

华中科技大学计算机科学与技术学院

# C 语言课程设计报告

题目：菜农种植信息管理系统

专 业：计算机科学与技术

班 级：计算机（校交）1601 班

学 号：U201611091

姓 名：黄涛

成 绩：

指导教师：甘早斌

完成日期：2017 年 9 月 12 日



# 目录

目录 .....	2
一、 系统需求分析.....	6
1. 基本信息的录入、修改和删除功能.....	6
2. 基本信息的查询功能.....	6
3. 数据统计功能.....	7
4. 数据存储功能.....	7
二、 总体设计 .....	8
1. 文件读写.....	8
2. 数据维护.....	9
3. 数据查询.....	9
4. 数据统计.....	11
三、 数据结构设计.....	12
1. 蔬菜种类信息数据结构.....	12
a) 数据文件存储结构 .....	13
b) 内存中存储结构 .....	13
2. 蔬菜基本信息数据结构.....	13
a) 数据文件存储结构 .....	13
b) 内存中存储结构 .....	14
3. 蔬菜种植信息数据结构.....	14
a) 数据文件存储结构 .....	14
b) 内存中存储结构 .....	15
4. 蔬菜基本信息搜索结果数据结构.....	15
四、 详细设计 .....	16
1. 开发工具选择.....	16
2. 人机交互界面设计.....	16
a) 菜农信息管理系统主窗口.....	16
b) 菜农信息管理系统主窗口菜单栏.....	18
c) 保存数据到备份文件窗口.....	21
d) 从备份文件读取蔬菜数据窗口.....	22
e) 新增蔬菜种植信息窗口.....	22
f) 修改蔬菜种植信息窗口.....	23
g) 删除蔬菜种植信息窗口.....	24
h) 输入年份窗口.....	24
i) 输入范围窗口.....	25
j) 单项搜索窗口.....	25
k) 双项搜索窗口.....	26
l) 信息显示窗口.....	26
3. 数据维护.....	27
a) 信息读取 .....	27
b) 信息保存 .....	31
c) 信息增加.....	32



d ) 信息修改.....	32
e ) 信息删除.....	32
f ) 信息搜索.....	32
g ) 信息排序.....	32
h ) 数据统计.....	33
i ) 链表内存释放.....	33
<b>五、 系统实现 .....</b>	<b>34</b>
1. 源程序文件结构.....	34
2. 源程序清单.....	35
<b>六、 运行测试与结果分析 .....</b>	<b>36</b>
1. 数据维护.....	36
a) 数据添加 .....	36
b) 数据修改 .....	37
c) 数据删除 .....	39
2. 数据查询.....	40
a) 查看蔬菜基本信息 .....	40
b) 查看蔬菜种类信息 .....	41
c) 查看今年蔬菜种植信息 .....	41
d) 查看含关键字蔬菜的种植信息.....	42
e) 查看含分类及营养成分的基本信息 .....	43
3. 数据统计.....	43
a) 统计所有蔬菜的种植面积总和 .....	43
b) 统计某种蔬菜已种植数量.....	44
4. 数据排序.....	44
a) 按种植面积排序种植信息.....	44
b) 按营养成分长度排序蔬菜基本信息.....	45
<b>七、 总结 .....</b>	<b>46</b>
<b>八、 参考文献 .....</b>	<b>47</b>
<b>九、 附录 .....</b>	<b>48</b>
I. 源程序清单.....	48
./CHomework.pro.....	48
./Csource/cppincludes.h.....	50
./Csource/ListAll.c.....	50
./Csource/ListAll.h.....	52
./Csource/MyStringUtil.c.....	53
./Csource/MyStringUtil.h.....	54
./Csource/PlantInfoListSort.c .....	55
./Csource/ PlantInfoListSort.h.....	57
./Csource/PlantingInfo.c.....	58
./Csource/PlantingInfo.h.....	63
./Csource/TypeInfo.c.....	66
./Csource/TypeInfo.h.....	68
./Csource/VegDataBackupUtil.c .....	69
./Csource/VegDataBackupUtil.h.....	71



./Csource/VegInfo.c.....	71
./Csource/VegInfo.h.....	76
./Csource/VegInfoListSort.c.....	79
./Csource/VegInfoListSort.h.....	81
./CppSource/add_plantinfo_dialog.cpp.....	81
./CppSource/ add_plantinfo_dialog.h.....	82
./CppSource/add_type_dialog.cpp.....	83
./CppSource/add_type_dialog.h.....	83
./CppSource/add_vegtype_dialog.cpp.....	84
./CppSource/add_vegtype_dialog.h.....	84
./CppSource/delete_plantinfo_dialog.cpp.....	85
./CppSource/delete_plantinfo_dialog.h.....	86
./CppSource/delete_type_dialog.cpp.....	86
./CppSource/delete_type_dialog.h.....	87
./CppSource/delete_veginfo_dialog.cpp.....	88
./CppSource/delete_veginfo_dialog.h.....	88
./CppSource/edit_plantinfo_dialog.cpp.....	89
./CppSource/edit_plantinfo_dialog.h.....	90
./CppSource/edit_type_dialog.cpp.....	90
./CppSource/edit_type_dialog.h.....	91
./CppSource/edit_veginfo_dialog.cpp.....	92
./CppSource/edit_veginfo_dialog.h.....	93
./CppSource/imformation_dialog.cpp.....	93
./CppSource/imformation_dialog.h.....	93
./CppSource/main.cpp.....	94
./CppSource/mainwindow.cpp.....	94
./CppSource/mainwindow.h.....	117
./CppSource/notfound_dialog.cpp.....	123
./CppSource/notfound_dialog.h.....	123
./CppSource/range_input_dialog.cpp.....	124
./CppSource/range_input_dialog.h.....	124
./CppSource/search2_input_dialog.cpp.....	125
./CppSource/search2_input_dialog.h.....	126
./CppSource/searchinput_dialog.cpp.....	126
./CppSource/searchinput_dialog.h.....	127
./CppSource/vegcode_input_dialog.cpp.....	127
./CppSource/vegcode_input_dialog.h.....	128
./CppSource/yearinput_dialog.cpp.....	128
./CppSource/yearinput_dialog.h.....	129
./UI/add_plantinfo_dialog.ui.....	130
./UI/add_type_dialog.ui.....	132
./UI/add_vegtype_dialog.ui.....	134
./UI/delete_plantinfo_dialog.ui.....	136
./UI/delete_type_dialog.ui.....	139



./UI/delete_veginfo_dialog.ui .....	140
./UI/edit_plantinfo_dialog.ui .....	143
./UI/edit_type_dialog.ui .....	145
./UI/edit_veginfo_dialog.ui .....	147
./UI/imformation_dialog.ui .....	150
./UI/mainwindow.ui .....	151
./UI/notfound_dialog.ui .....	163
./UI/range_input_dialog.ui .....	165
./UI/search2_input_dialog.ui .....	167
./UI/searchinput_dialog.ui .....	169
./UI/vegcode_input_dialog.ui .....	171
./UI/yearinput_dialog.ui .....	173
./database/typeinfo.info .....	174
./database/veginfo.info .....	174
./database/plantinfo.info .....	175
./database/backupFile.bak .....	176
II. 项目源码上传地址 .....	178



# 一、系统需求分析

菜农种植信息管理系统用于管理菜农种植的蔬菜相关信息，主要包括蔬菜种类信息、蔬菜基本信息、菜农种植信息，方便菜农对蔬菜种植信息进行维护、查询、统计，提高工作效率。

菜农种植信息管理系统要求实现一下几个方面的基本功能：

## 1. 基本信息的录入、修改和删除功能

菜农种植信息管理系统的基本信息主要包括以下三类：

- (1) 蔬菜种类信息：蔬菜种类编号、蔬菜种类名称等数据项。
- (2) 蔬菜基本信息：蔬菜编号、蔬菜名称、蔬菜种类、蔬菜营养成分等数据项。
- (3) 菜农种植信息：种植编号、种植蔬菜编号、种植年份、种植面积、收获重量等数据项。

系统应能实现以上三种基础数据信息的录入、修改和删除功能。在信息录入时，应简明的标识出所有用户应当提供的数据项，尽量一次输入就能够完成对象的添加；同时，在添加对象的时候应该还提供信息校验功能，保证添加的数据项合理、正确，能够形成正确的层级关系。

## 2. 基本信息的查询功能

系统应该实现对以上三种基础数据信息的查询功能，提供按多种条件分别进行查询的方式。具体包括：

### 1) 蔬菜种类信息查询功能

以分类编码为条件来查找并显示满足条件的蔬菜分类信息。例如，查找并显示分类编码为'3'的蔬菜分类信息。

### 2) 蔬菜基本信息查询功能

①蔬菜名称中文字符子串为条件查找，并显示蔬菜中包含指定子串的蔬菜基本信息。例如，查找并显示果菜类蔬菜名称中包含为“椒”的果菜类蔬菜基本信息。



②以分类码和营养成分为条件查找并显示满足条件的蔬菜基本信息。例如, 查找并显示分类码为'4'(叶菜类)且营养成分中含有铁的所有蔬菜基本信息。

### 3)蔬菜种植信息查询功能

①蔬菜部分名称(模糊查找)和种植年份为条件查找并显示满足条件的所有蔬菜种植信息。例如, 查找并显示蔬菜名称中含有椒的蔬菜且在“2015”年种植的所有蔬菜信息。

②以蔬菜名称为条件查找并显示满足条件的蔬菜种植信息。例如, 查找并显示蔬菜名称为“菠菜”的所有年份种植的菠菜信息(重量), 并按年份分别显示。

上述查询结果中, 如果有多条信息被查中, 则需提供逐条显示再进行选择的功能。

## 3. 数据统计功能

在以上三种基础数据信息的基础上, 提供多方面的数据统计功能, 具体包括:

①分别统计某年各类蔬菜(如叶菜类、根茎类等)种植总面积、收获总重量, 按总重量降序排序后, 输出分类名称、种植面积、收获总重量(按种类统计)。

②以所输入的起止年份为条件(如 2015-2017), 按蔬菜名称(如菠菜、黄瓜等)统计该三年内所种各种蔬菜总面积、收获总重量, 按总重量降序排序后, 输出蔬菜名称、分类名称、种植面积、总重量。

③分别统计某种类蔬菜的已有的数量。

④给定某个营养成分, 程序自动判定含有该营养成分的蔬菜, 并且显示输出所有含有该营养成分的蔬菜名称。

⑤有关该菜农蔬菜种植信息的其他方面数据统计。

## 4. 数据存储功能

以上三种信息在程序运行时, 以可以动态扩展的链表的形式存储在内存中; 同时, 在硬盘上以数据文件的形式实现长期保存。在程序退出时将内存中的数据与硬盘上的数据进行同步, 从而保证数据的一致性以及安全性。



## 二、总体设计

根据系统需求分析结果，将整个系统在宏观上分解成了以下几个功能模块，方便具体实现：



图 2.0 菜农种植信息管理系统功能模块

### 1. 文件读写

文件读写模块包括一系列与系统启动时环境建立与系统结束运行时数据保存的操作。它进一步划分为 4 个子模块：数据加载、数据保存、数据备份、数据恢复。

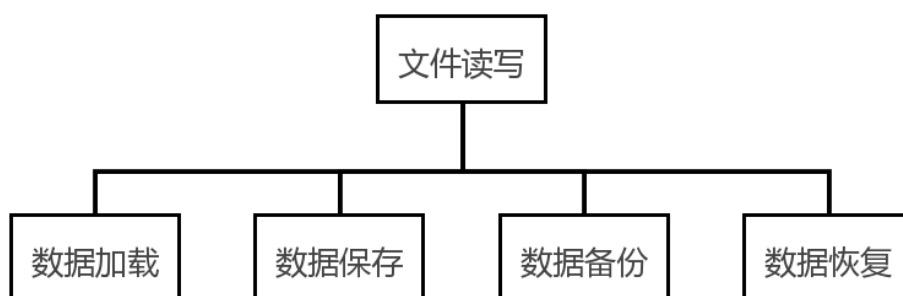


图 2.1 文件读写模块的子模块划分

- 数据加载子模块：用于将分别存储在多个数据文件中的代码数据和基础数据读入内存，构造数据链表，同时输出数据加载相关提示信息。
- 数据保存子模块：用于将内存中的链表数据按缺省路径分别保存到各个数据文件；当数据保存失败时，输出提示信息并让用户输入存储数据文件的新路径，然后按用户指定的新路径保存数据。





- **数据备份子模块：**用于将存放在多个数据文件中的代码数据和基础数据按用户指定路径转储到一个数据文件之中，供用户在系统数据被破坏或丢失后用来恢复系统。
- **数据恢复子模块：**与数据备份子模块在功能上相对应，能从一个备份数据文件恢复得到备份时间点的数据，并将恢复出来的数据加载到内存中。数据恢复子模块与数据备份子模块一起，用于提高系统的安全性和可靠性。

## 2. 数据维护

数据维护模块负责对三种基础数据信息的录入、修改和删除功能，需要保证数据的准确性、完整性和有效性。该模块按信息种类进一步划分为蔬菜种类信息维护、蔬菜基本信息维护、蔬菜种植信息维护三个子模块。

