操作的过程随时显示输出来。

* 1. **软件功能**

功能分为三个板块，分别是单链表、循环链表、双向链表的建立、插入、删除等基本操作的过程。

**单链表：**

●查看单链表定义，相应算法代码；

●建立一个带头结点的空单链表；

●指定插入位置及元素值到单链表中；

●随机插入5个元素到单链表的尾部；

●指定删除元素位置（从1开始），从单链表中删除；

●输入查找值，得到元素在单链表中位置；

●清空销毁单链表；

●能够调整演示的速度快慢；

**循环链表：**

●查看循环链表定义，相应算法代码；

●建立一个带头结点的空循环链表；

●指定插入位置及元素值到循环链表中；

●随机插入5个元素到循环链表的尾部；

●指定删除元素位置（从1开始），从循环链表中删除；

●输入查找值，得到元素在循环链表中位置；

●清空销毁循环链表；

●能够调整演示的速度快慢；

**双向链表：**

●查看双向链表定义，相应算法代码；

●建立一个带头结点的空双向链表；

●指定插入位置及元素值到双向链表中；

●随机插入5个元素到双向链表的尾部；

●指定删除元素位置（从1开始），从双向链表中删除；

●输入查找值，得到元素在双向链表中位置；

●清空销毁双向链表；

●能够调整演示的速度快慢；

上述所有功能采用面向对象的方法通过C++语言程序结合QT框架实现，后面会详细介绍。

* 1. **设计思想**

**实现思路**

1. 学习相应知识，做好必要的准备工作

由于以前都是采用控制台进行编程，即便涉及一些简单的图像界面，但是比较粗制简陋，无法入眼，并不是标准规范、人性化的用户交互界面，所以要完成本次的数据结构课程设计必须从零起步，学习可视化编程开发。在C++的一系列可视化开发框架下，我选择用Qt来实现程序的功能，因为Qt相对较为简单，容易上手入门，同时Qt是较为新兴的技术框架，并且跨平台开发，很有前景和实用性。通过几天的学习，理解掌握的Qt的必要知识，包括最为核心的信号和槽函数机制、UI控件的使用、Scene-View视图框架等核心技术。

1. 自顶向下设计

有了必要的准备知识，就可以进行程序的总体规划设计了。自顶向下分析是常用的分析方法，本次题目其实较为简单，常用的链表结构我们在学习数据结构课程时已经非常熟练，此次实现图形化界面的演示需要结合原有结构，融入图形化元素和用户界面接口，对程序的功能分析，显然程序的功能分为三个子功能模块，分别对每个模块进行设计即可完成整个任务。

1. 分模块实现

虽然程序有三个部分组成，但是每个部分的功能需要完全一样，用户界面完全一样，划分为三个模块，只要实现一个模块，其余两个模块只要非常短的时间就可完成。实际开发时，先实现单链表模块，完成以后，循环链表和双向链表只需要在前面基础上稍作修改即可。

1. 自底向上实现

具体实现时，先定义每个类的属性和相应函数，然后根据定义，设计相应算法自底向上进行实现，逐个击破，最终完成所有程序的设计。

* 1. **逻辑结构与物理结构**

## 1.4.1 单链表

节点定义：



单链表：



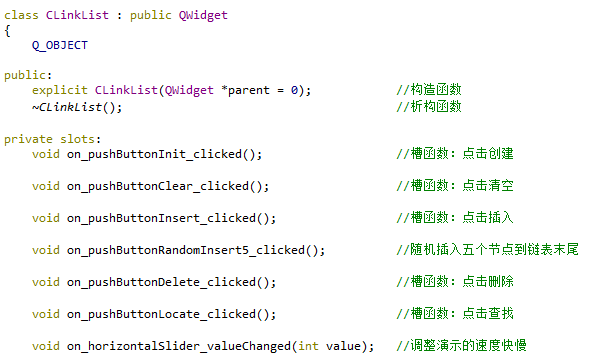


## 1.4.2 循环链表

节点定义：



循环链表：





## 1.4.3 双向（循环）链表

节点定义：



双向循环链表：





* 1. **开发平台**

**开发平台：**

计算机型号 ：惠普Pavilion M4

计算机内存 ：4.00GB

CPU ：Intel Core i5 2.6GHz

操作系统 ：Windows 10 家庭版

开发语言 ：C++（C++11标准以上）

开发框架 ：QT

集成开发环境：Qt Version 5.9.1

编译器 ：MinGW 32bit

**运行环境：**

可在上述集成环境下运行；

通过windeployqt.exe及Enigma Virtual Box进行整合压缩为发布为了一个LinkListVisualizer.exe文件，可在普通windows机型下运行。

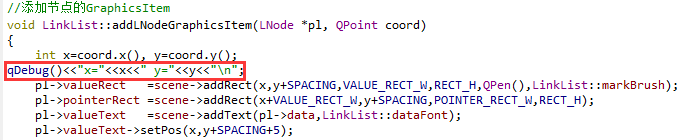
* 1. **系统的运行结果分析说明**

**1.6.1 调试及开发过程**

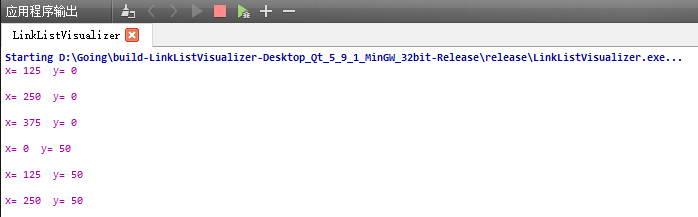
**1.6.1.1 调试**

本次开发采用的是新技术框架Qt，同时也是跨平台的，在Qt Creator中开发调试，Qt中包含了大量的库类，类似于java开发简便，Qt有较好的调试器，但是在本次开发中没有用到，偶尔遇到一些小麻烦或者小bug，我们只需在控制台中输出一些数据便可分析定位错误原因。

例如：



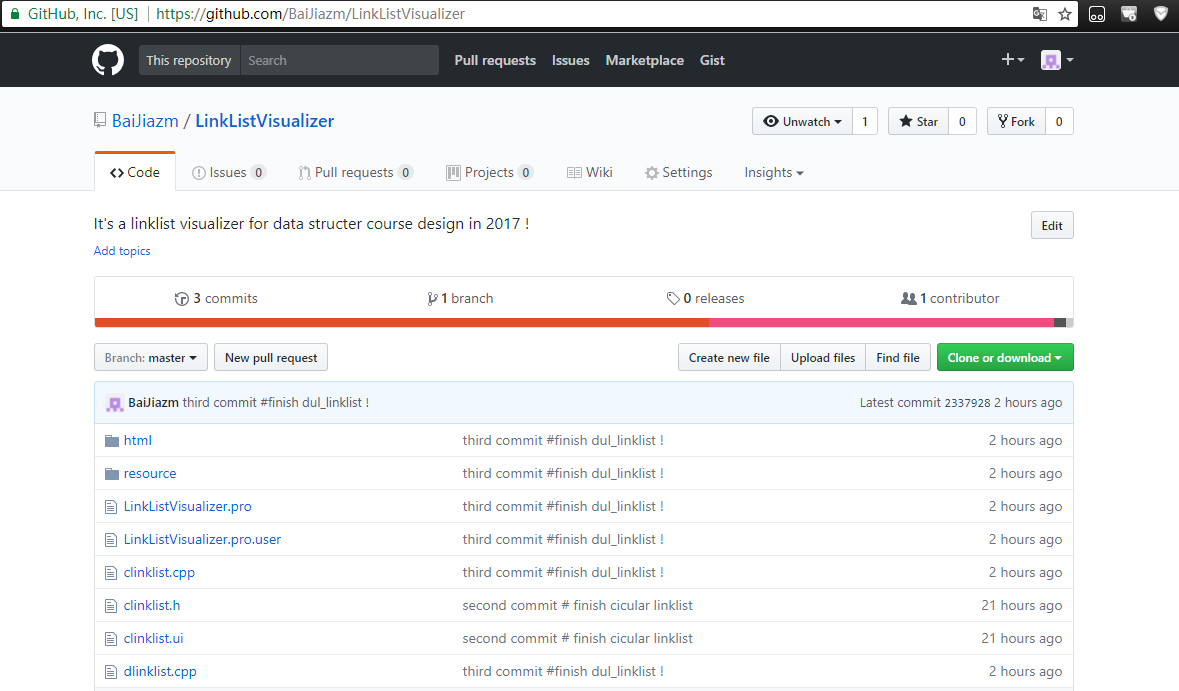
输出：



**1.6.1.2 开发**

本题目虽然有三个小题，但是三个子题目大同小异，架构类似，所以采用分模块逐步开发方式。先确定主窗口，控制好整体界面和架构，然后完成最简单的单链表设计和开发，单链表完成以后整个题目大部分工作已经完成，剩下循环链表和双向链表只需要简单修改就可完成。

本次开发用到了项目控制平台github，分为了三次提交，网址为<https://github.com/BaiJiazm/LinkListVisualizer>

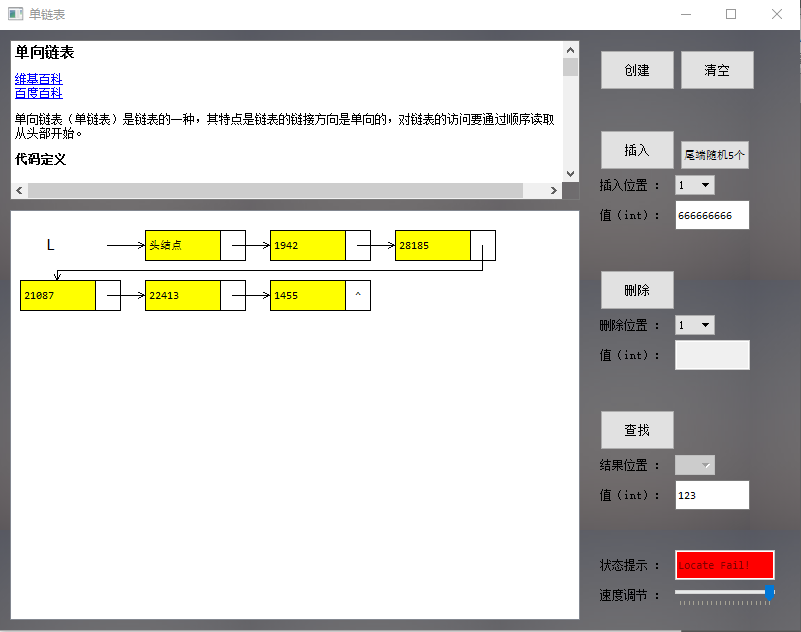
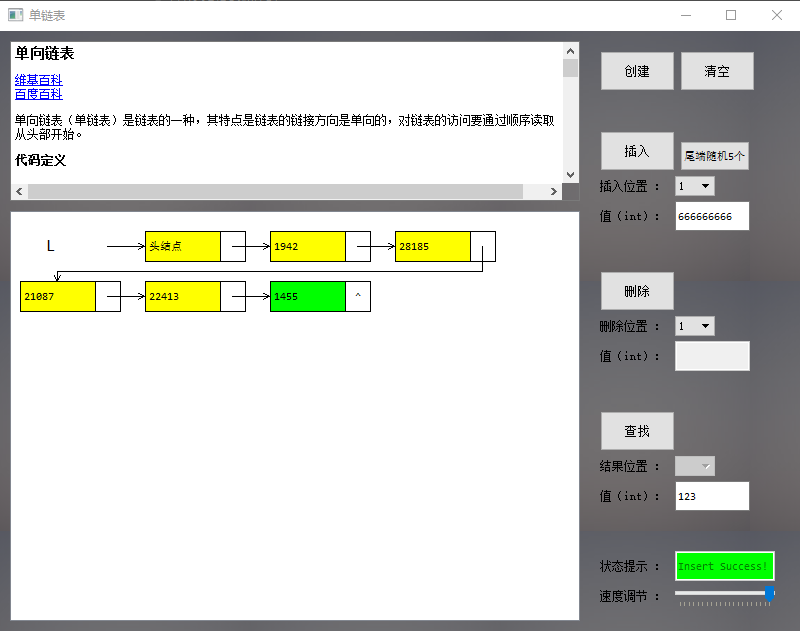