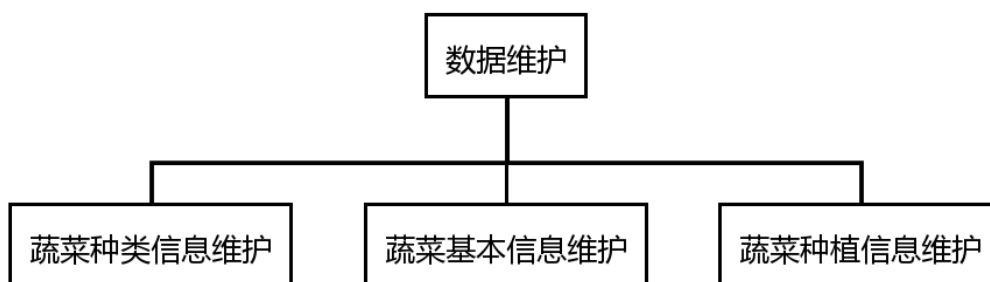


图 2.2 数据维护模块的子模块划分

- **蔬菜种类信息维护、蔬菜基本信息维护和蔬菜种植信息维护子模块：**三个模块分别用于蔬菜种类信息、蔬菜基本信息、蔬菜种植信息的录入、修改、删除等操作。要确保基础数据的正确性，确保用户在操作以上三种基础信息的时候不会破坏数据的层级结构。

## 3. 数据查询

数据查询模块提供对三种基础数据按多种条件进行查询的功能。该模块按信息种类分为蔬菜种类信息查询、蔬菜基本信息查询和蔬菜种植信息查询。

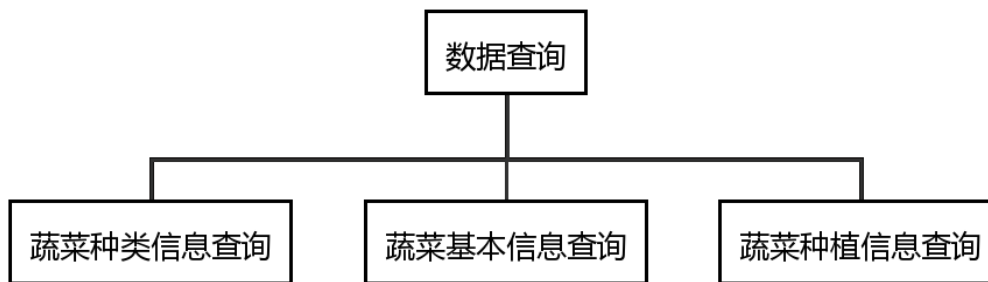


图 2.3 数据查询模块的子模块划分

- 蔬菜种类信息查询子模块：用于提供以提供分类编码方式查询蔬菜种类信息。

- 蔬菜基本信息查询子模块：用于提供五种方式查询蔬菜基本信息。

方式一：以蔬菜种类方式查看蔬菜基本信息，将展示该种类包含的所有蔬菜信息。

方式二：以提供蔬菜名关键字方式查看蔬菜基本信息，将展示蔬菜名包含关键字的所有蔬菜基本信息。

方式三：以提供一项蔬菜营养成分的方式查看蔬菜基本信息，将展示含有提供营养成分的所有蔬菜的基本信息。

方式四：以提供营养成分以及蔬菜种类的方式查看蔬菜基本信息，将会筛选出满足含有提供的营养成分以及蔬菜种类相符的蔬菜信息用于展示。

方式五：以多种排序方式对蔬菜基本信息进行展示，排序方式包括按编号排序、按蔬菜种类排序、按营养成分长度排序。

- 蔬菜种植信息查询子模块：用于提供八种方式查询蔬菜种植信息。

方式一：以蔬菜种植年份为条件查看蔬菜种植信息，将展示指定年份的所有蔬菜种植信息。

方式二：以蔬菜种类为条件查看蔬菜种植信息，将展示该种类下所有蔬菜的种植信息。

方式三：以提供蔬菜名关键字方式查看蔬菜种植信息，将展示蔬菜名包含关键字的蔬菜种植信息。

方式四：以提供一项营养成分的方式查看蔬菜种植信息，将展示包含该营养成分的所有蔬菜的种植信息。



方式五：以提供一项营养成分及蔬菜种类的方式查看蔬菜种植信息，将会筛选出满足包含该营养成分及蔬菜种类相符的蔬菜的种植信息。

方式六：以提供蔬菜名和关键字以及种植年份的方式查看种植信息，将会筛选出满足蔬菜名包含关键字以及种植年份相符的蔬菜种植信息。

方式七：以提供一个范围的方式查看蔬菜种植信息，可提供种植面积范围、收获重量范围、种植年份区间，获取对应数据项符在范围内的蔬菜种植信息。同时，获取到的种植信息将分别按种植面积升序、收获重量升序、收获重量降序的方式排列。

方式八：以多种排序方式对蔬菜种植信息进行展示，排序方式包括按种植编号排序、按蔬菜种类排序、按种植面积排序、按种植年份排序、按收获重量排序。

#### 4. 数据统计

数据统计模块提供对三种基础数据进行多方面统计的功能。按统计条件，该模块划分为 5 个子模块。

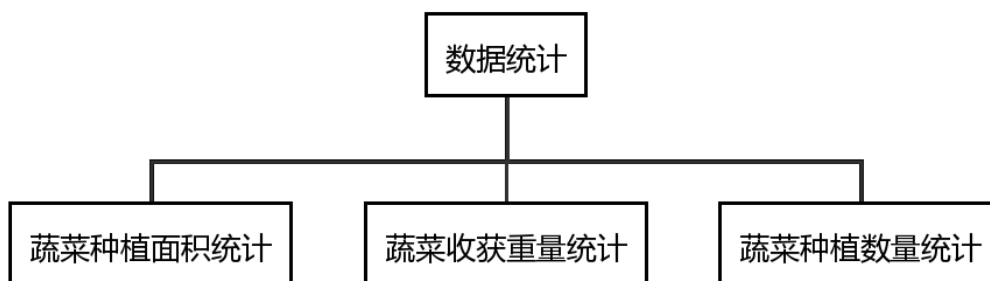


图 2.4 数据统计模块的子模块划分

- 蔬菜种植面积统计子模块：用于统计指定范围的蔬菜种植面积总和，可统计全部蔬菜种植信息、一类蔬菜种植信息、某年蔬菜种植信息。
- 蔬菜种植收获重量统计子模块：用于统计指定范围的蔬菜收获重量总和，可统计全部蔬菜种植信息、一类蔬菜种植信息、某年蔬菜种植信息。
- 蔬菜种植数量统计子模块：用于某种蔬菜种植信息的数量。



### 三、数据结构设计

按照任务要求，系统需要处理的基础信息有三种：蔬菜种类信息、蔬菜基本信息和蔬菜种植信息。这三种信息之间存在这样的关联：蔬菜种类下有对应种类的蔬菜、对应种类的蔬菜具有对应的种植信息，且在系统中常需要获取到某项种植信息对应的蔬菜信息，某种蔬菜信息对应的蔬菜种类。根据这三种基础信息的关系，可采用三个方向的十字交叉链表作为数据结构的实现，示意图见图 3.0。同时，在进行数据检索等操作时，对三种基础信息采用单向链表更为高效，也为每种信息单独设计了对应的链表。

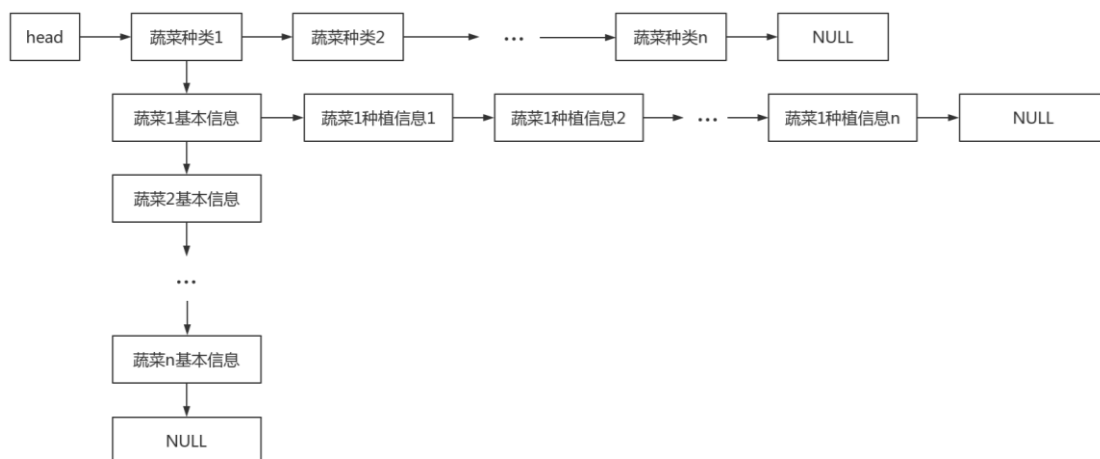


图 3.0 菜农种植信息管理系统三个方向十字交叉链表示意图

下面分别设计本系统所涉及的代码数据、基础数据和生成数据的数据结构，以及数据在内存和外存中的存储结构。

#### 1. 蔬菜种类信息数据结构

蔬菜种类信息代码表如表 3.1 所示。

表 3.1 蔬菜种类信息代码表

数据结构名称：蔬菜种类信息			数据结构标识：typeInfo		
数据项名称	数据项标识	数据类型	数据长度	取值范围	示例
种类编号	typeCode	char	1	‘0’ - ‘9’	‘1’
种类名称	typeName	string	10		“瓜类”



### a) 数据文件存储结构

蔬菜种类信息数据以文本的形式存储在“typeinfo.ini”文件中，通过读取时对文本数据进行进一步处理得到蔬菜种类结构体信息。

一项蔬菜种类信息的文本存储示例如下：

```
{'1','根茎类'}
```

其中，“{}”包含的为一项种类信息，“”“”包含的为一项数据项。按顺序分别为种类编号、种类名称。

### b) 内存中存储结构

将内存中的蔬菜种类信息结构体连接成链表。

表 3.2 蔬菜种类信息结构体

数据结构名称：蔬菜种类信息		数据结构标识：typeInfo
数据项名称	数据项标识	数据类型
种类编号	typeCode	char
种类名称	typeName	char[10]
下一种类信息节点	next	typeInfo *
对应蔬菜基本信息链表	vInfoList	vegInfo

## 2. 蔬菜基本信息数据结构

蔬菜基本信息代码表如表 3.3 所示。

表 3.3 蔬菜基本信息代码表

数据结构名称：蔬菜基本信息			数据结构标识：vegInfo		
数据项名称	数据项标识	数据类型	数据长度	取值范围	示例
蔬菜编号	vegCode	int	4	0 -	‘1’
蔬菜名称	vegName	string	20		“西瓜”
种类编号	vegType	char	1	‘0’ - ‘9’	‘1’
营养成分	facts	string	200		“维生素 C”

### a) 数据文件存储结构

蔬菜基本信息数据以文本的形式存储在“veginfo.ini”文件中，通过读取时对文本数据进行进一步处理得到蔬菜信息结构体。

一项蔬菜基本信息的文本存储示例如下：



```
{'1','白萝卜','1','植物蛋白、维生素 C 和叶酸等'}
```

其中，“{}”包含的为一项种类信息，“”“包含的为一项数据项。按顺序分别为蔬菜编号、蔬菜名称、种类编号、营养成分。

### b) 内存中存储结构

将内存中的蔬菜基本信息结构体连接成链表。

表 3.4 蔬菜基本信息结构体

数据结构名称：蔬菜种类信息		数据结构标识：vegInfo
数据项名称	数据项标识	数据类型
蔬菜编号	vegCode	int
蔬菜名称	vegName	char[20]
种类编号	vegType	char
营养成分	facts	char[200]
前一蔬菜信息节点	prev	vegInfo *
后一蔬菜信息节点	next	vegInfo *
对应种植信息链表	plnfoList	plantInfo *
对应种类信息结构体	father	typeInfo *

## 3. 蔬菜种植信息数据结构

蔬菜种植信息代码表如表 3.5 所示。

表 3.5 蔬菜种植信息代码表

数据结构名称：蔬菜种植信息		数据结构标识：plantInfo			
数据项名称	数据项标识	数据类型	数据长度	取值范围	示例
蔬菜编号	vegCode	int	4	0 -	'1'
种植编号	id	int	4	0 -	'1'
种植面积	area	int	4	0 -	'1'
收获重量	weight	float	4		“维生素 C”
种植年份	year	string	5		“2017”

### a) 数据文件存储结构

蔬菜种植信息数据以文本的形式存储在“plantinfo.ini”文件中，通过读取时对文本数据进行进一步处理得到蔬菜信息结构体。

一项蔬菜种植信息的文本存储示例如下：

```
{'1','1','6','23.330','2016'}
```



其中，“{}”包含的为一项种类信息，“” “包含的为一项数据项。按顺序分别为种植编号、蔬菜编号、种植面积、收获重量、种植年份。

#### b) 内存中存储结构

将内存中的蔬菜种植信息结构体连接成链表。

表 3.6 蔬菜种植信息结构体

数据结构名称：蔬菜种类信息		数据结构标识：vegInfo
数据项名称	数据项标识	数据类型
蔬菜编号	vegCode	int
种植编号	id	int
种植面积	area	int
收获重量	weight	float
种植年份	year	char[5]
前一种植信息节点	prev	plantInfo *
后一种植信息节点	next	plantInfo *
对应蔬菜信息结构体	father	typeInfo *

## 4. 蔬菜基本信息搜索结果数据结构

蔬菜基本信息搜索结果结构体用于储存搜索到的符合搜索条件的蔬菜基本信息对应的蔬菜编号数组及搜索到的结果数，便于 UI 层对搜索结果进行处理。

表 3.7 蔬菜基本信息搜索结果结构体

数据结构名称：蔬菜基本信息搜索结		数据结构标识：searchResult
数据项名称	数据项标识	数据类型
搜索结果数	resultNum	int
搜索到的蔬菜编号数组	vegCodeArray	int *



## 四、详细设计

本部分详细介绍系统的开发工具选择、界面设计、程序流程、数据读取、菜单结构、具体功能实现等。

### 1. 开发工具选择

本系统在在开发设计上需要实现的有：美观的界面、界面与核心代码良好地分离、开发难度低、跨平台移植难度低。基于上述开发设计，确定了使用 Qt 图形库作为界面层、C 语言作为核心代码层的双层结构。Qt 是一个跨平台 C++图形用户界面应用程序开发框架，可采用 C++语言进行界面的编写，同时，C++对 C 有着良好的兼容性，可以完美地与系统核心层进行通讯，降低开发适配难度。

系统的开发选用了 Qt Creator 这一集成开发环境，使用 Qt Creator 能够高效地开发 Qt 应用程序。

### 2. 人机交互界面设计

本系统采用了基于 Qt 实现的多窗口交互界面，通过多窗口的设计，能够更好的反应程序的层级关系。下面将分别对交互界面各窗口进行详细介绍。

#### a) 菜农信息管理系统主窗口

本窗口为程序运行的主窗口，程序启动时首先显示的是主窗口，几乎所有的功能都由该窗口实现并呈现，主窗口的运行流程图如图 4.2.0 所示。



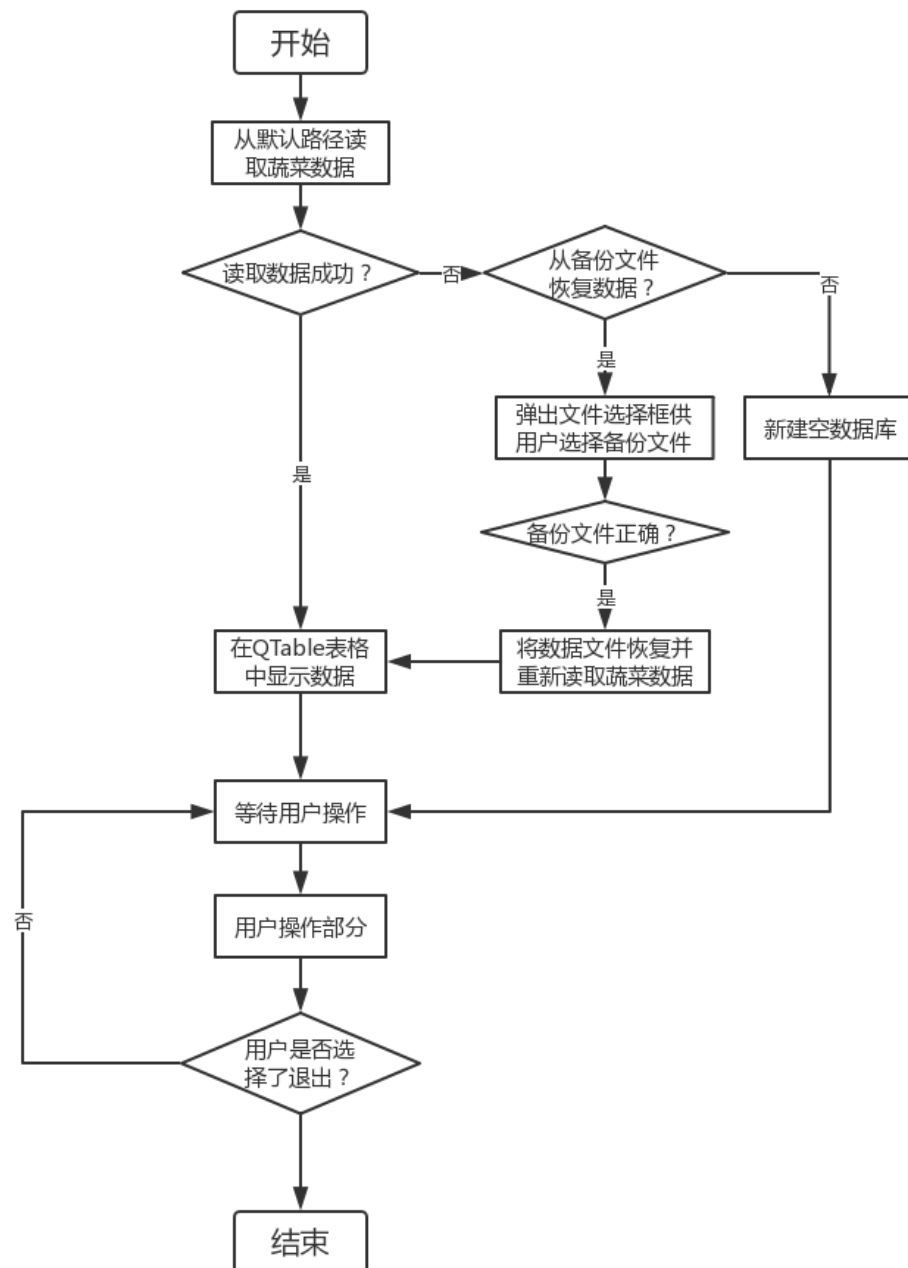


图 4.2.0 主程序运行流程图

如图 4.2.1 所示，窗口的控件结构自上而下主要包括显示程序菜单的菜单栏(menuBar)、显示程序信息的文本控件(label)、显示蔬菜种类信息等信息的表格控件(QTable)、显示程序状态信息的状态栏(statusBar)。



图 4.2.1 软件主窗口运行截图

#### b) 菜农信息管理系统主窗口菜单栏

如图 4.2.2 所示为主窗口菜单栏界面截图。主窗口下做的一切操作均通过选择菜单栏下的菜单项进行。

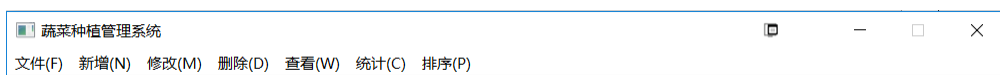


图 4.2.2 软件主窗口菜单栏截图

菜农信息管理系统用户操作分为七大类，分别为：文件、新增、修改、删除、查看、统计、排序。其中每个大类都有其对应的二级菜单或动作。具体菜单结构图如图 4.2.3 所示。

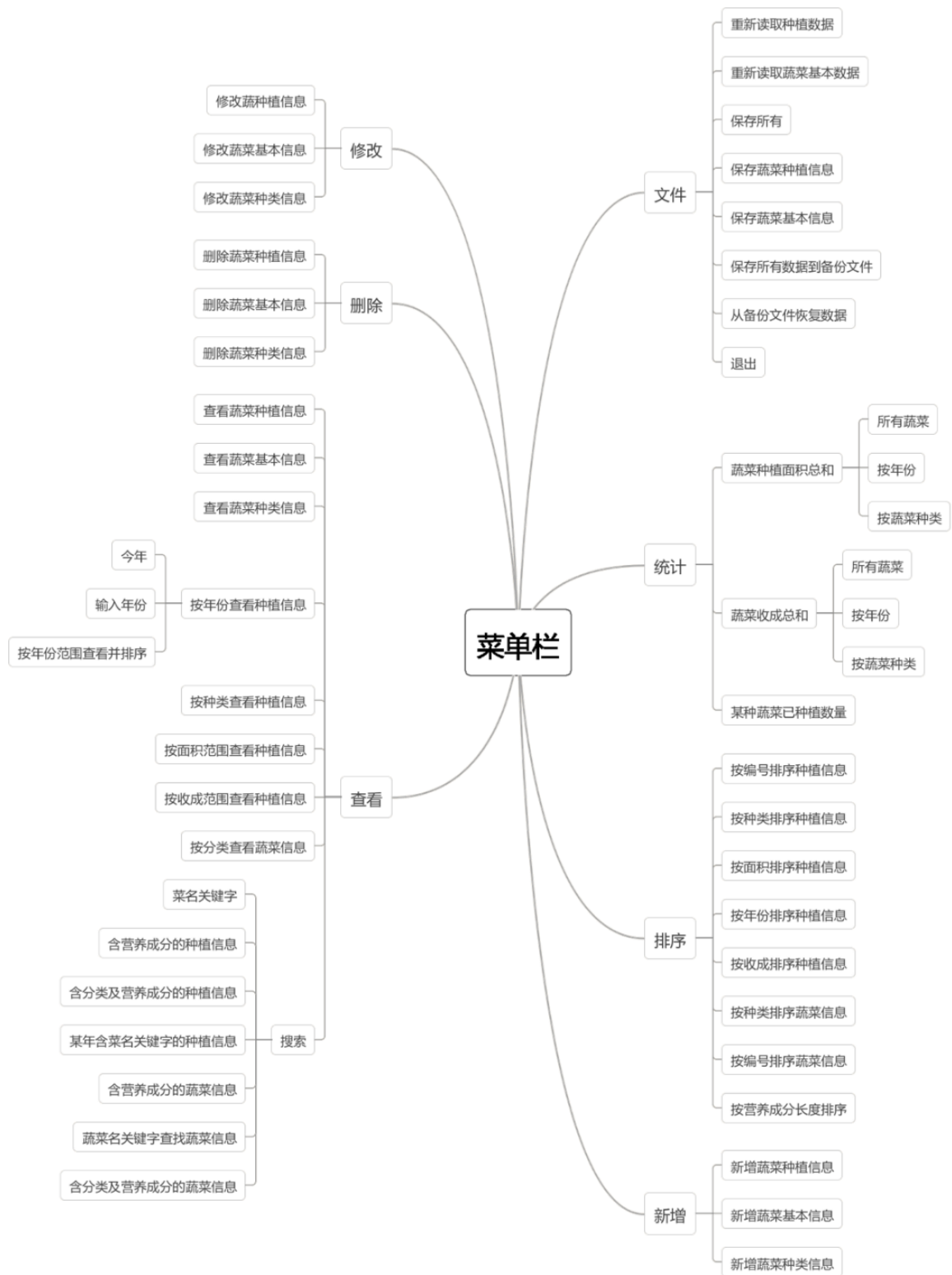


图 4.2.3 菜单结构图



各菜单项对应的具体操作如下表。

菜单项名称	菜单项功能
重新读取种植数据	从文件中重新读取蔬菜种植信息
重新读取蔬菜基本数据	从文件中重新读取蔬菜基本信息
保存所有	保存内存中的所有数据到文件中
保存蔬菜种植信息	保存蔬菜种植信息数据到文件中
保存蔬菜基本信息	保存蔬菜基本信息数据到文件中
保存所有数据到备份文件	保存所有数据到一个备份文件中
从备份文件回复数据	从备份文件中读取数据存储为数据文件
退出	退出程序

表 4.2.4 文件菜单功能对照表

菜单项名称	菜单项功能
新增蔬菜种植信息	新增一项蔬菜种植信息至链表尾部并存储到文件中
新增蔬菜基本信息	新增一项蔬菜基本信息至链表尾部并存储到文件中
新增蔬菜种类信息	新增一项蔬菜种类信息至链表尾部并存储到文件中

表 4.2.5 新增菜单功能对照表

菜单项名称	菜单项功能
修改蔬菜种植信息	修改一项蔬菜种植信息并存储到文件中
修改蔬菜基本信息	修改一项蔬菜基本信息并存储到文件中
修改蔬菜种类信息	修改一项蔬菜种类信息并存储到文件中

表 4.2.6 修改菜单功能对照表

菜单项名称	菜单项功能
删除蔬菜种植信息	删除一项蔬菜种植信息并存储到文件中
删除蔬菜基本信息	删除一项蔬菜基本信息并存储到文件中
删除蔬菜种类信息	删除一项蔬菜种类信息并存储到文件中

表 4.2.7 修改菜单功能对照表

菜单项名称	菜单项功能
查看蔬菜种植信息	表格控件显示当前蔬菜种植信息
查看蔬菜基本信息	表格控件显示当前蔬菜基本信息
查看蔬菜种类信息	表格控件显示当前蔬菜种类信息
查看今年种植信息	表格控件显示今年的蔬菜种植信息
查看指定年份种植信息	弹出输入年份窗口
按年份范围查看种植信息并排序	弹出输入范围窗口
按种类查看种植信息	表格控件显示指定种类的种植信息
按面积范围查看种植信息	弹出输入范围窗口
按收成范围查看种植信息	弹出输入范围窗口
按种类查看蔬菜基本信息	表格控件显示指定种类的蔬菜基本信息
搜索菜名关键字	弹出单项搜索窗口
搜索含营养成分的种植信息	弹出单项搜索窗口
搜索含分类及营养成分的种植信息	弹出双项搜索窗口
搜索某年含菜名关键字的种植信息	弹出双项搜索窗口
搜索含营养成分的蔬菜信息	弹出单项搜索窗口
含分类及营养成分的蔬菜信息	弹出双项搜索窗口

表 4.2.8 查看菜单功能对照表



菜单项名称	菜单项功能
统计所有蔬菜种植面积	统计所有蔬菜种植面积并弹出窗口显示
统计一年蔬菜种植面积	统计一年蔬菜种植面积并弹出窗口显示
统计一类蔬菜种植信息	统计一类蔬菜种植面积并弹出窗口显示
统计所有蔬菜收获重量	统计所有蔬菜收获重量并弹出窗口显示
统计一年蔬菜收获重量	统计一年蔬菜收获重量并弹出窗口显示
统计一类蔬菜收获重量	统计一类蔬菜收获重量并弹出窗口显示
统计某种蔬菜种植记录数	统计一种蔬菜的种植记录数并弹出窗口显示

表 4.2.8 统计菜单功能对照表

菜单项名称	菜单项功能
按编号排序种植信息	按编号升序排列蔬菜种植信息并在表格控件上显示
按种类排序种植信息	按种类升序排列蔬菜种植信息并在表格控件上显示
按面积排序种植信息	按面积升序或降序排列蔬菜种植信息并在表格控件上显示
按年份排序种植信息	按年份升序或降序排列蔬菜种植信息并在表格控件上显示
按收成排序种植信息	按收成升序或降序排列蔬菜种植信息并在表格控件上显示
按种类排序蔬菜信息	按种类升序排列蔬菜基本信息并在表格控件上显示
按编号排序蔬菜信息	按编号升序排列蔬菜基本信息并在表格控件上显示
按营养成分长度排序	按营养成分长度升序排列蔬菜基本信息并在表格控件上显示