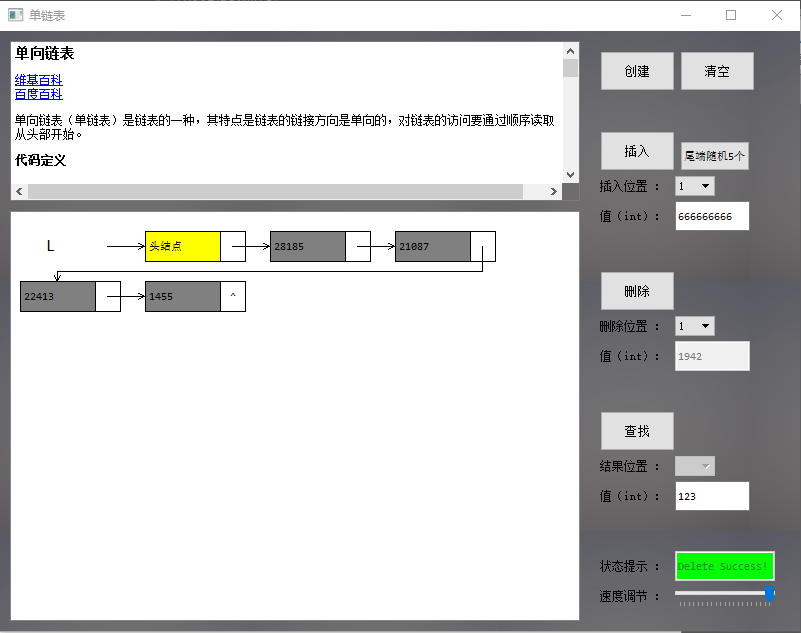
**1.6.2 成果分析**

**1.6.2.1 正确性**

经过多次不同角度的验证，程序表现的十分优秀，与预期没有任何差错。

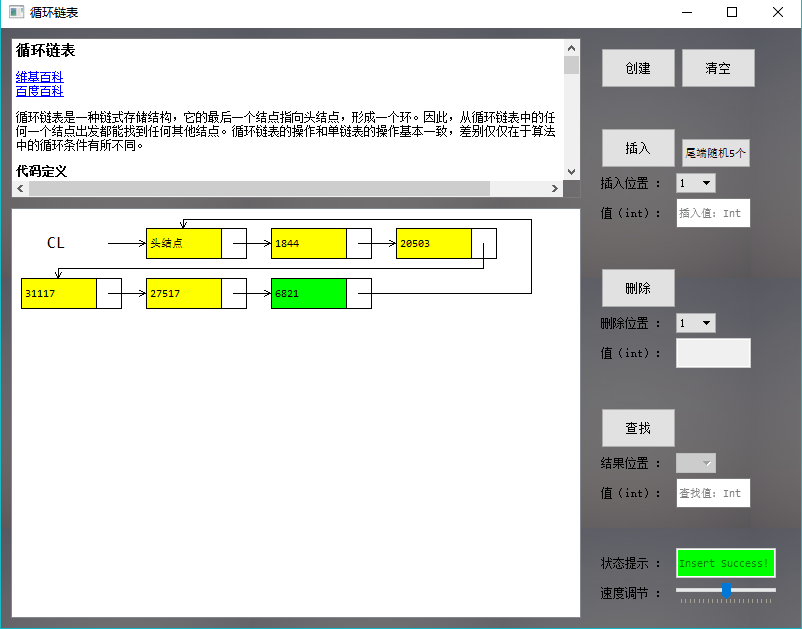
**单链表运行样例图：**

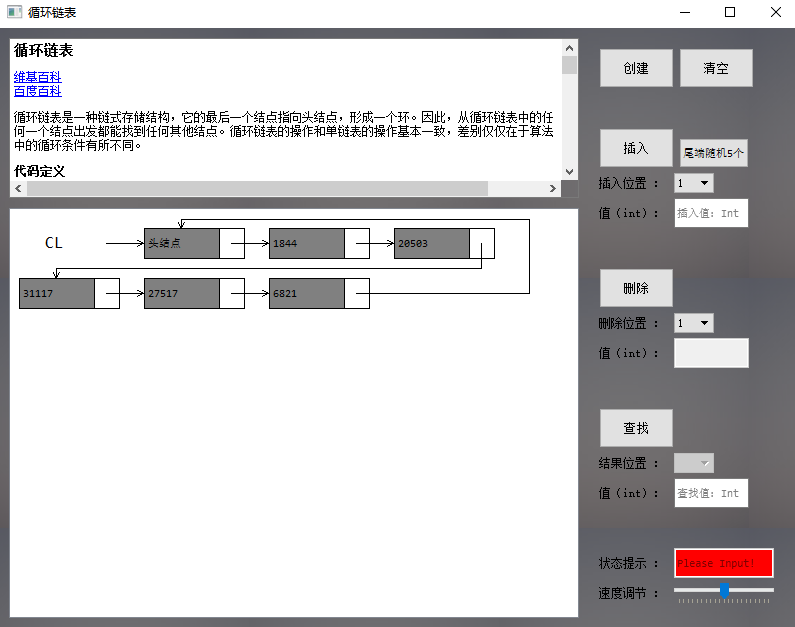


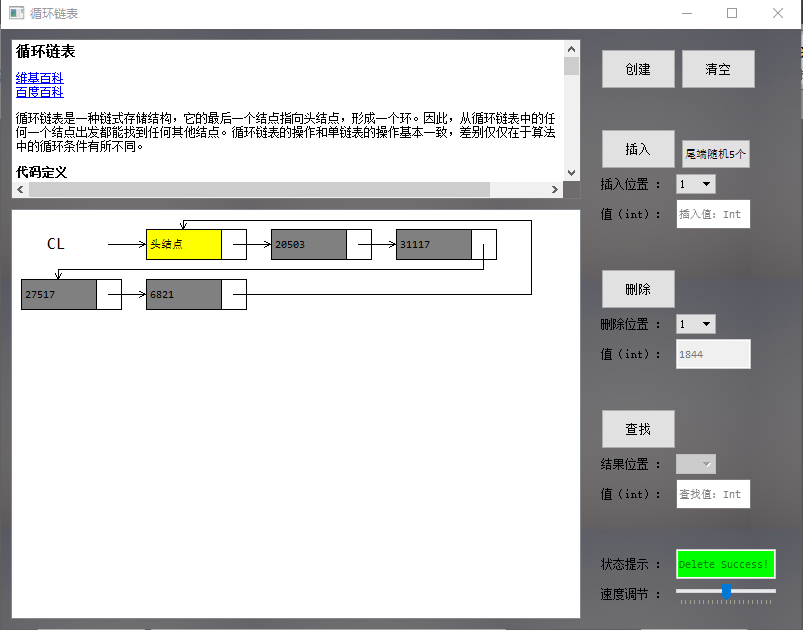


**循环链表运行样例图：**



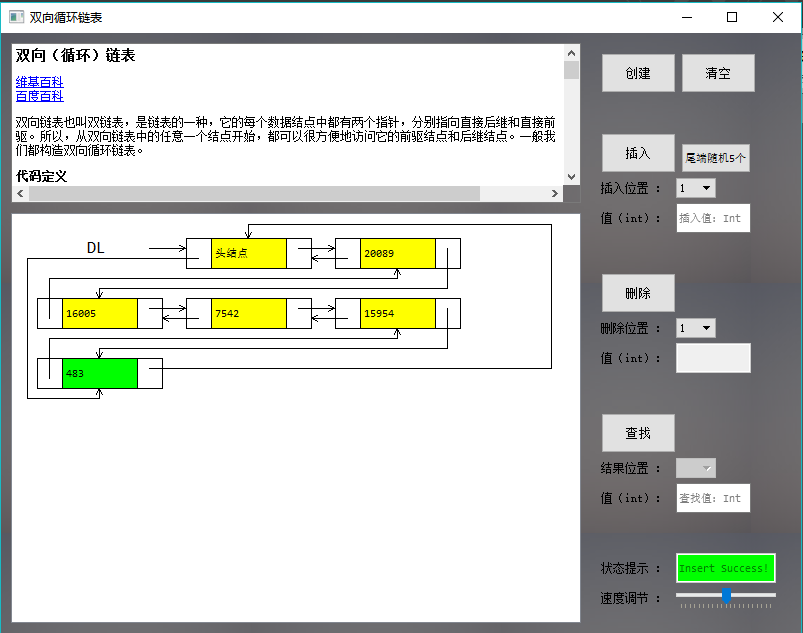


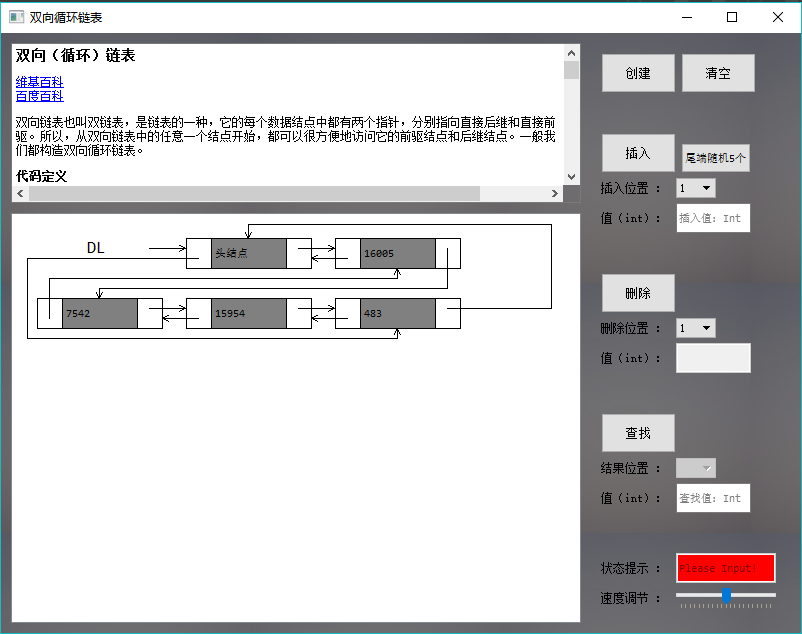




**双向（循环）链表运行样例图：**





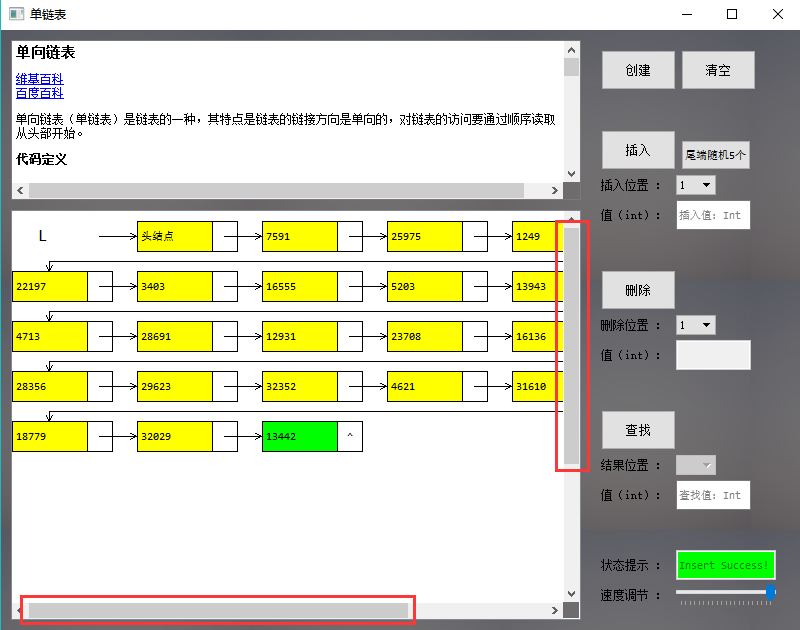


**1.6.2.2 稳定性**

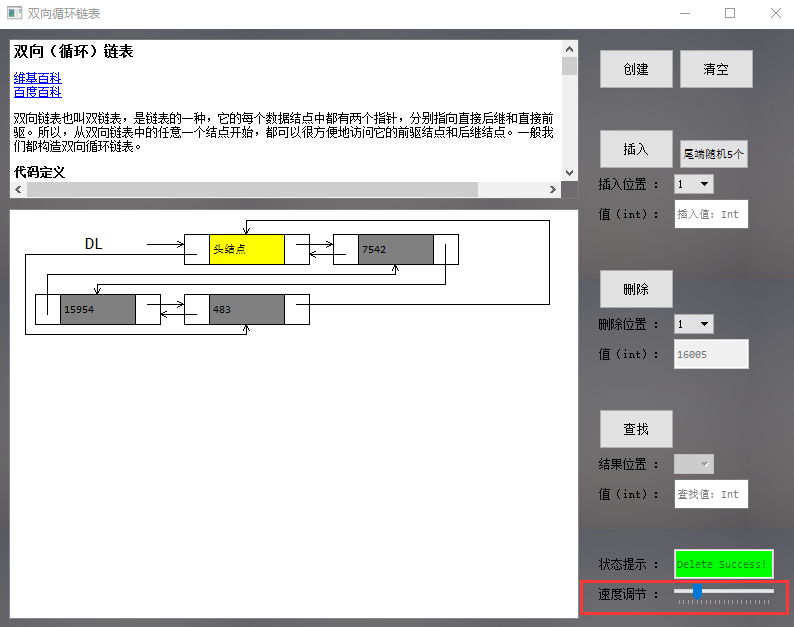
1.图形的大小均采用了宏定义，可根据需要随时调整。



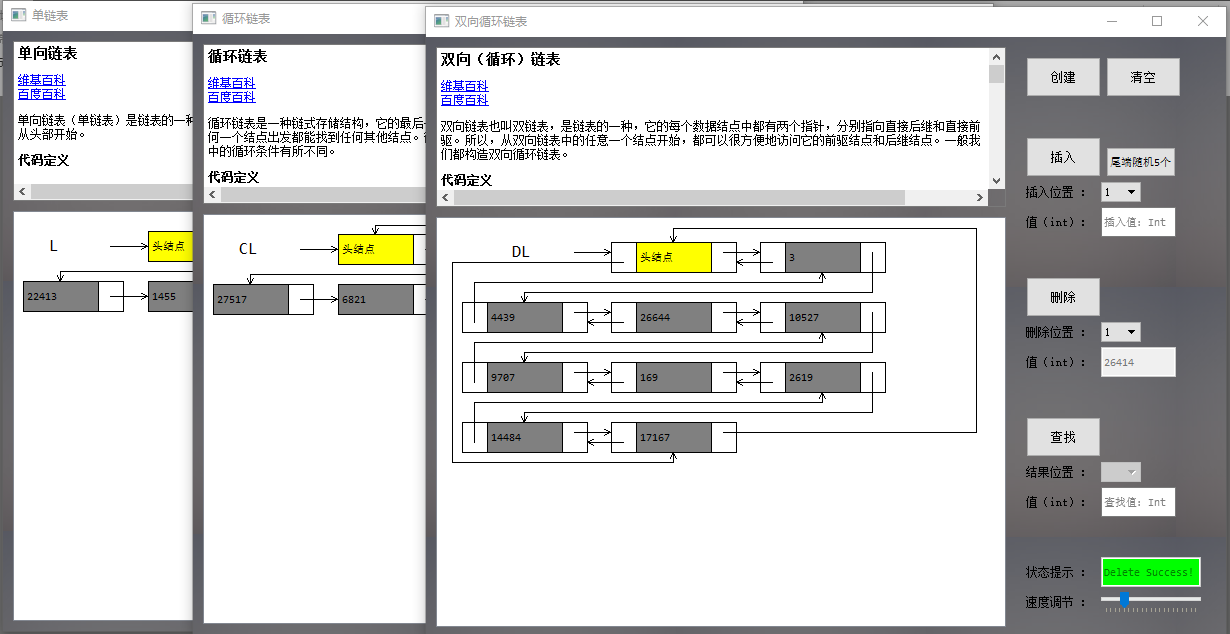
可通过下拉框和左右拉框看到调整大小和未显示的区域。



2.在不同速度调节下，不同速度均表现的十分稳定。



3.三个小部件均能同时稳定正确运行。



**1.6.2.3 容错能力**

本程序有非常好的容错能力，如下：

1. 按钮只有在能够进行操作的情况下，才会处于有效状态，

例如，初始时，只能“创建”，不能“插入”、“删除”等操作；