表 4.2.8 排序菜单功能对照表

### c) 保存数据到备份文件窗口

用户在选择菜单项【文件->保存所有数据到备份文件】后会弹出文件保存对话框供用户选择目录和文件名，点击保存后，程序会获取到要保存到的目的文件的绝对路径，并将分别存储于三个文件中的蔬菜种类信息、蔬菜基本信息、蔬菜种植信息存储于备份文件中。

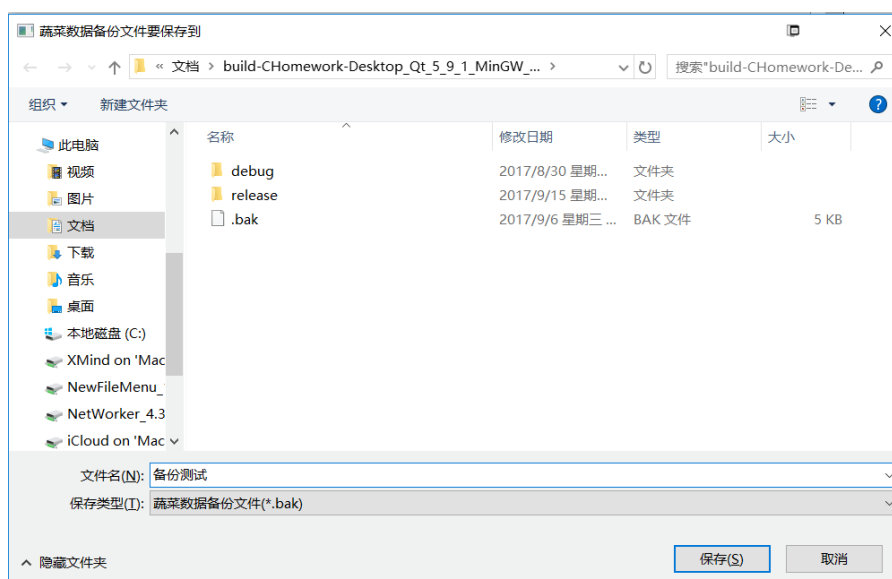


图 4.2.9 保存数据到备份文件窗口截图



#### d) 从备份文件读取蔬菜数据窗口

用户在选择菜单项【文件->从备份文件读取所有蔬菜数据】后会弹出打开文件对话框供用户选择目录和文件，点击打开后，程序会获取到要打开的文件的绝对路径，并读取文件内容，若内容正确，将会将蔬菜数据恢复到默认数据文件中，若内容错误将会弹出读取错误对话框。

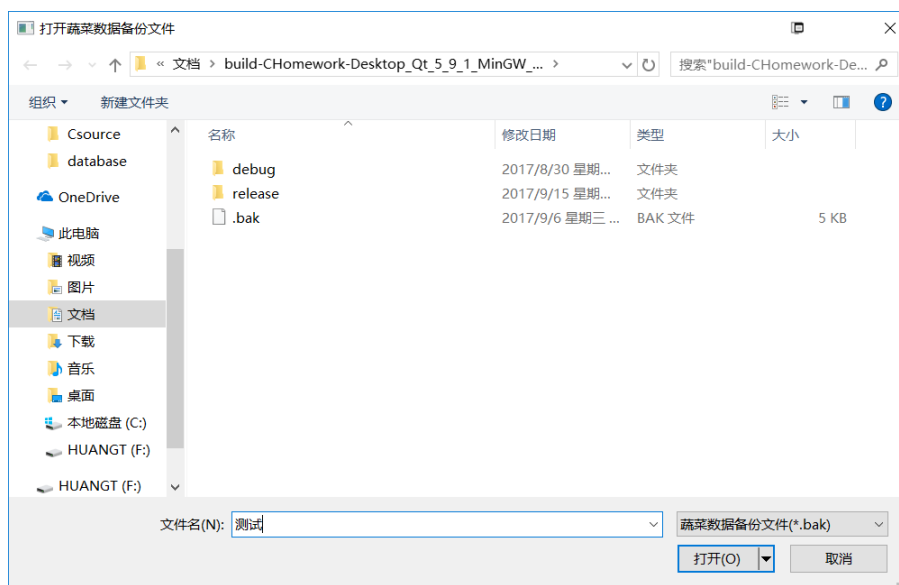


图 4.2.10 从备份文件读取蔬菜数据窗口截图

#### e) 新增蔬菜种植信息窗口

用户在选择菜单项【新增->蔬菜种植信息】后会弹出新增蔬菜种植信息窗口，窗口内含有输入框控件供用户输入新增蔬菜种植信息需要的数据，点击【OK】后程序会先对用户输入的信息进行判断，若用户输入的蔬菜在数据库中不存在将会弹出“未找到该蔬菜”提示窗口，若输入项均正确，程序将会新建一项蔬菜种植信息连接至链表尾部，同时保存至文件并刷新主窗口表格显示内容。

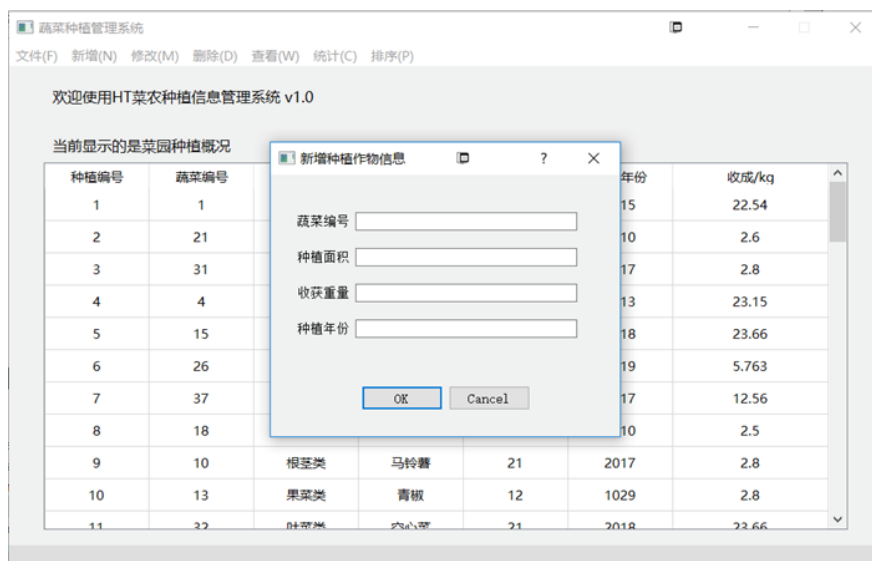


图 4.2.11 新增蔬菜种植信息窗口截图

#### f) 修改蔬菜种植信息窗口

用户在选择菜单项【修改→蔬菜种植信息】后会弹出修改蔬菜种植信息窗口，窗口内含有输入框控件供用户输入修改蔬菜种植信息需要的数据，点击【OK】后程序会先对用户输入的信息进行判断，若输入项均正确，程序会在链表中遍历查找符合条件的蔬菜种植信息，若数据库中无法找到符合条件的种植信息将会弹出“未找到该蔬菜”提示窗口，若找到种植信息将会根据用户输入修改信息内容，输入为空的项不作修改，同时保存至文件并刷新主窗口表格显示内容。

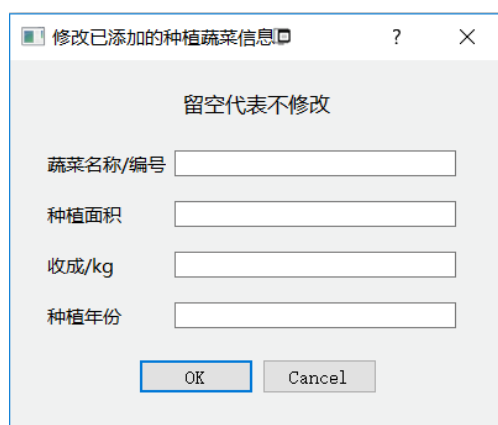


图 4.2.12 修改蔬菜种植信息窗口截图



### g) 删除蔬菜种植信息窗口

用户在选择菜单项【删除→蔬菜种植信息】后会弹出删除蔬菜种植信息窗口，窗口内含有输入框控件供用户输入要删除的蔬菜种植编号，点击【OK】后程序会先对用户输入的信息进行判断，若输入项均正确，程序会在链表中遍历查找符合条件的蔬菜种植信息，若数据库中无法找到符合条件的种植信息将会弹出“未找到该蔬菜”提示窗口，若找到种植信息将会删除该条种植信息，同时保存至文件并刷新主窗口表格显示内容。

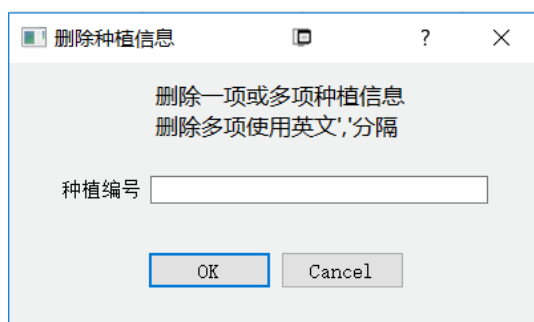


图 4.2.13 删除蔬菜种植信息窗口截图

### h) 输入年份窗口

用户在选择【查看→按年份查看种植信息→输入年份】、【统计→蔬菜种植面积总和→按年份→输入年份】等菜单项后会弹出输入年份窗口，该窗口含有输入框，供用户输入需要的年份，点击【OK】后将会对输入的内容进行判断，若输入的为年份，则将输入的值以信号量(signals)的形式传递到位于主程序的槽(slots)中，主程序的槽函数根据用户输入年份的需求类型做出相应的处理。

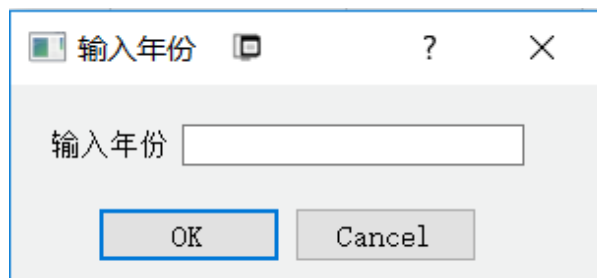


图 4.2.14 输入年份窗口截图





### i) 输入范围窗口

用户在选择【查看->按面积范围查看种植信息】、【查看->按收成范围查看种植信息】等菜单项后会弹出输入范围窗口，该窗口含有输入框，供用户输入起始值和终止值，点击【OK】后将会对输入的内容进行判断，将输入的较大值与较小值存入用于存储范围的类的对象中，以信号量(signals)的形式传递到位于主程序的槽(slots)中，主程序的槽函数根据用户输入范围的需求类型做出相应的处理。

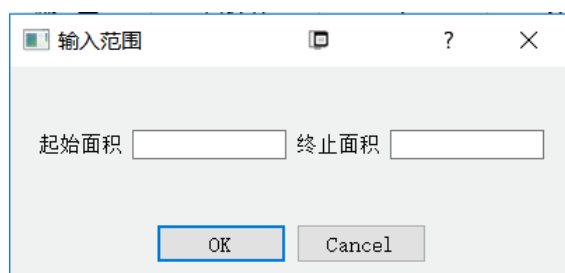


图 4.2.15 输入范围窗口截图

### j) 单项搜索窗口

用户在选择【查看->搜索->菜名关键字】、【查看->搜索->含营养成分的种植信息】等菜单项后会弹出单项搜索窗口，该窗口含有输入框，供用户输入要搜索的内容，点击【OK】后将会对输入的内容进行判断，将输入内容以信号量(signals)的形式传递到位于主程序的槽(slots)中，主程序的槽函数根据用户搜索的需求类型做出相应的处理。

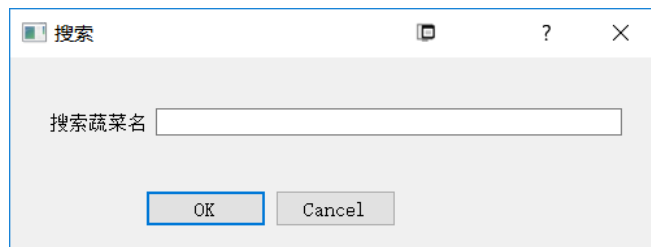


图 4.2.16 单项搜索窗口截图



### k) 双项搜索窗口

该窗口主要满足用户多条件搜索的需求。在选择【查看->搜索->含分类及营养成分的种植信息】、【查看->搜索->某年含菜名关键字的种植信息】等菜单项后会弹出双项搜索窗口，该窗口含有输入框，供用户输入要搜索的内容，点击【OK】后将会对输入的内容进行判断，将输入内容以信号量(signals)的形式传递到位于主程序的槽(slots)中，主程序的槽函数根据用户搜索的需求类型做出相应的处理。

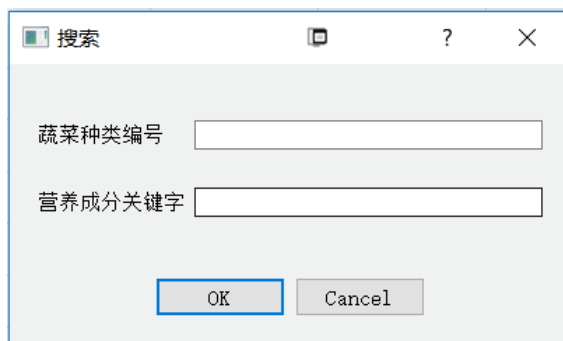


图 4.2.17 双项搜索窗口截图

### l) 信息显示窗口

该窗口主要用于显示某些提示信息。在选择【统计->蔬菜种植面积总和】、【统计->蔬菜收成总和】等菜单项后或有信息需要显示时会弹出双项搜索窗口，该窗口含有文本控件供显示信息。