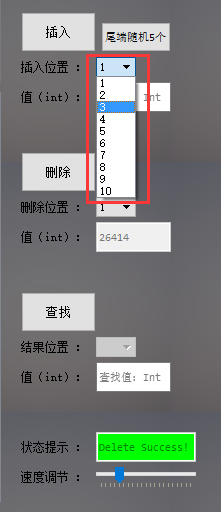


在没有节点时，不能“删除”；



1. 输入利用控件保证输入的正确性；

插入删除位置通过选择保证正确；



输入的合理性通过输入控件保证，当要求输入数字时，输入其它非法字符将不会处理，输入范围在【-999999999,999999999】，其它范围是无法输入的。

* 1. **操作说明**

本题目分为了三个部分，分别是单链表、循环链表、双向链表的基本操作演示。通过前面介绍的运行环境下进行程序的运行。

## 1.7.1 主界面操作

**1.7.1.1 运行**

可直接点击下图所示的打包程序进入运行。

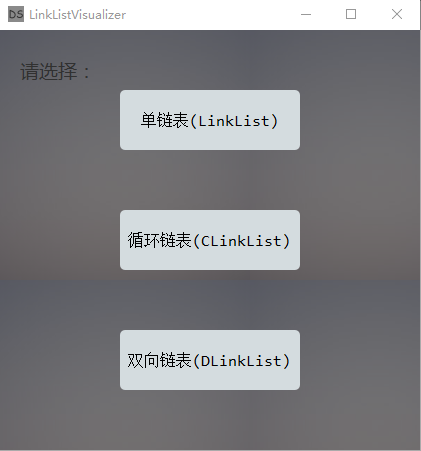


进入主界面

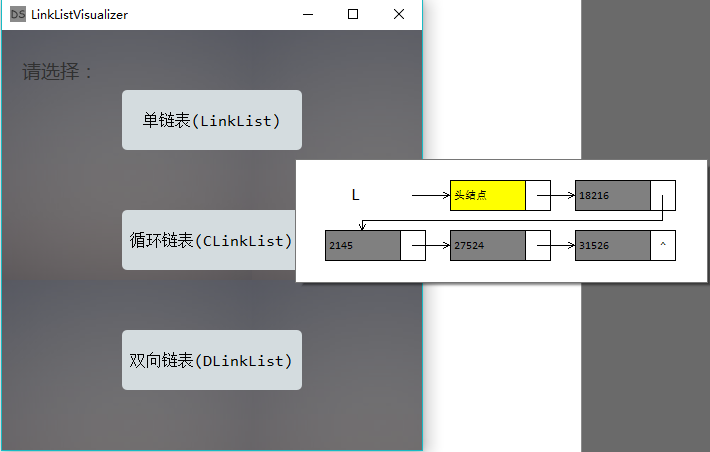


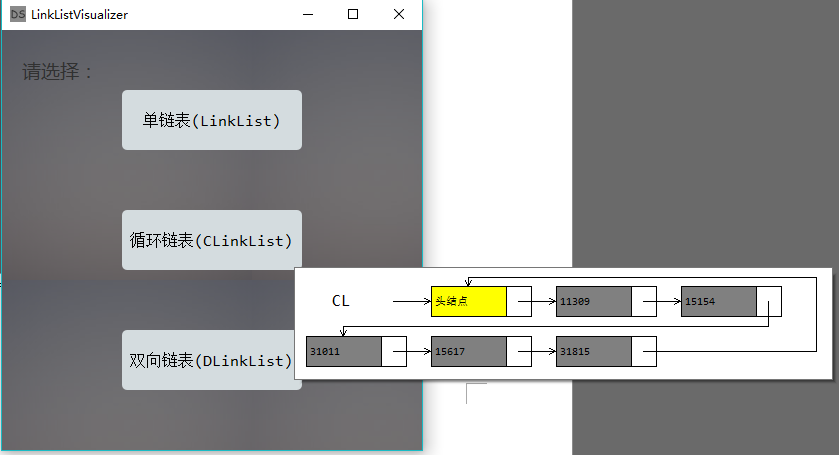
**1.7.1.2 主界面**

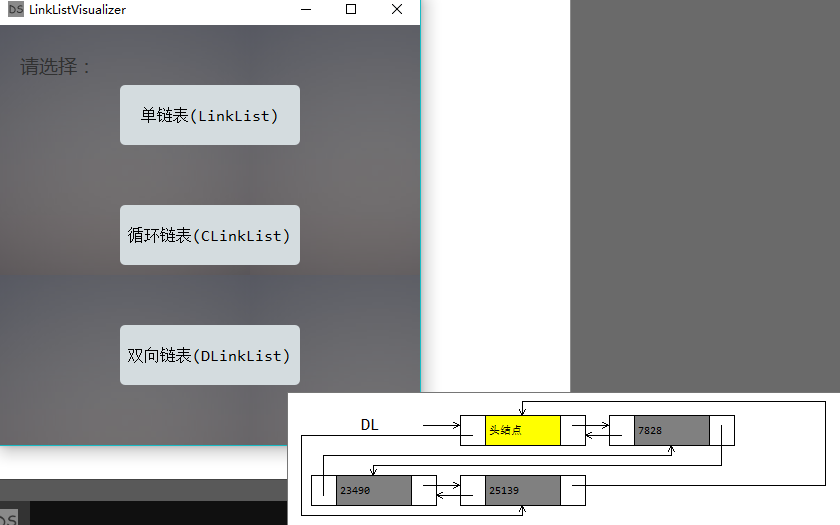
按下相应按钮进入相应功能。



将鼠标悬停在相应按钮上可得到相应提示。







## 1.7.2 单链表

**1.7.2.1 进入功能区**

点击主菜单的“单链表”进入



可以看到初始界面：

按钮或输入控件较亮表示可以点击或输入，否则处于不可编辑状态。

**1.7.2.2 调节演示速度**

右下角有操作后的状态提示及过程显示的速度快慢调节器，状态提示不可编辑，速度调节器可以通过鼠标直接“点击”或者点击后按“左右方向键”进行速度调节，最右段为最快速度。



**1.7.2.3 创建单链表**

点击创建按钮可以创建一个单链表，或者在已经创建一个单链表后销毁并重新创建新的单链表。



提示创建成功。

**1.7.2.4 插入节点**

插入节点有两种方式，可以手动插入，也可自动插入。

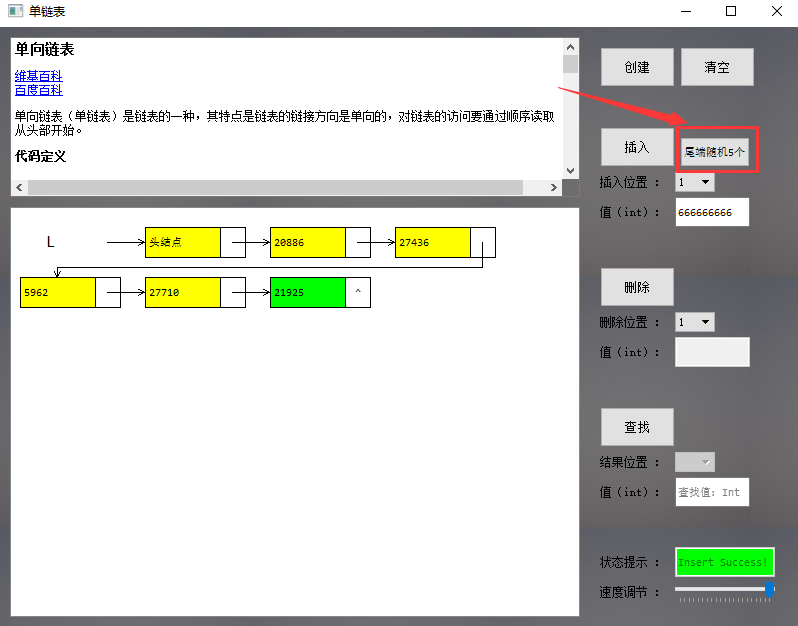
手动插入节点方式：

1. 点击插入位置下拉框，选择插入位置（从1开始）；
2. 在插入文本框中输入插入值；
3. 点击插入；



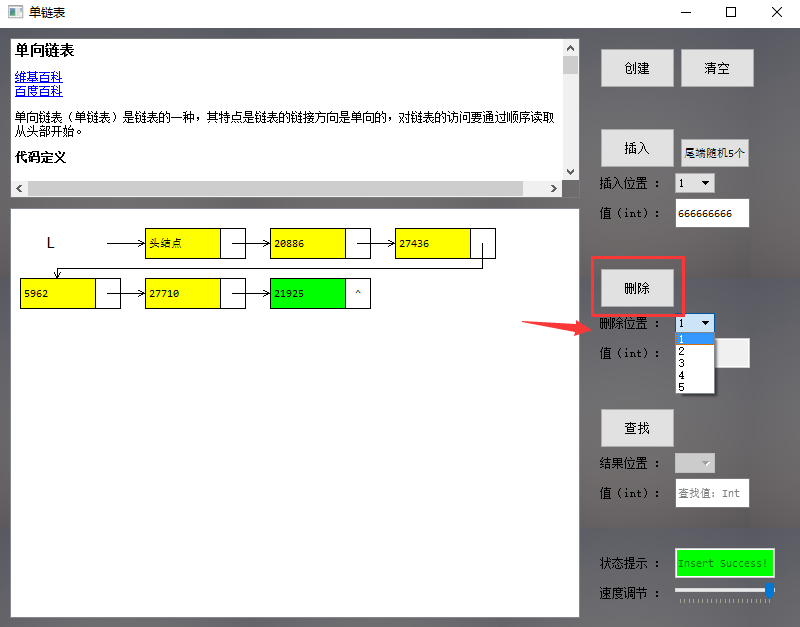
自动插入节点方式：

点击“尾端随机5个”按钮，即可自动插入五个随机值到链表尾部去。

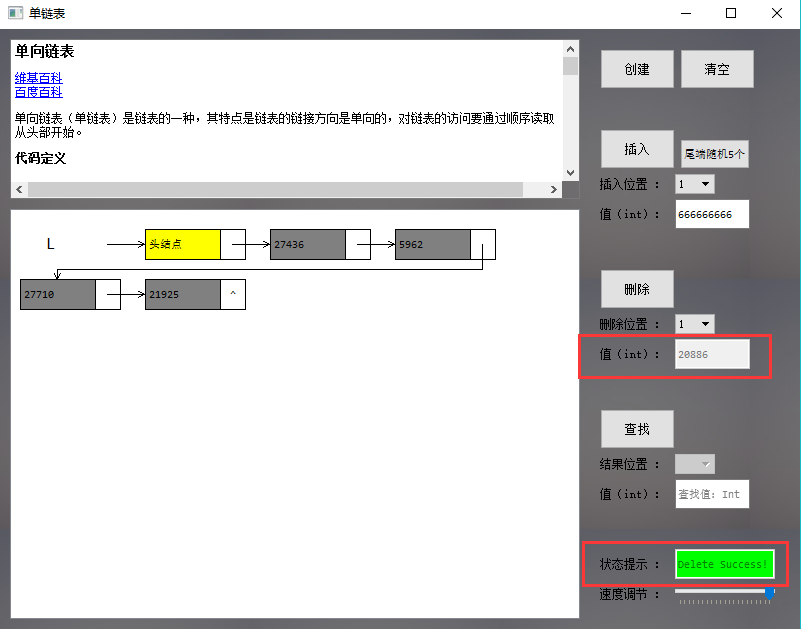


**1.7.2.5 删除节点**

点击删除位置下拉框，可选择相应的删除节点位置，然后点击删除按钮。

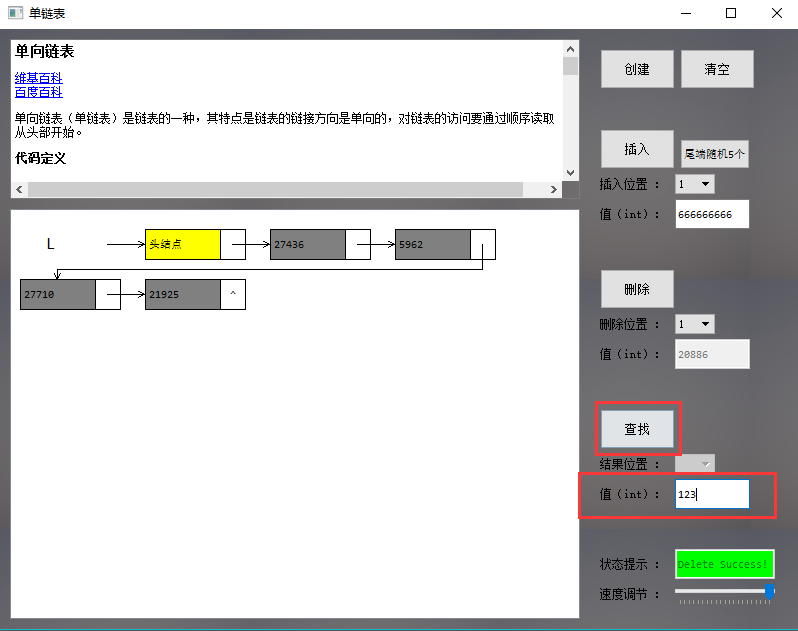


删除后

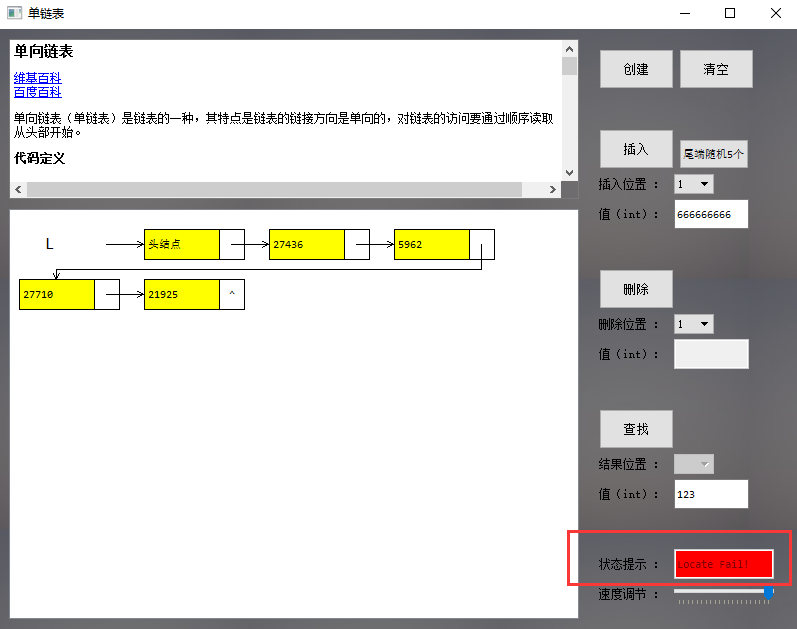


**1.7.2.5 查找节点**

在查找值中输入要查找的值，点击“查找”按钮。

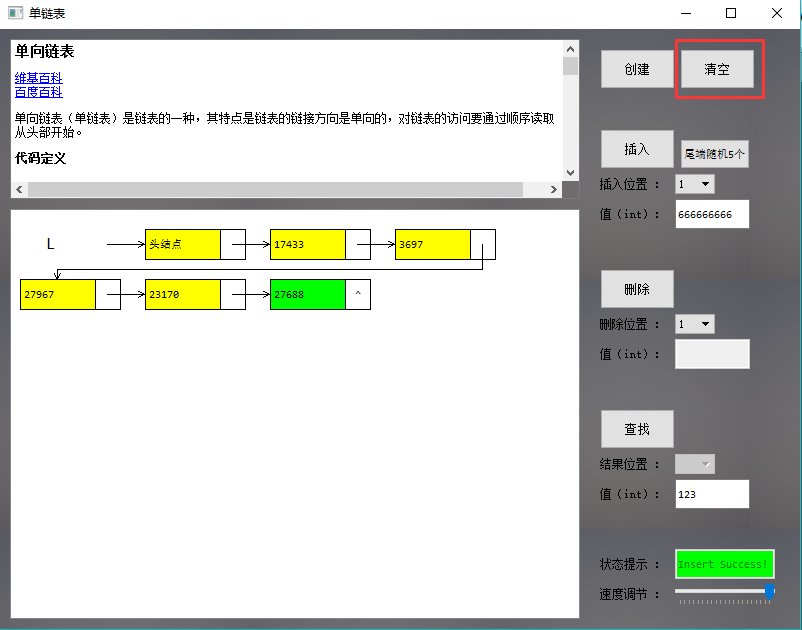


查找结果：



**1.7.2.6 销毁链表**

点击清空按钮，清空单链表，释放资源。



## 1.7.3 循环链表

循环链表的操作与单链表完全一样，不再赘述。

## 1.7.4 双向（循环）链表

双向循环链表的操作与单链表完全一样，不再赘述。

**第二部分 综合应用设计说明**

1. 1. **题目**

题号2★★★：上海的地铁交通网络已经基本成型，建成的地铁线十多条，站点上百个，现需建立一个换乘指南打印系统，通过输入起点和终点站，打印出地铁换乘指南，指南内容包括起点站、换乘站、终点站。

1. 图形化显示地铁网络结构，能动态添加地铁线路和地铁站点。
2. 根据输入起点和终点站，显示地铁换乘指南。
3. 通过图形界面显示乘除路径。
   1. **软件功能**

功能主要功能包含地铁网络线路的查看，地铁换乘指南查询，动态添加线路站点等。

**线路图查看：**

●查看上海地铁网络线路图，包括通过键盘、鼠标拖放、放大缩小等简易查看操作；

●查看地铁线路信息，包括线路段、包含站点等；

●查看地铁站的详细信息，包括站点地理坐标、所属线路等

**换乘指南查询：**

●提供地铁换乘查询，可通过视图方便的查看乘坐路线和换乘路线；

●提供最小出行时间的换乘策略指南

●提供最小换乘次数的换乘策略指南

**动态添加线路：**

●动态添加线路，可根据需要新增线路；

●动态添加站点，可根据需要新增站点；

●动态添加连接，可根据需要新增站点连接；

●文本方式简易添加，方便快捷；

**帮助：**

●提供软件使用帮助和说明；

**关于：**

●提供软件制作信息；

上述所有功能采用面向对象的方法通过C++语言程序结合QT框架实现，后面会详细介绍。

* 1. **设计思想**

**实现思路**

**1.系统分析，明确功能要求**

在做应用程序之前都会先明确软件的需求是什么，知道了要实现那些功能才能去细化用什么方法实现，实现的系统架构是什么，初步规划程序的基本流程和核心算法。在认真读题后，明确功能要求后，开始下一步操作。

**2.MVC模式系统架构**

MVC模式即为模型-视图-控制器结构模式，这种软件架构模式是一种常用的设计。在本次系统结构中，大致采用此种模式，前后端实现分离，前端主要考虑与用户的交互，例如设计怎样的视图，地铁网络线路图如何展现等，后端主要是数据和算法的处理，将复杂的功能和数据交由后端处理，前后端实现函数接口，从而完成整个软件系统的架构设计。

**3.设计后端数据结构和接口**

本次综合应用程序，基本上的数据结构都有涉及，例如网络线路的连接情况需用图结构、算法实现有队列，静态存储有线性表，名字到存储位置有哈希映射，分析好数据结构之后，根据需要将后端系统的接口进行设计，例如最短路径的查询、添加线路站点等均需接口。