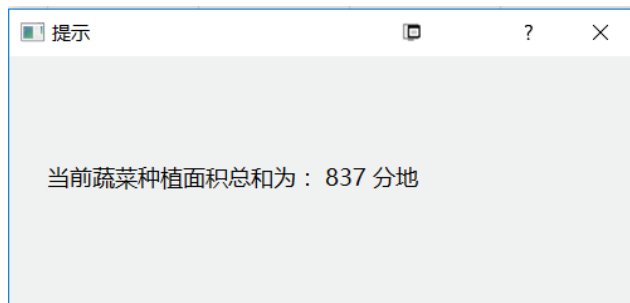


图 4.2.17 信息显示窗口截图



### 3. 数据维护

本节主要介绍了各信息的读取、保存、增加、删除、修改、搜索、排序等操作的具体实现。

#### a) 信息读取

信息读取操作指的是程序从数据文件中读取数据并连接成链表。由于蔬菜种类信息、蔬菜基本信息、蔬菜种植信息的信息读取算法基本相同，选取蔬菜种类信息的读取进行详细介绍。

为了保证代码的可读性、模块化，同时适配十字交叉链表与单项链表均使用的需求，读取数据操作由两部分代码构成。

① 第一部分：从文件中读取一项蔬菜种类信息。该部分函数声明为

```
typeInfo getTypeItemFromFile(FILE * typeInfoFile);
```

其中，函数参数为可写打开的种类数据文件指针，返回值为读取到的一项蔬菜种类信息结构体指针。调用此函数时，函数会从文件指针处向后读取一项蔬菜种类信息并返回，若数据错误或遇到文件尾，返回 NULL。读取一项信息函数流程图如图 3.1.0 所示。

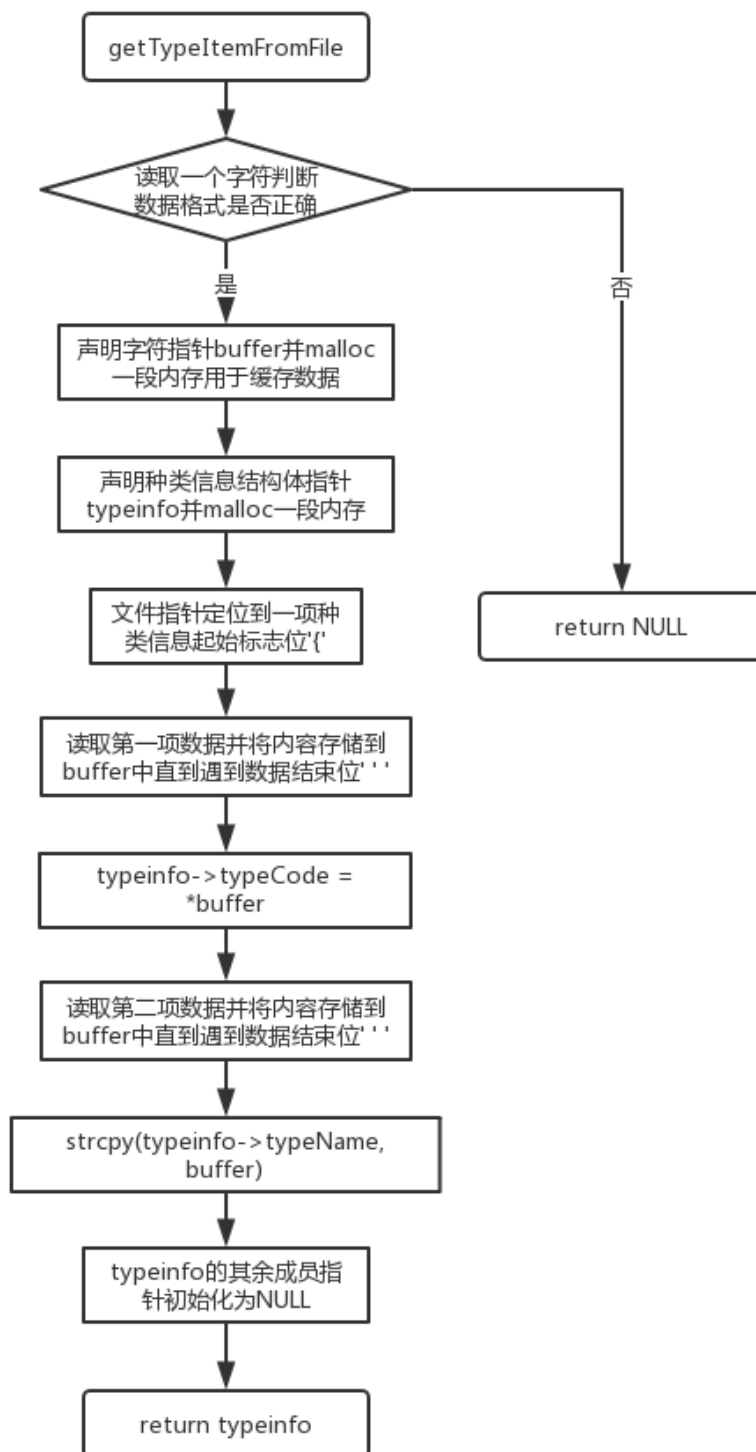


图 3.1.0 读取一项蔬菜种类信息流程图

② 第二部分：将读取到的单项信息文件链接成链表。该部分包含两个函数。  
读取文件并构建单向链表的函数声明如下：



```
typeInfo loadTypeInfoFromFile (void);
```

函数返回值为单向链表的表头指针。若读取失败则返回 NULL。构建单向链表的流程图如图 3.1.1 所示。

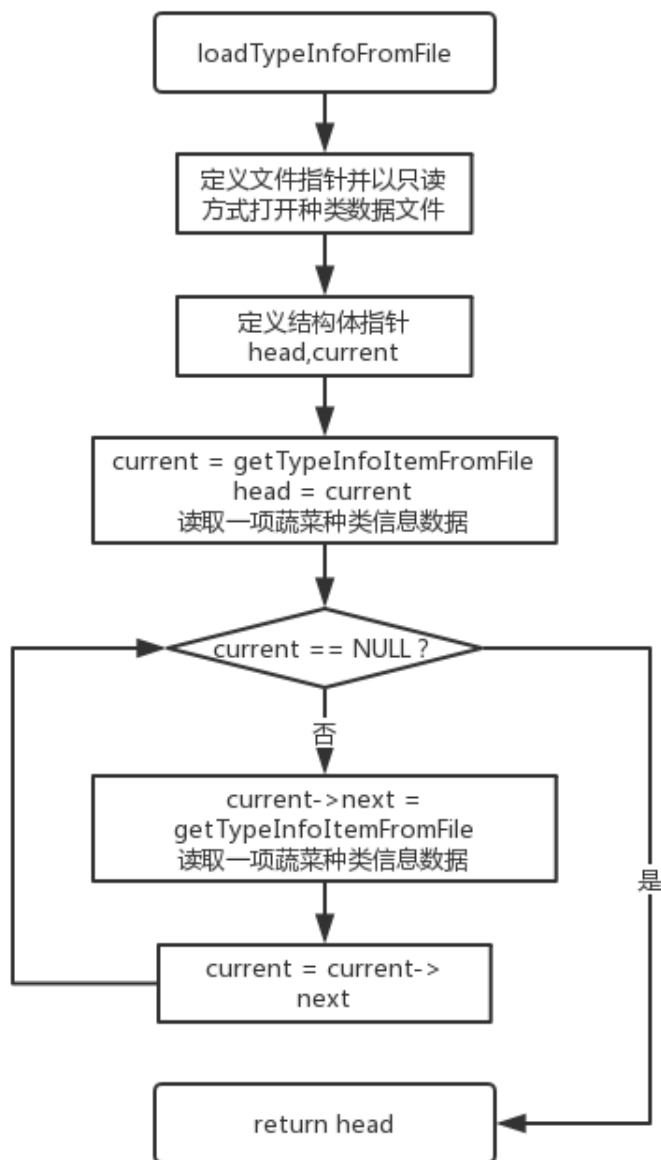


图 3.1.1 构建单向链表流程图

读取文件并构建十字交叉链表的函数声明如下：

```
typeInfo loadFarmListFromFile(void);
```

函数返回值为十字链表的头指针。若读取失败则返回 NULL。构建十字交叉链表流程图如图 3.1.2 所示。

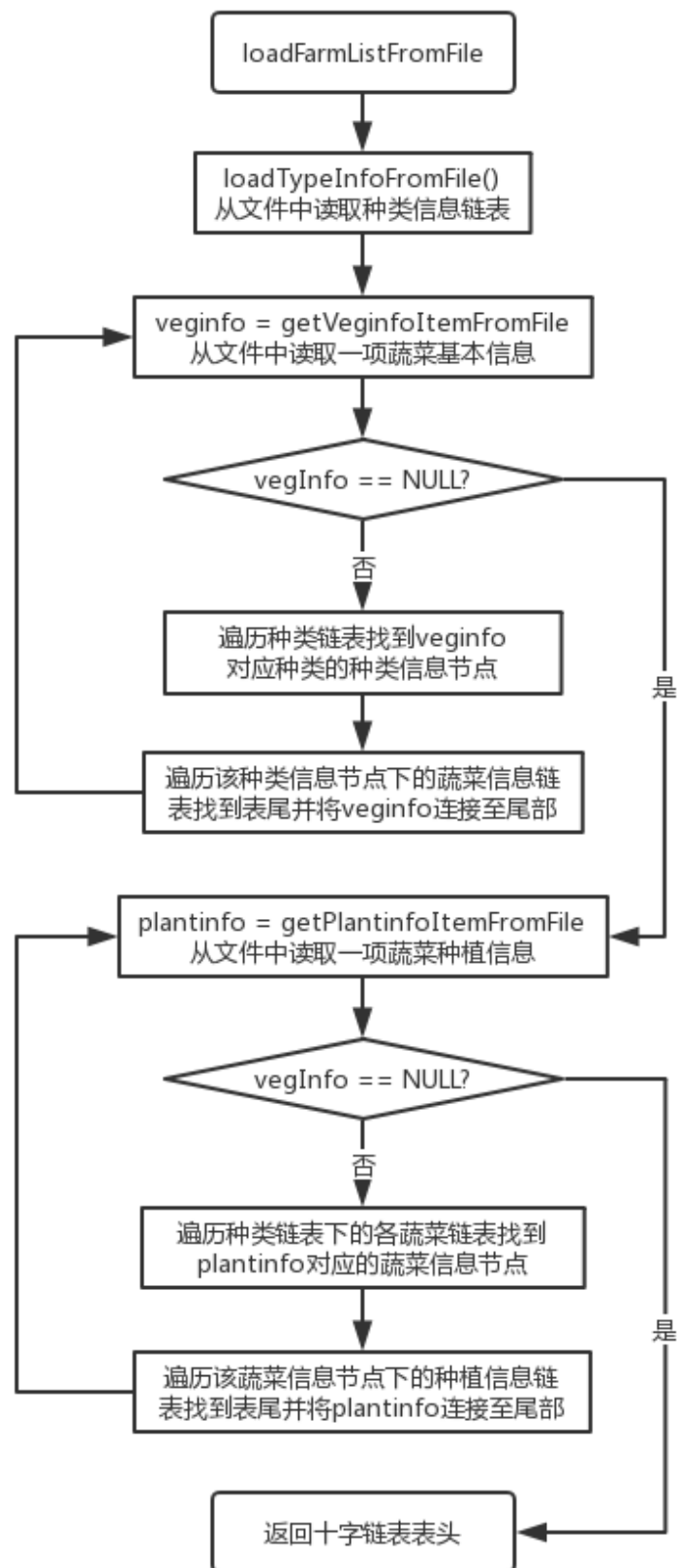


图 3.1.2 构建十字交叉链表流程图



## b) 信息保存

信息保存操作是指将内存中的数据链表存储至数据文件中。由于蔬菜种类信息、蔬菜基本信息、蔬菜种植信息信息保存算法基本相同，本节以蔬菜种类信息保存为例详细介绍。

信息保存需要将当前内存中的数据按照指定的格式存入文本文件中。蔬菜种类信息保存函数声明如下：

```
void saveTypeInfoToFile(typeInfo head);
```

函数参数为蔬菜种类信息链表表头。信息保存操作流程如图 3.2.1 所示。

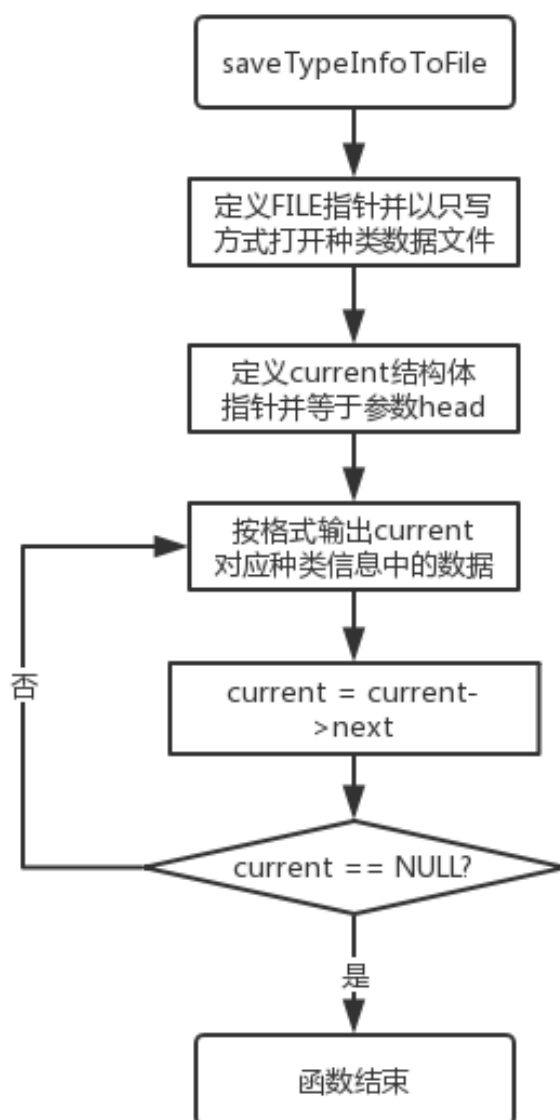


图 3.2.1 保存蔬菜种类信息流程图



### c) 信息增加

信息增加操作是指在数据中增加一项信息，用户在交互界面中新增信息后，界面层会传递用户输入的各项字符串数据给核心层，核心层申请一段内存新建一项信息结构体并根据字符串数据进行初始化，放入链表表尾，同时保存数据至文件中。