**4.分块实现，逐步求精**

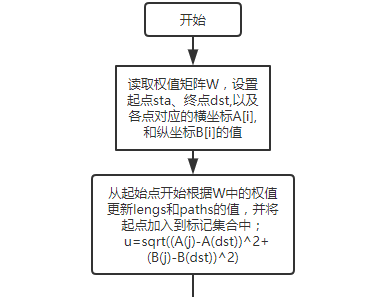
系统结构和数据结构、算法等相应设计好后，就开始逐步实施了，从前端开始，对每个功能要求的前端界面和交互进行细化，实现前端代码，对每个功能的前端实现后，完善后端接口函数，完成所有前端后，后端也基本同步完成，从而完成整个程序的实现。

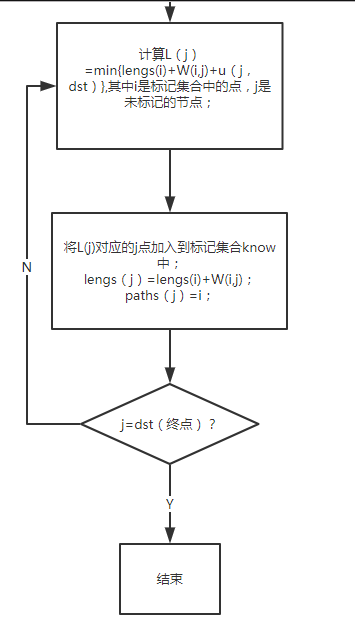
**数据结构分析**

本次程序设计站点、线路及其关系的存储表示和相应的算法，所有线路和站点名字是唯一的，表示线路的网络关系时，每个站点可看做一个节点，站点的联系看做一条边，网络结构显然用图结构存储，对每个站点和线路来讲，不会有删除操作，会有添加要求，最频繁的是查询需求，所以用顺序表存放最为合适，将站点和线路名和存储位置的下标值进行哈希关联，可方便访问。

核心算法流程

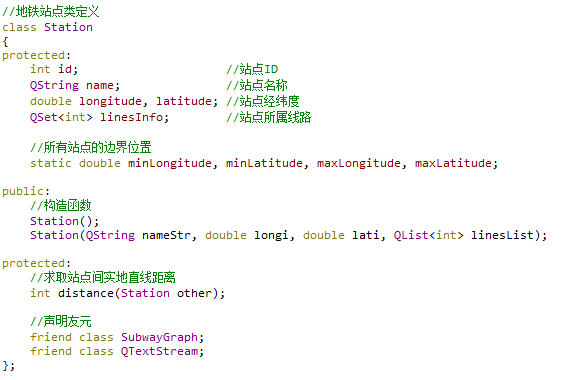
迪杰斯特拉算法求短距离换乘：





* 1. **逻辑结构与物理结构**

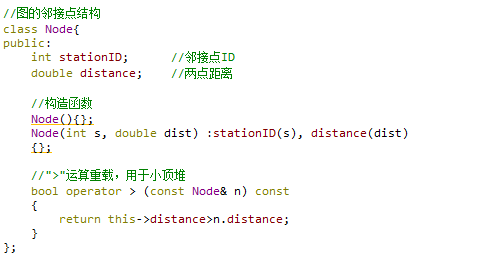
## 2.4.1 站点

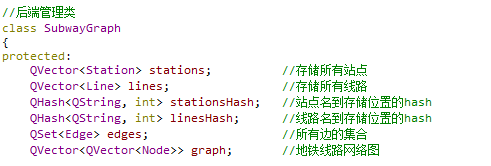


## 2.4.2 线路

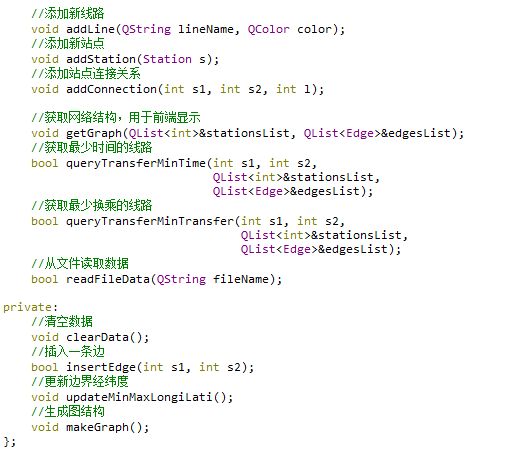


## 2.4.3 后端实现类









* 1. **开发平台**

**开发平台：**

计算机型号 ：惠普Pavilion M4

计算机内存 ：4.00GB

CPU ：Intel Core i5 2.6GHz

操作系统 ：Windows 10 家庭版

开发语言 ：C++（C++11标准以上）

开发框架 ：QT

集成开发环境：Qt Version 5.9.1

编译器 ：MinGW 32bit

**运行环境：**

可在上述集成环境下运行；

通过windeployqt.exe及Enigma Virtual Box进行整合压缩为发布为了一个LinkListVisualizer.exe文件，可在普通windows机型下运行。

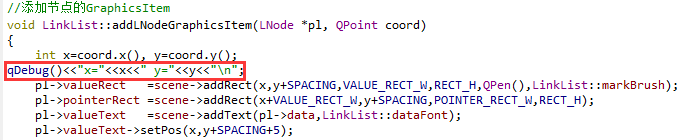
* 1. **系统的运行结果分析说明**

**2.6.1 调试及开发过程**

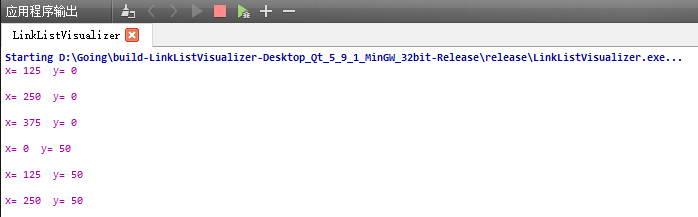
**2.6.1.1 调试**

本次开发采用的是新技术框架Qt，同时也是跨平台的，在Qt Creator中开发调试，Qt中包含了大量的库类，类似于java开发简便，Qt有较好的调试器，但是在本次开发中没有用到，偶尔遇到一些小麻烦或者小bug，我们只需在控制台中输出一些数据便可分析定位错误原因。

例如：



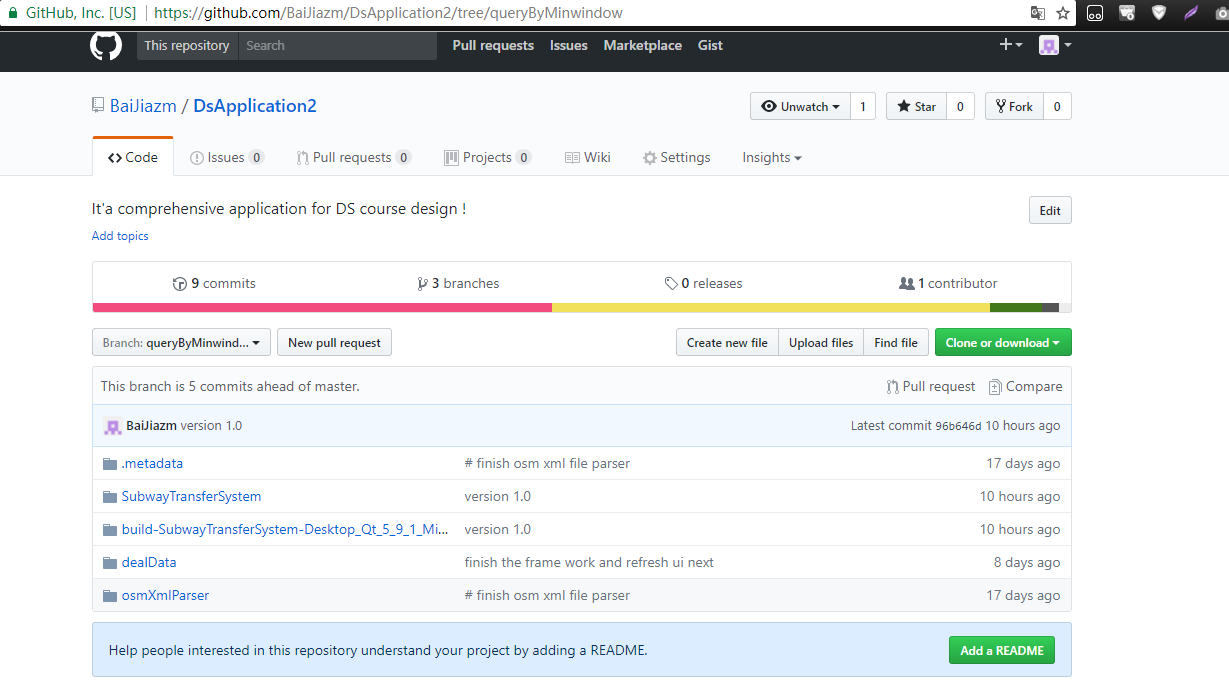
输出：



**2.6.1.2 开发**

有了前面Qt数据结构算法实现设计题的基础，对Qt的了解也由浅入深了，开发也相对熟练。开发时，遵循将后端系统和前端用户交互分开，先思考后端系统的核心数据结构的表示和存储，如何实现题目所要求的相应功能，设计核心算法，然后设计了前端和后端的程序接口，搭建好整个综合应用程序的主体框架，最后是分布实施，将各个模块由大化小，逐步求精，完成整个程序的开发。

本次开发用到了项目控制平台github，网址为<https://github.com/BaiJiazm/DsApplication2/tree/queryByMinwindow>



**2.6.2 成果分析**

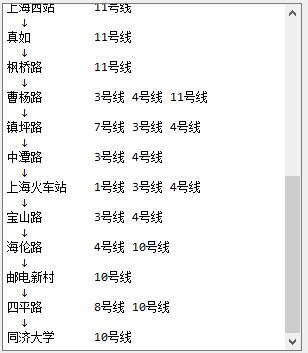
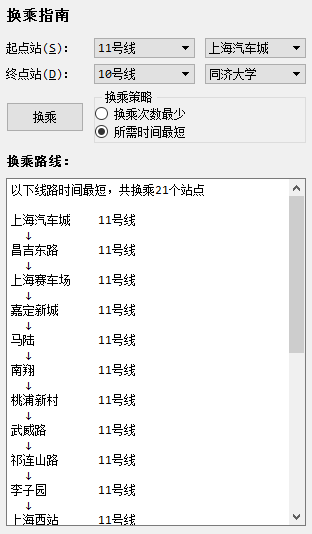
**2.6.2.1 正确性**