### d) 信息修改

信息修改操作是指修改数据中一项信息，用户在交互界面中修改信息后，界面层会传递用户输入的各项字符串数据给核心层，核心层通过遍历的方式找到内存中符合条件的信息结构体并修改其数据，同时保存数据至文件中。

### e) 信息删除

信息删除操作是指删除数据中一项信息，用户在交互界面中删除信息后，界面层会传递用户输入的各项字符串数据给核心层，核心层通过遍历的方式找到内存中符合条件的信息结构体，将该结构体的前一节点与后一节点链接，释放删除结构体，对其余信息进行重新编号，同时保存数据至文件中。

### f) 信息搜索

信息搜索操作是指在数据中搜索符合条件的信息，用户在交互界面中搜索信息后，界面层会传递用户输入的各项字符串数据给核心层，核心层通过遍历的方式找到内存中符合条件的信息结构体，并将其编号存储在 `int` 数组中，之后将存有满足条件信息编号的 `int` 数组返回。

### g) 信息排序

本程序在排序算法中选择了归并排序算法。归并排序（MERGE-SORT）是建立在归并操作上的一种有效的排序算法,该算法是采用分治法（Divide and Conquer）的一个非常典型的应用。将已有序的子序列合并，得到完全有序的序列；即先使每个子序列有序，再使子序列段间有序。算法的时间复杂度为  $O(n \log n)$ ，空间复杂度为  $O(n)$ 。归并算法的排序示意图如图 3.7.0 所示。



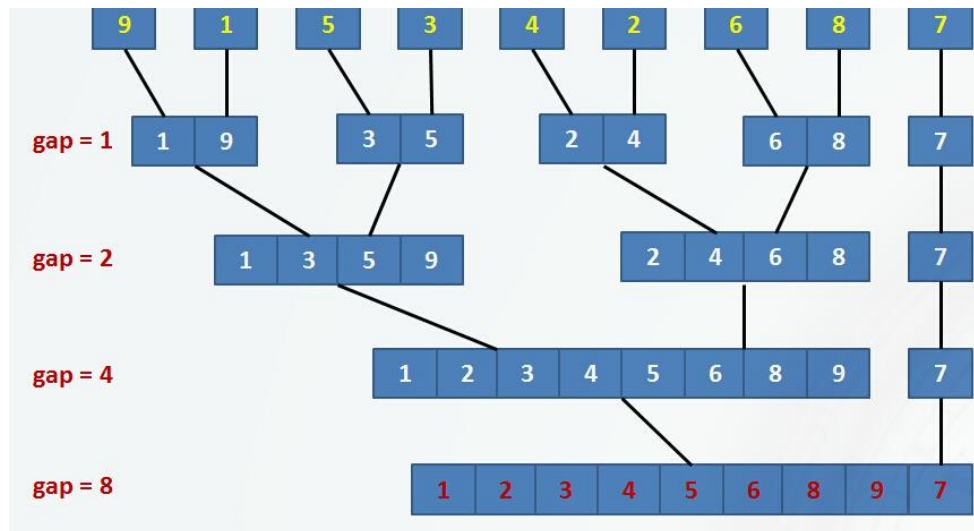


图 3.7.0 归并算法示意图

为了满足程序的多种排序需求，程序定义了多种排序方式，将其定义在枚举 `enum` 中，算法根据排序方式自动选择比较对象并将排序后的链表返回。链表归并排序函数声明如下：

```
plantInfo PlantInfoListMergeSort (plantInfo first, int sortType);
```

#### h) 数据统计

数据统计功能用于对蔬菜数据进行和运算、求平均数运算、计数器运算等统计操作，实现方式为按用户要求的信息类型对信息进行遍历并将需要的统计数据统计，统计后返回统计结果。

#### i) 链表内存释放

为了维持系统的高效，防止程序过多地占用系统资源，程序设计了释放链表内存的函数。本函数通过遍历链表对链表中每一个成员释放内存的方式释放整个链表的内存。



## 五、系统实现

现在展示程序所有部分的具体实现。

### 1. 源程序文件结构

```
./PROJECT_ROOT
| CHomework.pro // QtCreator 项目配置文件
| CHomework.pro.user
|
|— CppSource // 界面层 C++源文件目录
|   add_plantinfo_dialog.cpp // 新增蔬菜种植信息窗口
|   add_plantinfo_dialog.h
|   add_type_dialog.cpp // 新增蔬菜种类信息窗口
|   add_type_dialog.h
|   add_vegtype_dialog.cpp // 新增蔬菜基本信息窗口
|   add_vegtype_dialog.h
|   delete_plantinfo_dialog.cpp // 删除蔬菜种植信息窗口
|   delete_plantinfo_dialog.h
|   delete_type_dialog.cpp // 删除蔬菜种类信息窗口
|   delete_type_dialog.h
|   delete_veginfo_dialog.cpp // 删除蔬菜基本信息窗口
|   delete_veginfo_dialog.h
|   edit_plantinfo_dialog.cpp // 修改蔬菜种植信息窗口
|   edit_plantinfo_dialog.h
|   edit_type_dialog.cpp // 修改蔬菜种类信息窗口
|   edit_type_dialog.h
|   edit_veginfo_dialog.cpp // 修改蔬菜基本信息窗口
|   edit_veginfo_dialog.h
|   imformation_dialog.cpp // 显示提示信息窗口
|   imformation_dialog.h
|   main.cpp // 形式上的 main 函数
|   mainwindow.cpp // 主窗口
|   mainwindow.h
|   notfound_dialog.cpp // 无法找到时弹出的提示窗口
|   notfound_dialog.h
|   range_input_dialog.cpp // 输入范围窗口
|   range_input_dialog.h
|   search2_input_dialog.cpp // 二项搜索窗口
|   search2_input_dialog.h
|   searchinput_dialog.cpp // 单项搜索窗口
|   searchinput_dialog.h
|   vegcode_input_dialog.cpp // 输入蔬菜基本信息编号窗口
|   vegcode_input_dialog.h
|   yearinput_dialog.cpp // 输入年份窗口
|   yearinput_dialog.h
|
|— Csource // 核心层 C 源文件目录
|   cppincludes.h // C++调用 C 头文件的头文件
```



```
|      ListAll.c           // 构建十字交叉链表
|      ListAll.h
|      main.c             // 用于测试的核心层 main 函数
|      MyStringUtil.c     // 我的字符串处理函数库
|      MyStringUtil.h
|      PlantInfoListSort.c // 蔬菜种植信息链表归并排序
|      PlantInfoListSort.h
|      PlantingInfo.c     // 蔬菜种植信息
|      PlantingInfo.h
|      sys.h              // 程序的一些错误定义等宏定义
|      TypeInfo.c         // 蔬菜种类信息
|      TypeInfo.h
|      VegDataBackupUtil.c // 蔬菜数据备份恢复
|      VegDataBackupUtil.h
|      VegInfo.c          // 蔬菜基本信息
|      VegInfo.h
|      VegInfoListSort.c  // 蔬菜基本信息链表归并排序
|      VegInfoListSort.h
|      VegTypeEnum.c      // 蔬菜种类枚举
|      VegTypeEnum.h
|
|-----database          // 测试文件存放目录
|      backupFile.bak    // 备份文件
|      plantinfo.info    // 蔬菜种植信息数据文件
|      typeinfo.info     // 蔬菜种类信息数据文件
|      veginfo.info      // 蔬菜基本信息数据文件
|
|-----UI                // Qt 界面窗口配置文件
|      add_plantinfo_dialog.ui
|      add_type_dialog.ui
|      add_vegtype_dialog.ui
|      delete_plantinfo_dialog.ui
|      delete_type_dialog.ui
|      delete_veginfo_dialog.ui
|      edit_plantinfo_dialog.ui
|      edit_type_dialog.ui
|      edit_veginfo_dialog.ui
|      information_dialog.ui
|      mainwindow.ui
|      notfound_dialog.ui
|      range_input_dialog.ui
|      search2_input_dialog.ui
|      searchinput_dialog.ui
|      vegcode_input_dialog.ui
|      yearinput_dialog.ui
```

## 2. 源程序清单

(见附录 I)



## 六、运行测试与结果分析

### 1. 数据维护

#### a) 数据添加

- 新增蔬菜种植信息。测试用例如下：

测试用例编号	蔬菜编号	种植面积	收获重量	种植年份
1	23	2	2.333	2017
2	90	12	2.413	2008

表 6.1.1 数据添加测试用例

测试用例 1 测试截图：

新增种植作物信息

蔬菜编号 23

种植面积 2

收获重量 2.333

种植年份 2017

OK Cancel

蔬菜种植管理系统

文件(F) 新增(N) 修改(M) 删除(D) 查看(W) 统计(C) 排序(P)

欢迎使用HT菜农种植信息管理系统 v1.0

当前显示的是菜园种植概况

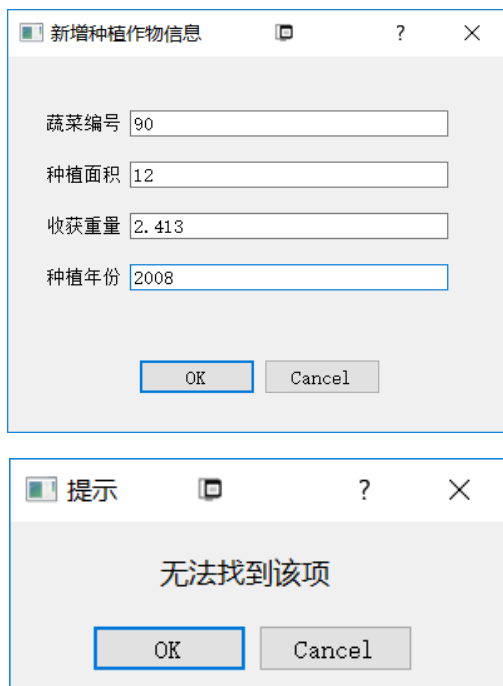
种植编号	蔬菜编号	蔬菜种类	蔬菜名称	种植面积/分	种植年份	收成/kg
43	14	果菜类	尖椒	12	2010	32.5
46	36	菌类	草菇	35	2010	22.6
47	36	菌类	草菇	35	2010	22.6
48	33	菌类	木耳	21	2017	2.8
49	38	菌类	香菇	12	2009	32.8
50	19	果菜类	丝瓜	21	2011	23.66
51	35	菌类	平菇	23	2014	56.763
52	21	瓜类	西瓜	30	2012	52.56
53	3	根茎类	葱	33	2013	110.2
54	30	叶菜类	韭菜	13	2012	23.273
55	23	瓜类	白瓜	2	2017	2.333

菜园种植数据读取完成,当前种植数目:55

测试用例 1 测试通过，程序成功添加符合输入的蔬菜种植信息并在表格底部显示。



测试用例 2 测试截图：



测试用例输入的蔬菜编号在蔬菜基本信息数据库中不存在，程序成功弹出无法找到提示，测试通过。

同时，该系统数据库中的所有蔬菜种类信息、蔬菜基本信息、蔬菜种植信息均通过软件进行新增，数据新增的操作得到了充分的检验，证明其是稳定正确的。

#### b) 数据修改

- 修改蔬菜种植信息，测试用例如下：

测试用例编号	种植编号	种植面积	收获重量	种植年份
1	23	不修改	233.23	2017
2	90	12	不修改	2008

表 6.1.2 数据修改测试用例

测试用例 1 测试截图：



当前显示的是菜园种植概况

种植编号	蔬菜编号	蔬菜种类	蔬菜名称	种植面积/分	种植年份	收成/kg
16	23	瓜类	白瓜	21	2017	2.8
17	18	果菜类	南瓜	12	1029	2.8
18	19	果菜类	丝瓜	21	2018	23.66
19	35	菌类	平菇	1	2019	5.763
20	21	瓜类	西瓜	20	2017	12.56
21	3	根茎类	葱	3	2018	100.2
22	30	叶菜类	韭菜	15	2018	63.273
23	37	菌类	金针菇	32	2016	201.5
24	29	叶菜类	菠菜	20	2014	321.1
25	20	果菜类	扁豆	15	2015	200.3
26	22	瓜类	甜瓜	10	2017	30

图 6.1.3 测试用例 1 修改前数据

修改已添加的种植蔬菜信息

留空代表不修改

蔬菜名称/编号 23

种植面积

收成/kg 233.23

种植年份 2017

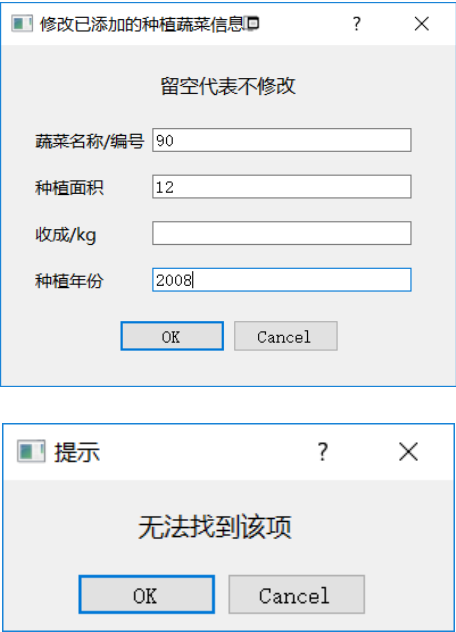
OK Cancel

当前显示的是菜园种植概况

种植编号	蔬菜编号	蔬菜种类	蔬菜名称	种植面积/分	种植年份	收成/kg
19	35	菌类	平菇	1	2019	5.763
20	21	瓜类	西瓜	20	2017	12.56
21	3	根茎类	葱	3	2018	100.2
22	30	叶菜类	韭菜	15	2018	63.273
23	37	菌类	金针菇	32	2017	233.23
24	29	叶菜类	菠菜	20	2014	321.1
25	20	果菜类	扁豆	15	2015	200.3
26	22	瓜类	甜瓜	10	2017	30
27	2	根茎类	胡萝卜	10	2017	30.2
28	3	根茎类	葱	3	2017	2.31
29	5	根茎类	小葱	10	2018	2.173

图 6.1.4 测试用例 1 修改后数据

测试用例 2 测试截图：



测试用例输入的蔬菜编号在蔬菜基本信息数据库中不存在，程序成功弹出无法找到提示，测试通过。

c) 数据删除

- 删除编号为 49,50,53,54 的蔬菜种植信息。

当前显示的是菜园种植概况						
种植编号 45	蔬菜编号 34	蔬菜种类 菌类	蔬菜名称 银耳	种植面积/分 20	种植年份 2011	收成/kg 12.50
46	14	果菜类	尖椒	12	2010	32.5
47	36	菌类	草菇	35	2010	22.6
48	33	菌类	木耳	21	2017	2.8
49	38	菌类	香菇	12	2009	32.8
50	19	果菜类	丝瓜	21	2011	23.66
51	35	菌类	平菇	23	2014	56.763
52	21	瓜类	西瓜	30	2012	52.56
53	3	根茎类	葱	33	2013	110.2
54	30	叶菜类	韭菜	13	2012	23.273
55	23	瓜类	白瓜	2	2017	2.333

图 6.1.5 测试用例修改前数据



图 6.1.5 测试用例修改后数据

测试用例输入多项种植编号，同时删除多项验证通过。

2. 数据查询

出于报告篇幅考虑，本文只展示五种数据查询方式。

a) 查看蔬菜基本信息

点击菜单【查看->蔬菜基本信息】可查看蔬菜基本信息。





图 6.2.1 查看蔬菜基本信息

## b) 查看蔬菜种类信息

点击菜单【查看->蔬菜种类信息】可查看蔬菜种类信息。



图 6.2.2 查看蔬菜种类信息

## c) 查看今年蔬菜种植信息

点击菜单【查看->按年份查看蔬菜种植信息->今年】可查看今年的蔬菜种植信息。



图 6.2.3 按年份查看蔬菜种植信息

#### d) 查看含关键字蔬菜的种植信息

点击菜单【查看->搜索->菜名关键字】可查看含关键字的蔬菜种植信息。

测试用例：搜索含有“白”字的蔬菜种植信息。

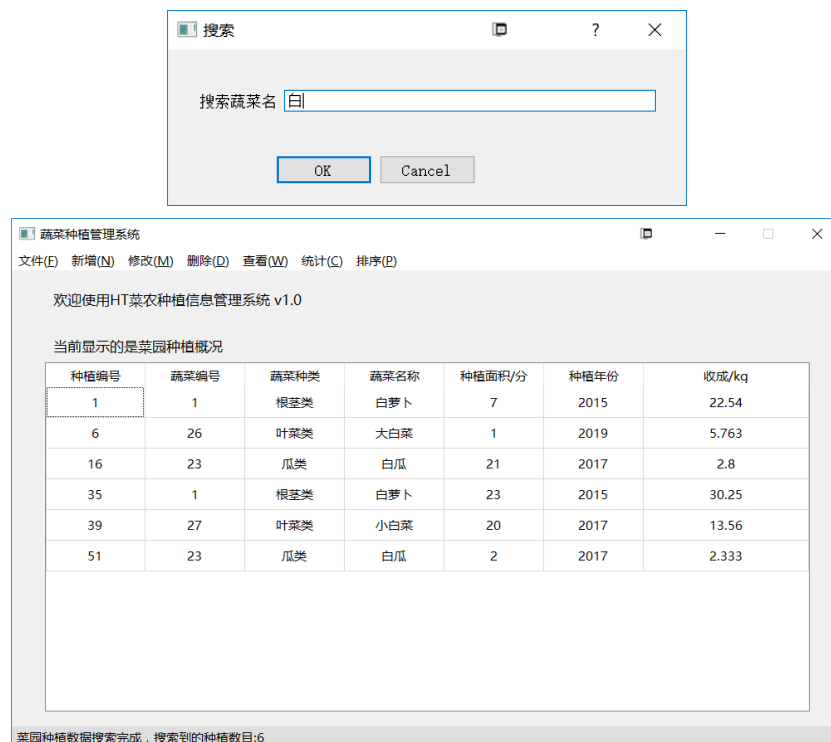


图 6.2.4 查看含关键字蔬菜的种植信息



### e) 查看含分类及营养成分的基本信息

点击菜单【查看->搜索->含分类及营养成分的基本信息】可查看含分类及营养成分的蔬菜基本信息。

测试用例：搜索含有“维生素”且蔬菜种类为 1 的蔬菜基本信息。



图 6.2.5 查看含分类及营养成分的种植信息

## 3. 数据统计

由于篇幅限制，本处只展示两项统计。

### a) 统计所有蔬菜的种植面积总和

点击菜单【统计->蔬菜种植面积总和->所有蔬菜】可统计所有蔬菜的种植面积总和。



图 6.3.1 统计所有蔬菜的种植面积总和

#### b) 统计某种蔬菜已种植数量

点击菜单【统计->某种蔬菜已种植数量】可统计某种蔬菜的种植信息个数。这里，我们使用蔬菜编号为 23 的白瓜作测试。

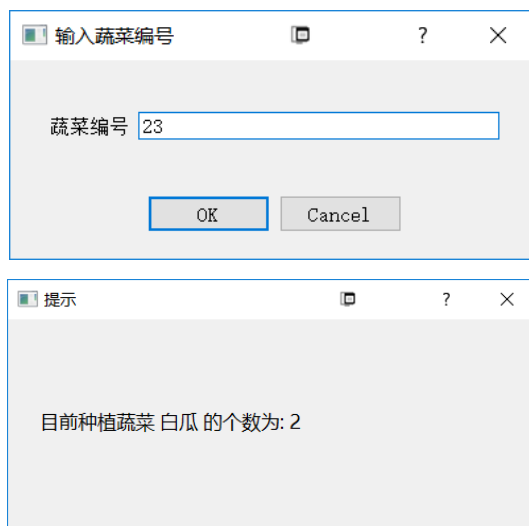


图 6.3.2 统计某种蔬菜已种植数量

## 4. 数据排序

由于篇幅限制，本处只展示两项排序。

#### a) 按种植面积排序种植信息

点击菜单【排序->按面积排序种植信息->从大到小】可将蔬菜种植信息按面积降序排列并显示。



当前显示的是菜园种植概况						
种植编号	蔬菜编号	蔬菜种类	蔬菜名称	种植面积/分	种植年份	收成/kg
47	36	菌类	草菇	35	2010	22.6
38	14	果菜类	尖椒	35	2013	100.54
23	37	菌类	金针菇	32	2017	233.23
50	21	瓜类	西瓜	30	2012	52.56
49	35	菌类	平菇	23	2014	56.763
37	35	菌类	平菇	23	2011	15.23
36	20	果菜类	扁豆	23	2007	100.1
35	1	根茎类	白萝卜	23	2015	30.25
42	33	菌类	木耳	22	2009	16.8
48	33	菌类	木耳	21	2017	2.8
43	22	叶菜类	空心菜	21	2018	23.66

图 6. 4. 2 按种植面积排序种植信息

b) 按营养成分长度排序蔬菜基本信息

点击菜单【排序->按营养成分长度排序蔬菜信息】可将蔬菜基本信息按营养成分长度降序排序并显示。

蔬菜编号	蔬菜种类	蔬菜名称	营养成分
30	叶菜类	韭菜	维生素C、维生素B1、维生素B2、尼克酸、胡萝卜素、碳水化合物及矿物质
31	叶菜类	芹菜	维生素A、维生素B1、维生素B2、维生素C和维生素P，钙，铁，磷等矿物质
10	根茎类	马铃薯	钙、磷、铁、钾、碘、胡萝卜素、硫胺素、核黄素、尼克酸、维生素C等
32	叶菜类	空心菜	维生素A原、B族维生素、维生素C及烟酸和蛋白质、脂肪、钙、磷、铁等
29	叶菜类	菠菜	类胡萝卜素、维生素C、维生素K、矿物质（钙质、铁质等）、辅酶Q10等
33	菌类	木耳	钙、磷、铁、胡萝卜素、维生素B1、维生素B2、烟酸、磷脂、固醇等
34	菌类	银耳	蛋白质、碳水化合物、脂肪、粗纤维、无机盐、少量维生素B类等
24	瓜类	黄瓜	维生素B2、维生素C、维生素E、胡萝卜素、尼克酸、钙、磷、铁等
23	瓜类	白瓜	核黄素、维生素A、维生素C、胆固醇、视黄醇当量、胡萝卜素等
19	果菜类	丝瓜	钙、磷、铁及维生素B1、维生素C、皂甙、木糖胶、瓜氨酸等
25	瓜类	苦瓜	细、幼、磷、胡萝卜素、尼克酸、硫胺素、抗坏血酸等

图 6. 3. 2 按营养成分长度排序蔬菜基本信息



## 七、总结

本次课设前后花费了很多的时间和精力，由于大一只学了 C 语言，而基于 C 语言的图形库很多都不能高效轻易地编写图形界面，于是通过查找资料找到了 Qt 这一 C++ 跨平台图形库，从头开始学习 Qt。通过查阅 Qt 手册、阅读 Qt 源码，对 Qt 的机制了解了大概，了解了一个跨平台图形库需要实现的内容，惊叹 Qt 设计的巧妙。

同时，这也是我第一次一个人独立完成桌面端软件这样大的项目，对自己的编程能力提升很大。在这次课程设计中，我先进行总体框架的构思，确定了使用 Qt 作为图形层、C 语言作为核心代码层的模式，在核心层的总体构思中，要充分考虑到代码的复杂度，尽量通过模块化的方式实现每一个功能并将结构简化，函数中需要对外显示的内容放入头文件声明，只需要内部调用的内容在 .c 源文件中声明而不暴露在外，保持代码的整洁性。同时，为了提升代码的可读性，程序中变量、函数的命名均采用驼峰命名法或下划线命名法，程序中的某些定义信息如数据文件地址均采用宏定义的方式避免程序编写时打错，并对每一个函数、变量的功能进行了清晰的解释。在代码中的逻辑部分也有清晰的备注指引，力求读者能够轻易地看懂。

数据结构设计部分，为了实现对数据文件的直观查看与编辑，自己设计了一个文本类型的数据读写的格式，通过该格式可以清晰地区分数据文件中的每一个元素与项目。在代码部分，由于本系统主要用到三种数据结构，因此，将三种数据结构的代码分别编写于三个源文件中，保证结构的清晰度，对于链表的归并排序部分，由于代码较长且相对独立，也编写于独立的源文件中。本项目的数据结构的要求是既实现十字交叉链表将三种类型数据联系在一起，又同时单独实现三种结构的单链表，因此，单独编写了将三种数据读取成十字交叉链表以及将十字交叉链表转化成单链表的代码。

图形界面设计部分，为了实现更清晰的程序功能结构，采用菜单栏的方式而非在界面中添加代码，菜单栏分类明确，菜单结构清晰，同时为了提升数据的可读性，使用了 Qt 的 QTable 表格控件来显示三种信息，相比于命令行的方式，表格的方式支持拖动，可以更轻松地阅读。为了保证结构层级清晰，需要



用户交互输入信息、显示提示的界面部分均通过新窗口的方式实现，同时窗口数能够精简尽量精简，采用数据标记的方式标记每次打开同一窗口的动作类型，做出相应的操作。

程序开发过程。首先单独进行核心层代码的编写，为了方便调试，本部分代码在集成开发环境 CLion 中使用纯 C 语言环境进行编写。在核心层编写完毕后，将项目源文件转移到 Qt 工程中进行图形层的编写，C 语言代码部分通过 C++ 的 `extern C` 拓展在 C++ 中引用。

这样大的项目在调试过程中也遇到了很多困难。由于 C 语言和 C++ 实际上是两种不同的语言，在调试中有很多核心层的错误如打开文件出错导致的段错误无法通过设置断点找到，只能看到汇编代码，给问题的查找增加了许多麻烦，只能通过回到 CLion 开发环境用纯 C 的代码调试。同时这也提醒了我在编写代码时要专心，尽量保证代码的正确性，减少 bug 出现的概率，提高开发效率。文本文件在不同平台下的读取问题也让我查找了很久。在 windows 下，文本文件每一行的最后一个字符只需要 `'n'` 而不需要 `'r'`，而 mac 或 linux 下则会读出 `'r'`，因此在跨平台移植时要注意不同平台在读写文件时的细微差异。

最后，虽然这套菜农蔬菜管理系统耗费了我很多精力，但是通过自己的编写，也学到了很多知识。通过这次课程设计，我入门了 Qt 图形库，了解了桌面图形库的工作机制，学会了通过搜索和查阅手册了解一些 API 与错误信息，学到了链表排序的归并算法，也锻炼了我的编程能力，为以后更大的项目打好基础，这次课程设计使我受益匪浅，付出的辛苦也是值得的。

## 八、参考文献

李开主编. C 语言实验与课程设计. 科学出版社, 2011.3.