经过多次不同角度的验证，程序表现的十分优秀，与预期没有任何差错。

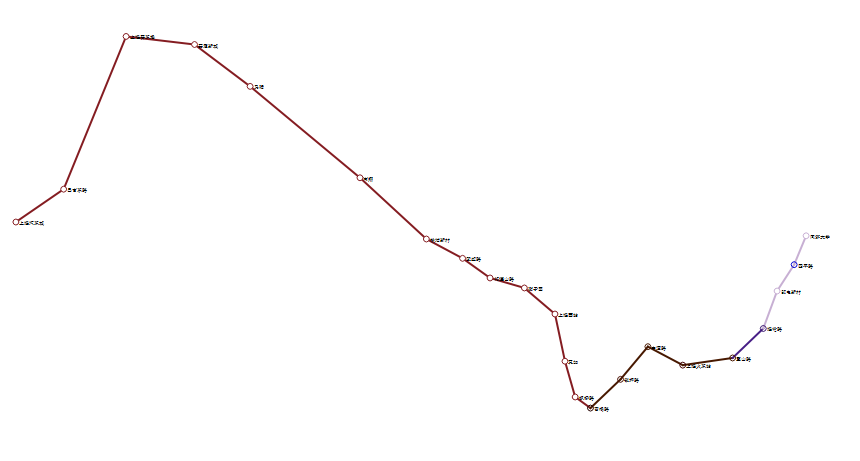
**换乘查询：**

**最短时间查询：**

例如选择“上海汽车城”到“同济大学”，线路如下：

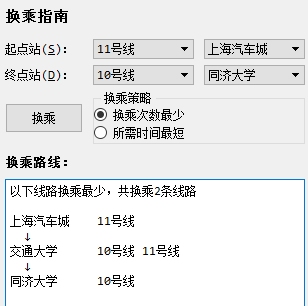


**视图显示线路：**

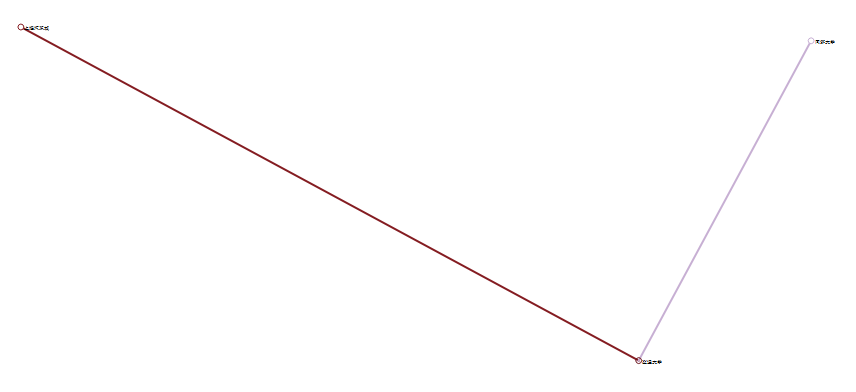


**最少换乘查询：**

例如选择“上海汽车城”到“同济大学”，线路如下：

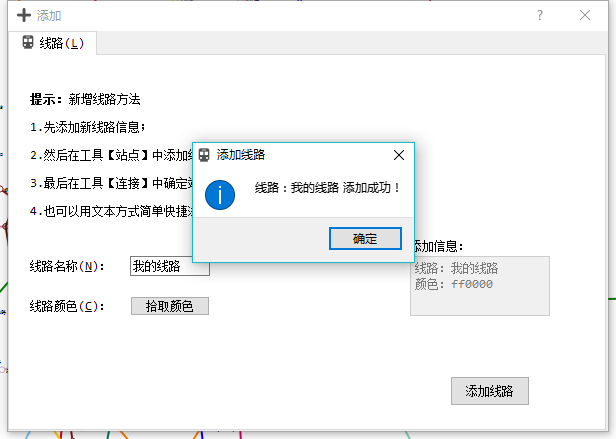


**换乘显示：**

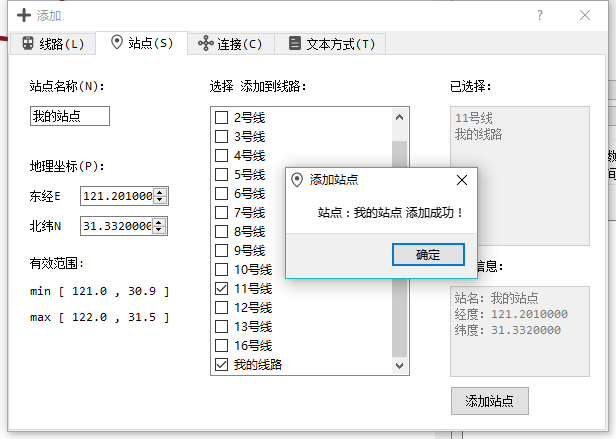


**一系列动态添加：**

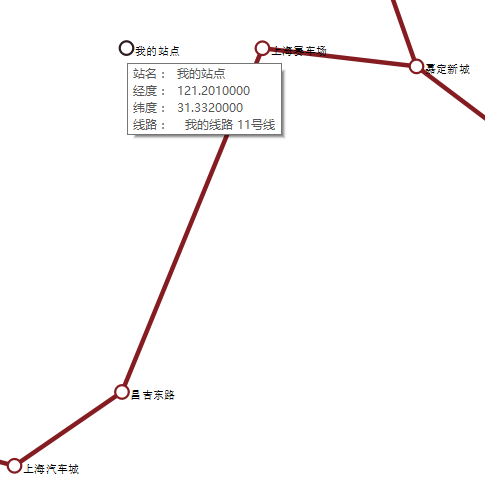
**添加新线路成功：**



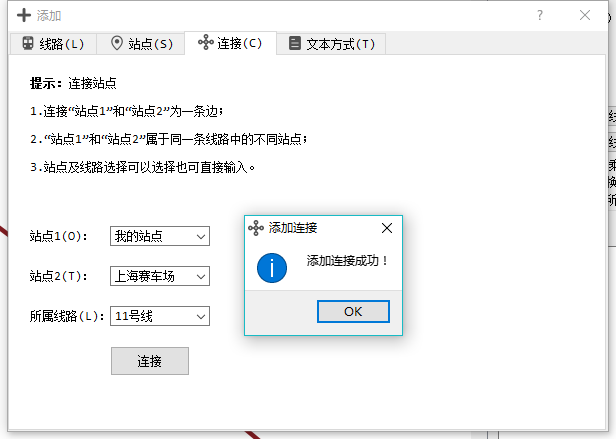
**添加新站点成功：**



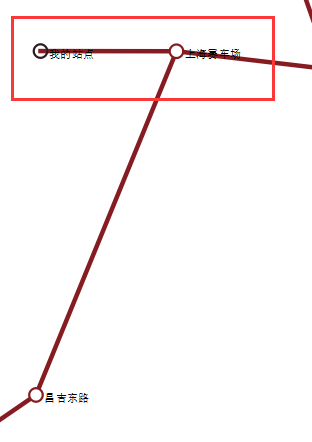
**地图显示新加站点：**



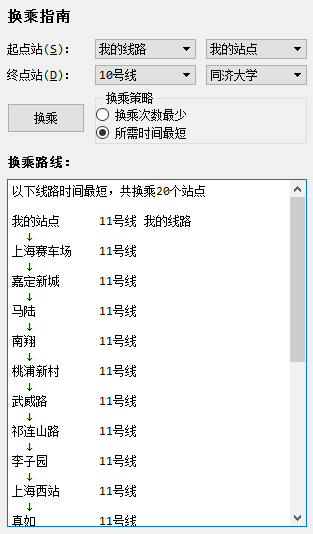
**添加新连接成功：**



**地图显示：**

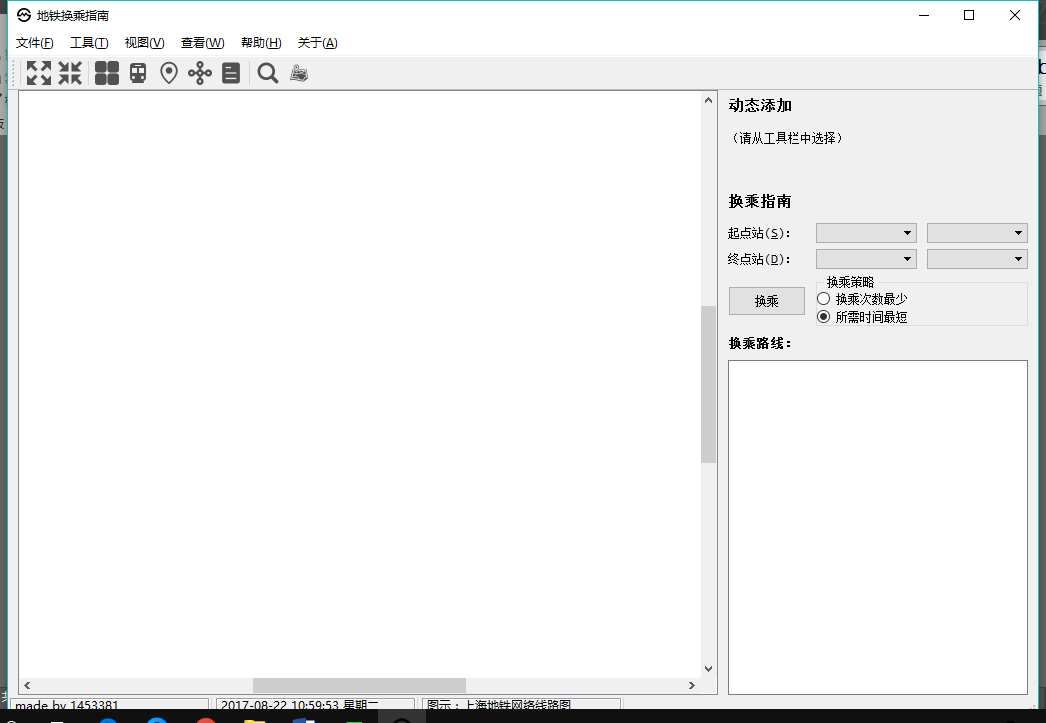


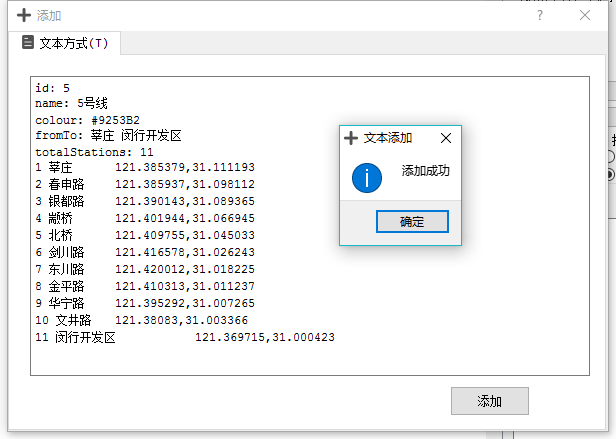
**换乘检验：**



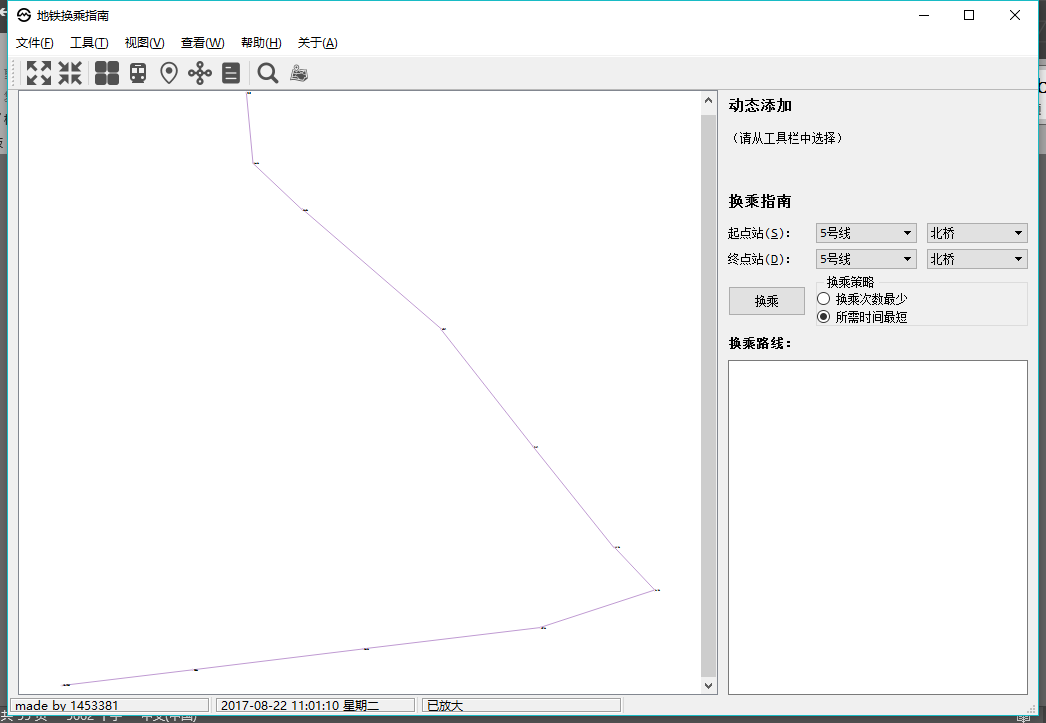
**文本添加：**

现无数据，通过添加，得到线路：





添加后显示：



**2.6.2.2 稳定性**

1. 无论是在放大窗口或是小窗口程序都未有任何异常，视图和右侧栏会随窗口大小自动调节；

2. 运行速度非常快，所有操作都是在1s之内响应和得出结果；

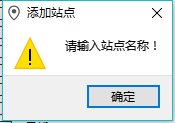
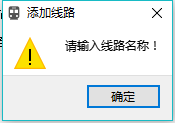
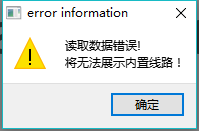
3. 所有操作都会及时更新，例如：添加新的线路和站点后，所有用户接口界面中的数据都会进行立马更新，视图也会立马显示新加线路和站点；

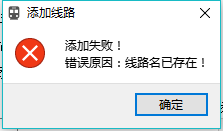
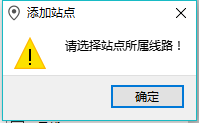
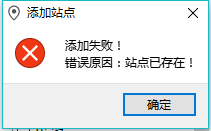
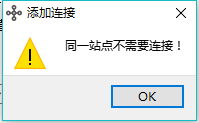
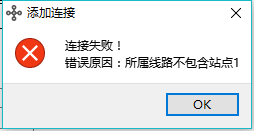
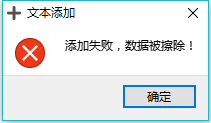
4. 经过多次调试和反复测试，程序没有任何异常情况。

**2.6.2.3 容错能力**

本应用程序做了很多针对用户输入和非法操作的容错处理，无论用户怎样操作，很难出现一些非法情况，即使有，也会被程序处理，固有非常好的容错能力，一些样例如下：

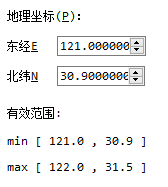
1.非法操作均有提示：



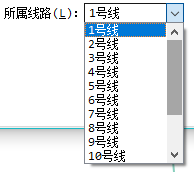
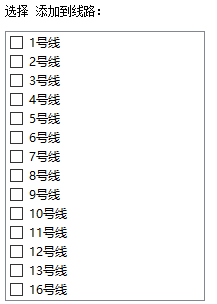
     

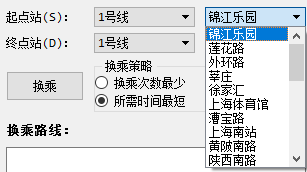
2.输入利用控件保证输入的正确性；

输入的合理性通过输入控件保证，例如输入经纬度时，输入其它非法字符将不会处理，输入范围被确定，其它范围是无法输入的。



添加或选择只能从指定数据中选择，





* 1. **操作说明**

按照题目的要求，本应用程序主要包含两大功能，换乘查询和动态添加线路。通过前面介绍的运行环境下进行程序的运行。

## 2.7.1 视图查看

1.视图放大：

点击工具栏中放大按钮；

或选择“视图”下“放大”；

或按下快捷键“Ctrl+L”；

或直接在视图区域进行鼠标滚轮放大操作

可进行视图的放大，方便查看。



2.视图缩小：

点击工具栏中缩小按钮；

或选择“视图”下“缩小”；

或按下快捷键“Ctrl+S”；

或直接在视图区域进行鼠标滚轮缩小操作

可进行视图的缩小，方便查看。



3.鼠标操作

鼠标除能直接放大和缩小视图外，在查看视图信息有重要作用：

鼠标滚轮直接放大缩小视图；

通过鼠标点击后拖动，可直接进行视图移动；

将鼠标悬停在站点上，会提示站点的详细信息，如悬停在“曹杨路”站上，会有曹杨路地铁站的详细信息；

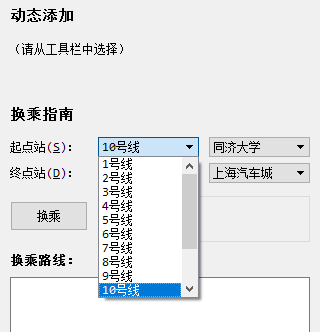


将鼠标悬停在线路上，会提示线路的详细信息，如悬停在“曹杨路”站和“镇坪路”之间线路上，会有线路的详细信息；



## 2.7.2 换乘查询

1.选择起始和目标站点，直接选择即可，如下：

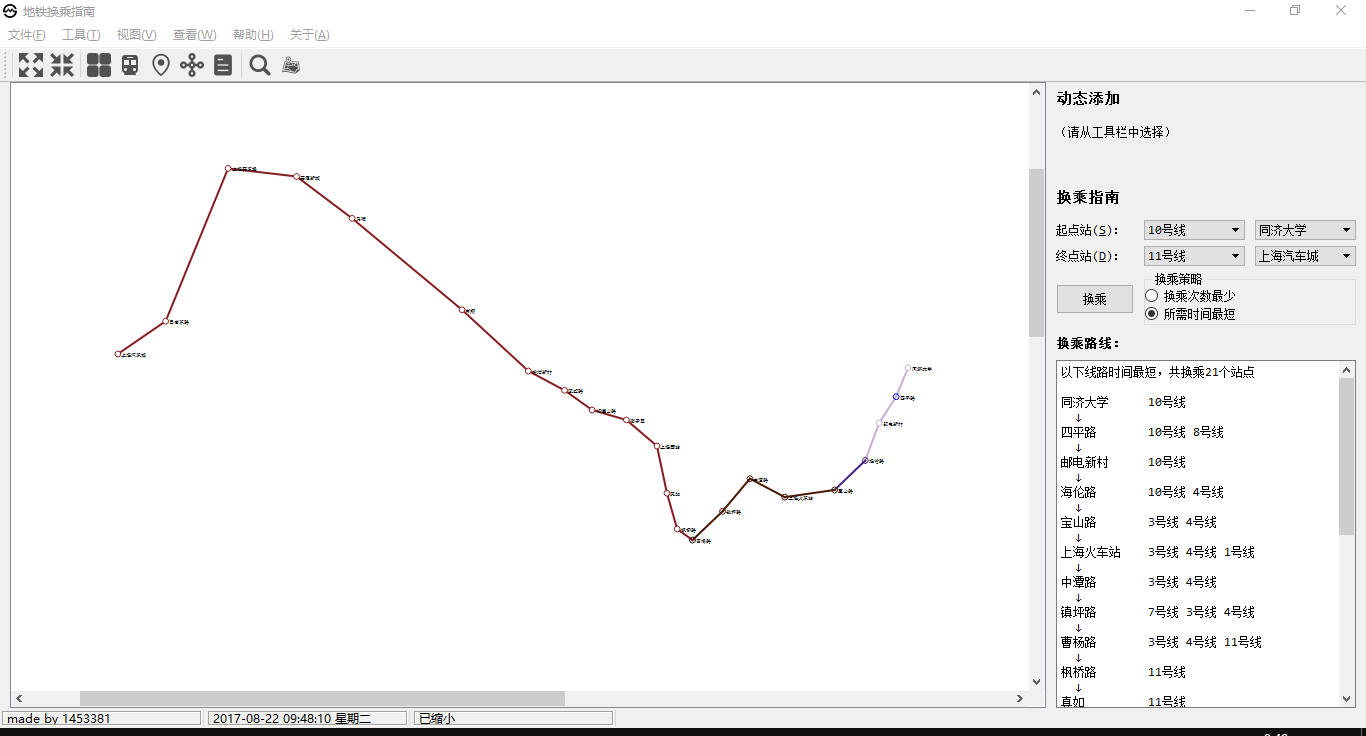


2.最小时间换乘查询：

选择“起点站”、“终点站”、“换乘策略”为“所需时间最短”后，点击换乘，可得相应线路。

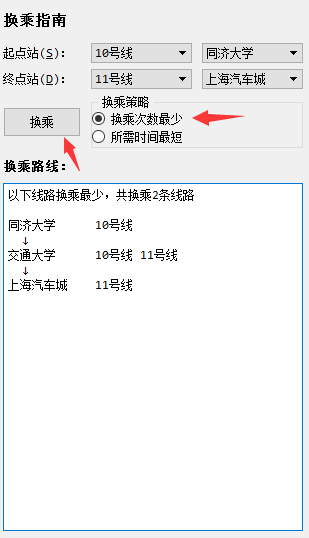


换乘后文本栏会有相应线路，视图区有详细路径，通过前面所述视图查看操作可方便查看。

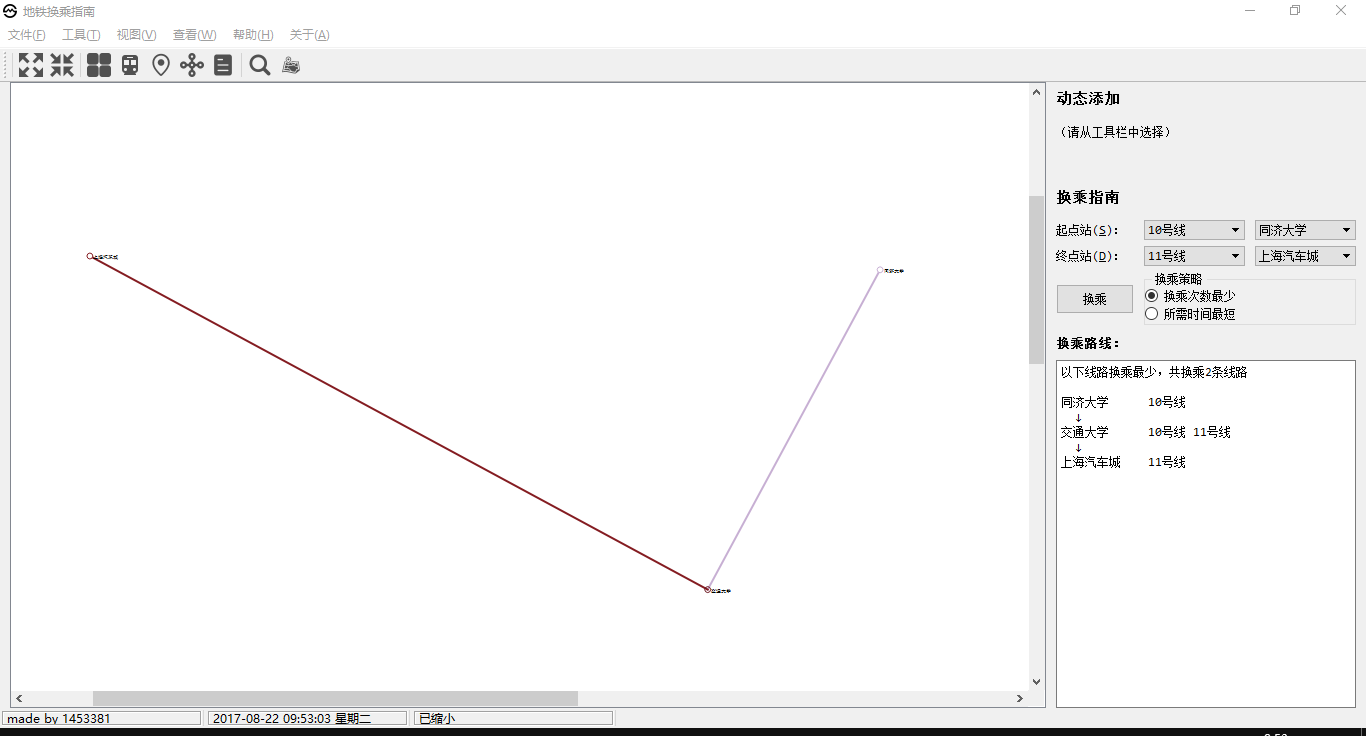


3.换乘次数最少查询：

选择“起点站”、“终点站”、“换乘策略”为“换乘次数最少”后，点击换乘，可得相应线路。



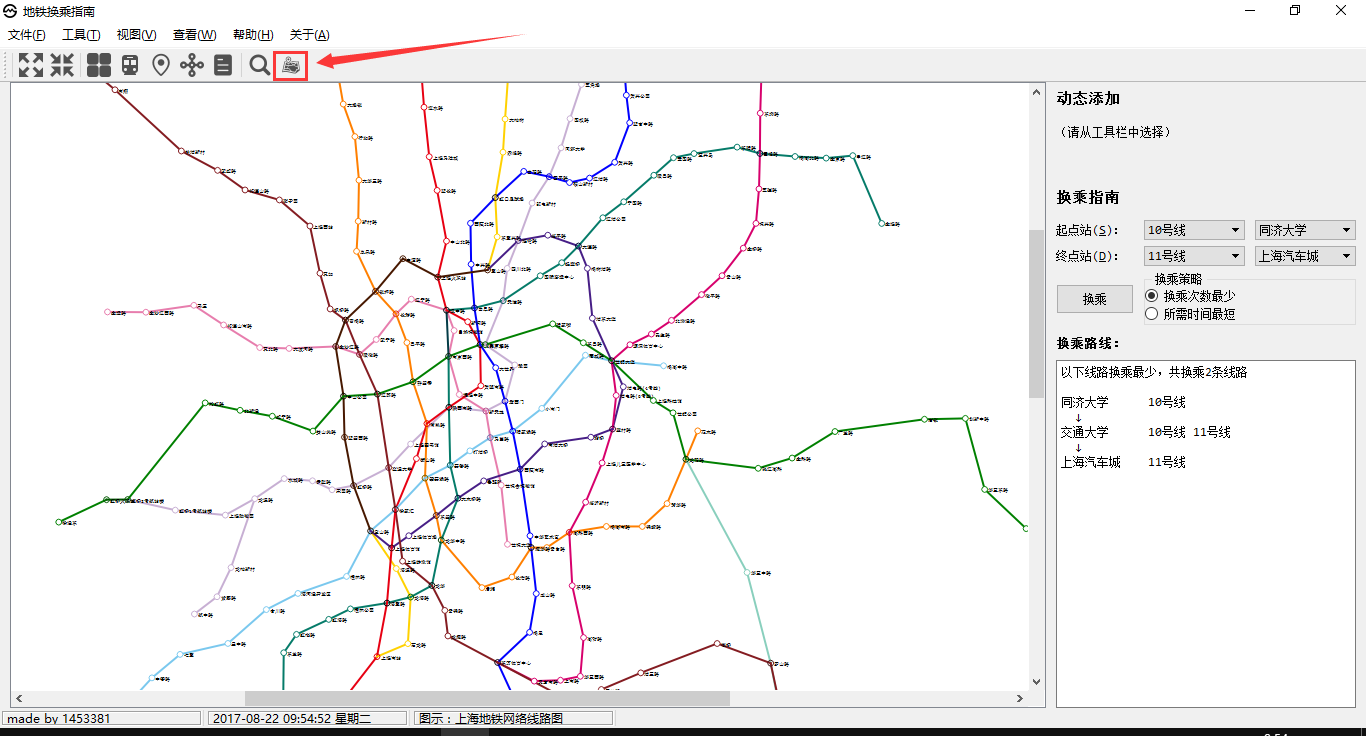
换乘后文本栏会有相应线路换乘方法，视图区绘制出，通过前面所述视图查看操作可方便查看。



1. 换乘查询后，点击下图按钮，

或选择“视图”下“地铁图”，

或按下快捷键“Ctrl+M”可重新查看网络线路图。



## 2.7.3 动态添加

动态添加主要包含添加线路，添加站点，添加连接，文本简易添加，在工具栏中如下：



1. 添加线路

点击工具栏中如下；

或选择“工具”下“添加”中“线路”；

或快捷键“Ctrl+Shift+L”

可添加线路



上述操作后弹出添加对话框，按相应操作可添加线路。



1. 添加站点

点击工具栏中如下；

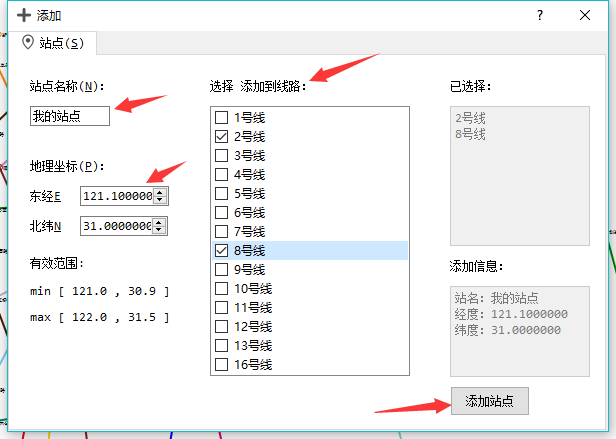
或选择“工具”下“添加”中“站点”；

或快捷键“Ctrl+Shift+S”；

可添加站点



上述操作后弹出添加对话框，按相应操作可添加站点。



1. 添加连接

点击工具栏中如下；

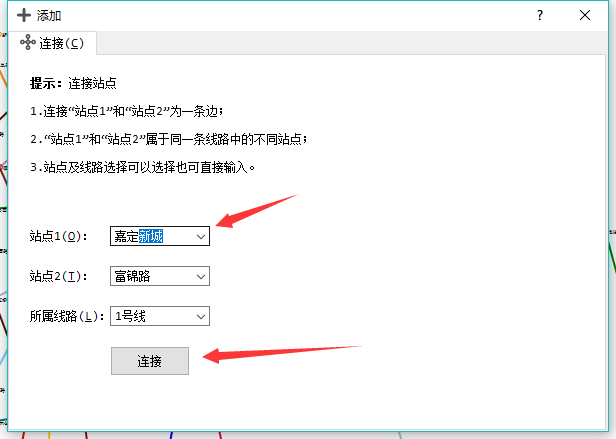
或选择“工具”下“添加”中“连接”；

或快捷键“Ctrl+Shift+C”；

可添加连接



上述操作后弹出添加对话框，按相应操作可添加连接。（其中站点可直接输入快捷查找）



1. 添加连接

点击工具栏中如下；

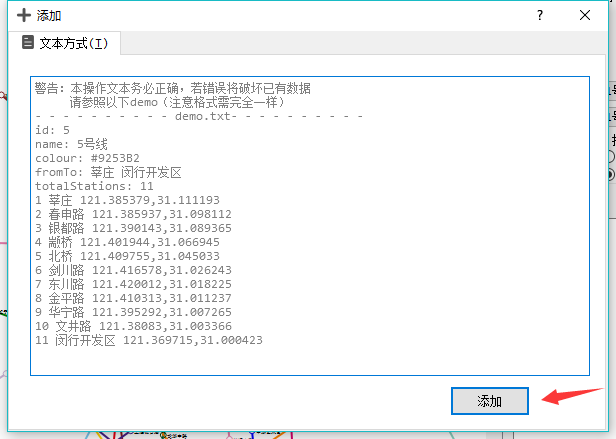
或选择“工具”下“添加”中“文本方式”；

或快捷键“Ctrl+Shift+T”；

可文本添加



上述操作后弹出添加框，按提示输入即可。

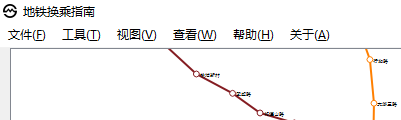


## 2.7.4 工具栏和状态栏

选择“查看”，弹出“工具栏”和“状态栏”选项，通过点击其前选择框，可选择是否显示。



无工具栏，如下：

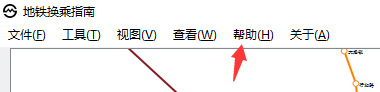


状态栏，如下：

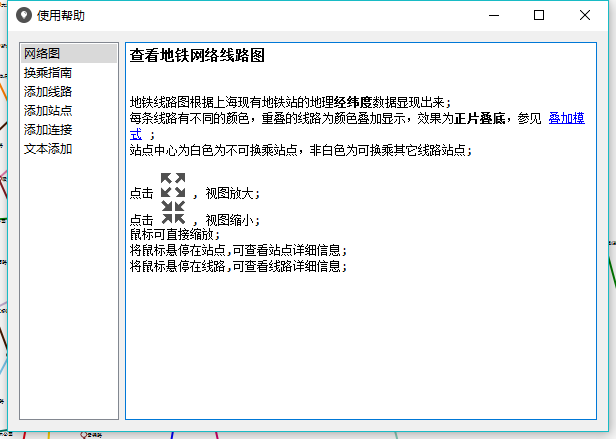


## 2.7.5 帮助

选择“帮助”或按快捷键“Ctrl+H”，可弹出帮助对话框，其中包含详细帮助文档。

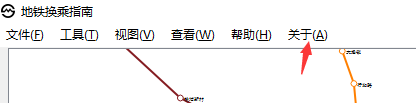


使用帮助对话框：



## 2.7.6 关于

选择“关于”，其下有关于程序和创作者的信息。



## 2.7.7 退出

选择“文件”下“退出”，或快捷键“Ctrl+E”即可退出程序。

**第三部分 实践总结**

1. 1. **所做的工作**

学习和运用了Qt5的编程架构和相关知识；

学习了Github平台项目托管的运用；

独立完成课程设计中的算法实现题目的所有设计、编程实现、测试等工作；

独立完成课程设计中的综合应用题目的所有设计、编程实现、测试等工作；

完成课程设计报告一份

* 1. **总结和收获**

**能力提升：**

完成数据结构课程设计后，感觉自身的自学能力有了很大提升。在课程设计以前，基本上所有的编程知识都是由老师言传身受、一点一滴的对我们进行指导，我们自学的东西较少，自学能力也锻炼的少。但是这次的课程设计的所有工作，都是由我们自己去选择相应的技术学习实现，仅仅掌握基本编程语言的我们通过自学来完成这次课程设计。在这个过程中，大学生尤其是计算机专业的大学生所需要的自学能力得到了必须的锻炼，对以后所有的技术和知识的自学，这将是一个有意义的开端，现在我们已经掌握了如何快速学习一门技术的能力，对以后学习和工作中的所有挑战做好了准备。

**收获：**

虽然课程设计只有看似算法设计和综合应用实现两个题目，但是从布置课程设计题目到完成的整个过程中，不仅仅是学习到了以前未曾接触过的标准用户交互程序，从简陋的控制台程序提升到界面友好、美观、人性化的应用程序，打开了编程的一扇大门，同时也从心态、自学能力、解决未知问题能力等内在的综合实力有了全新的提升。具体来说，学习了Qt的编程框架，包括元对象机制、基本控件的掌握、绘图框架、模型编程、文件操作等相关知识；学习到了Github开源项目托管平台的使用，将其运用到了实际的项目；学习到了openstreetmap等开源地图数据的获取和提炼等知识，自身技术能力又上了一个台阶。在软实力上，打开了用户界面编程的大门，激发自身对编程的热情和后面学习的动力；通过自学实现了课程设计，自身的学习能力得到了有效锻炼；开发的过程中，无论是设计还是编程，都会遇到各种各样的问题，通过独立解决这些疑难杂症，在独立解决问题的能力方面也得到了很大提升；最重要的是，树立不惧所有未知知识、解决疑难问题、接受各种挑战的信心。

**个人体会：**

在学习和实现的过程中体会和领悟还是比较的多，这里挑出感触最深的几点吧。第一，做事情一定要找最重要的事情先做，这样做事情才有效率，不拖拉。由于个人比较追求完美，对程序的用户界面和一些细节很是用心，但是这样就好浪费很多时间，效率不高，应该先做好最重要的事情，例如最不可缺少的功能实现，主体实现然后细节修改，而不是本末倒置。第二，学到的东西不一定都用的到，所以开发还是有一些功利心较好，用到什么然后去学习相应的知识并用到实处，这样才能学以致用，恰到好处。在一开始的学习中，每天看书看教程，发现过去了几天还是什么都不会，然后边学边做，才领悟到学习方法和要点。

**第四部分 参考文献**

1. 霍亚飞.Qt Creator快速入门第二版[M].北京：北京航空航天大学出版社,2014-1
2. 霍亚飞，程梁. QT5 编程入门[M]. 北京：北京航空航天大学出版社,2015-1
3. 严蔚敏，吴伟民.数据结构（C语言版）[M].北京：清华大学出版社,2007
4. 大塚弘记.GitHub入门与实践[M].北京：人民邮电出版社,2015-7