曹计昌主编. C 语言与程序设计. 电子工业出版社, 2013.1.



## 九、附录

### I. 源程序清单

#### ./CHomework.pro

```
#-----  
#  
# Project created by QtCreator 2017-07-17T14:28:52  
#  
#-----  
  
QT      += core gui  
  
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets  
  
TARGET = CHomework  
TEMPLATE = app  
  
# The following define makes your compiler emit warnings if you use  
# any feature of Qt which has been marked as deprecated (the exact  
# warnings  
# depend on your compiler). Please consult the documentation of the  
# deprecated API in order to know how to port your code away from it.  
DEFINES += QT_DEPRECATED_WARNINGS  
  
# You can also make your code fail to compile if you use deprecated  
# APIs.  
# In order to do so, uncomment the following line.  
# You can also select to disable deprecated APIs only up to a certain  
# version of Qt.  
#DEFINES += QT_DISABLE_DEPRECATED_BEFORE=0x060000    # disables all  
# the APIs deprecated before Qt 6.0.0  
  
SOURCES += \  
    CppSource/main.cpp \  
    CppSource/mainwindow.cpp \  
    Csource/MyStringUtil.c \  
    Csource/PlantingInfo.c \  
    Csource/VegInfo.c \  
    Csource/VegTypeEnum.c \  
    CppSource/add_plantinfo_dialog.cpp \  
    CppSource/add_vegtype_dialog.cpp \  
    CppSource/edit_plantinfo_dialog.cpp \  
    CppSource/edit_veginfo_dialog.cpp \  
    CppSource/imformation_dialog.cpp \  
    CppSource/notfound_dialog.cpp \  
    CppSource/yearinput_dialog.cpp \  
    CppSource/delete_plantinfo_dialog.cpp \  
    CppSource/searchinput_dialog.cpp \  
    CppSource/delete_veginfo_dialog.cpp \  
    Csource/PlantInfoListSort.c \  
    Csource/ListAll.c \  
    Csource/TypeInfo.c \
```





```
Csource/VegInfoListSort.c \  
CppSource/add_type_dialog.cpp \  
CppSource/delete_type_dialog.cpp \  
CppSource/edit_type_dialog.cpp \  
CppSource/search2_input_dialog.cpp \  
CppSource/range_input_dialog.cpp \  
CppSource/vegcode_input_dialog.cpp \  
Csource/VegDataBackupUtil.c  
  
HEADERS += \  
    CppSource/mainwindow.h \  
    Csource/MyStringUtil.h \  
    Csource/PlantingInfo.h \  
    Csource/sys.h \  
    Csource/VegInfo.h \  
    Csource/VegTypeEnum.h \  
    Csource/cppincludes.h \  
    CppSource/add_plantinfo_dialog.h \  
    CppSource/add_vegtype_dialog.h \  
    CppSource/edit_plantinfo_dialog.h \  
    CppSource/edit_veginfo_dialog.h \  
    CppSource/imformation_dialog.h \  
    CppSource/notfound_dialog.h \  
    CppSource/yearinput_dialog.h \  
    CppSource/delete_plantinfo_dialog.h \  
    CppSource/searchinput_dialog.h \  
    CppSource/delete_veginfo_dialog.h \  
    Csource/PlantInfoListSort.h \  
    Csource/ListAll.h \  
    Csource/TypeInfo.h \  
    Csource/VegInfoListSort.h \  
    CppSource/add_type_dialog.h \  
    CppSource/delete_type_dialog.h \  
    CppSource/edit_type_dialog.h \  
    CppSource/search2_input_dialog.h \  
    CppSource/range_input_dialog.h \  
    CppSource/vegcode_input_dialog.h \  
    Csource/VegDataBackupUtil.h  
  
FORMS += \  
    UI/mainwindow.ui \  
    UI/add_plantinfo_dialog.ui \  
    UI/add_vegtype_dialog.ui \  
    UI/edit_plantinfo_dialog.ui \  
    UI/edit_veginfo_dialog.ui \  
    UI/imformation_dialog.ui \  
    UI/notfound_dialog.ui \  
    UI/yearinput_dialog.ui \  
    UI/delete_plantinfo_dialog.ui \  
    UI/searchinput_dialog.ui \  
    UI/delete_veginfo_dialog.ui \  
    UI/add_type_dialog.ui \  
    UI/delete_type_dialog.ui \  
    UI/edit_type_dialog.ui \  
    UI/search2_input_dialog.ui \  
    UI/range_input_dialog.ui \  
    UI/vegcode_input_dialog.ui
```

**./Csource/cppincludes.h**

```
#ifndef CPPINCLUDES_H
#define CPPINCLUDES_H
/*
 * Qt Cplusplus includes
 */

extern "C" {
#include "Csource/MyStringUtil.h"
#include "Csource/PlantingInfo.h"
#include "Csource/PlantingInfo.h"
#include "Csource/sys.h"
#include "Csource/VegInfo.h"
#include "Csource/VegTypeEnum.h"
#include "Csource/PlantInfoListSort.h"
#include "Csource/ListAll.h"
#include "Csource/TypeInfo.h"
#include "Csource/VegInfoListSort.h"
#include "Csource/VegDataBackupUtil.h"
}

#endif // CPPINCLUDES_H
```

**./Csource/ListAll.c**

```
//
// Created by hunto on 2017/8/31 0031.
//

#include "ListAll.h"

TypeInfo loadFarmListFromFile(void) {
    TypeInfo result;
    /*读取蔬菜种类信息*/
    result = loadTypeInfoFromFile();
    TypeInfo current = result;
    /*读取蔬菜基本信息*/
    FILE * vegInfoFile = fopen(VEG_INFO_PATH, "r");
    while (1) {
        vegInfo vInfo = getItemFromFile(vegInfoFile);
        if (vInfo == NULL) {
            fclose(vegInfoFile);
            break;
        }
        for (current = result; current != NULL; current =
current->next) {
            if (current->typeCode == vInfo->vegType) {
                vInfo->father = current;
                //add to this type
                vegInfo currentVInfoList = current->vInfoList;
                if (currentVInfoList == NULL) {
                    current->vInfoList = vInfo;
                    break;
                }
                while (currentVInfoList->next != NULL) {
                    currentVInfoList = currentVInfoList->next;
                }
            }
        }
    }
}
```



```
        currentVInfoList->next = vInfo;
    }
}
}
/*读取蔬菜种植信息*/
current = result;
FILE * plantInfoFile = fopen(PLANT_INFO_PATH, "r");
while (1) {
    plantInfo pInfo = getPlantItemFromFile(plantInfoFile);
    int isFind = 0; //用于跳出外层循环
    if (pInfo == NULL) {
        fclose(plantInfoFile);
        break;
    }
    for (current = result; current != NULL && isFind == 0;
current = current->next) {
        vegInfo currentVInfoList = current->vInfoList;
        for (; currentVInfoList != NULL; currentVInfoList =
currentVInfoList->next) {
            if (currentVInfoList->vegCode == pInfo->vegCode) {
                isFind = 1;
                pInfo->father = currentVInfoList;
                plantInfo currentPInfo =
currentVInfoList->pInfoList;
                if (currentPInfo == NULL) {
                    currentVInfoList->pInfoList = pInfo;
                    break;
                }
                while (currentPInfo->next != NULL) {
                    currentPInfo = currentPInfo->next;
                }
                currentPInfo->next = pInfo;
            }
        }
    }
}
return result;
}
```

```
plantInfo getPlantInfoListFromFarmList (typeInfo farmList) {
    /*过程简单不写了, 就是从头遍历把种植信息都复制一遍并接起来*/
    typeInfo tInfo = farmList;
    int isFirst = 1;
    plantInfo result;
    result = (plantInfo)malloc(sizeof(struct plant info));
    plantInfo current = result;
    for (; tInfo != NULL; tInfo = tInfo->next) {
        vegInfo vInfoList = tInfo->vInfoList;
        for (; vInfoList != NULL; vInfoList = vInfoList->next) {
            plantInfo pInfo = vInfoList->pInfoList;
            for (; pInfo != NULL; pInfo = pInfo->next) {
                if (isFirst) {
                    isFirst = 0;
                    *result = *pInfo;
                }
                else {

```



```
        current->next = (plantInfo)malloc(sizeof(struct
plant info));
        current->next->prev = current;
        current = current->next;
        *current = *pInfo;
    }
}
}
current->next = NULL;
return result;
}
```

```
vegInfo getVegInfoFromFarmList (typeInfo farmList) {
    /*过程简单不写了，就是从头遍历把基本信息都复制一遍并接起来*/
    typeInfo tInfo = farmList;
    int isfirst = 1;
    vegInfo result = (vegInfo)malloc(sizeof(struct veg info));
    vegInfo current = result;
    for (; tInfo != NULL; tInfo = tInfo->next) {
        vegInfo vInfoList = tInfo->vInfoList;
        for (; vInfoList != NULL; vInfoList = vInfoList->next) {
            if (isfirst) {
                isfirst = 0;
                *result = *vInfoList;
            }
            else {
                current->next = (vegInfo)malloc(sizeof(struct
veg info));
                current->next->prev = current;
                current = current->next;
                *current = *vInfoList;
            }
        }
        current->next = NULL;
        return result;
    }
}
```

./Csource/ListAll.h

```
//
// Created by hunto on 2017/8/31 0031.
//
```

```
#ifndef CSOURCE_LISTALL_H
#define CSOURCE_LISTALL_H
```

```
#include "VegInfo.h"
#include "PlantingInfo.h"
#include "TypeInfo.h"
```

/\*结构体互相嵌套调用\*/

```
typedef struct type info *typeInfo;
typedef struct plant info *plantInfo;
typedef struct veg info *vegInfo;
```



```
/*
 * @usage: 从文件中读取数据并建立十字交叉链表
 * @return: 建立好的十字交叉链表表头
 */
typeInfo loadFarmListFromFile(void);

/*
 * @usage: 从十字交叉链表中构建种植信息单向链表
 * @return: 建立好的种植信息链表表头
 */
plantInfo getPlantInfoListFromFarmList (typeInfo farmList);

/*
 * @usage: 从十字交叉链表中构建基本信息单向链表
 * @return: 建立好的基本信息链表表头
 */
vegInfo getVegInfoFromFarmList (typeInfo farmList);
#endif //CSOURCE_LISTALL_H

./Csource/MyStringUtil.c

//
// Created by 黄涛 on 2017/7/17.
//

#include "MyStringUtil.h"

int getIntFromString (char * head){
    char * str = head;
    int result = 0;
    while (*(++str) != '\0');
    str--;
    for (int i = 0; ; i++, str--) {
        result += (*str - '0') * pow(10, i);
        if (str == head)
            break;
    }
    return result;
}

char * getStringFromInt (int num) {
    char * buffer = (char *) malloc(sizeof(char) * 10);
    buffer[9] = '\0';
    int a = num, i;
    for (i = 8; i >= 0; i--){
        buffer[i] = a % 10 + '0';
        a = a/10;
    }
    for (i = 0; buffer[i] == '0'; i++);
    return (buffer + i);
}

float getFloatFromString (char *head) {
    return atof(head);
}

char * getStringFromFloat (float num) {
    char *result = (char *) malloc(sizeof(char) * 20);
    sprintf(result, "%.3f", num);
}
```



```
        return result;
    }

    int strinstr(char *str, char *inthis) {
        char *small = str, *big = inthis;
        while(*big != *small) {
            big++;
        }
        while(*(big++) == *(small++)){
            if (*small == '\n' || *small == '\0') {
                return 1;
            }
        }
        return 0;
    }
}
```

### ./Csource/MyStringUtil.h

```
//
// Created by 黄涛 on 2017/7/17.
//

#ifndef CHOMEWORK_MYSTRINGUTIL_H
#define CHOMEWORK_MYSTRINGUTIL_H

#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"

/*
 * @param: 字符串
 * @usage: 字符串转 int
 * @return: int
 */
int getIntFromString (char * head);

/*
 * @param: int
 * @usage: int 转字符串
 * @return: 字符串
 */
char * getStringFromInt (int num);

/*
 * @param: 字符串
 * @usage: 字符串转 float
 * @return: float
 */
float getFloatFromString (char *head);

/*
 * @param: float
 * @usage: float 转字符串
 * @return: 字符串
 */
```



```
char * getStringFromFloat (float num);

/*
 * @params: str:检测包含性的字符串  inthis:大的字符串
 * @usage: 检测一个字符串是否包含在另一字符串中
 * @return: 包含--1 不包含--0
 */
int strstr(char *str, char *inthis);

#endif //CHOMEWORK_MYSTRINGUTIL_H

./Csource/PlantInfoListSort.c

//
// Created by hunto on 2017/8/31 0031.
//

#include "PlantInfoListSort.h"
#include "MyStringUtil.h"
#include "VegInfo.h"

/*仅需内部调用因此在内部声明保证代码整洁性*/
void PlantInfoListDivide(plantInfo first, plantInfo *left, plantInfo
*right);
plantInfo PlantinfoListMergeInPlace (plantInfo left, plantInfo right,
int sortType);
int PlantInfoCompare (plantInfo a, plantInfo b, int sortType);

plantInfo PlantInfoListCopy(plantInfo info) {
    plantInfo current, copy = NULL, prev;
    plantInfo result = (plantInfo)malloc(sizeof(struct plant info));
    *result = *info;
    prev = result;
    current = info->next;
    while (current != NULL) {
        copy = (plantInfo)malloc(sizeof(struct plant info));
        *copy = *current;
        prev->next = copy;
        prev = copy;
        current = current->next;
    }
    copy->next = NULL;
    return result;
}

plantInfo PlantInfoListMergeSort(plantInfo first, int sortType) {
    plantInfo left, right;
    if (first == NULL || first->next == NULL) {
        return first;
    }
    PlantInfoListDivide(first, &left, &right);
    left = PlantInfoListMergeSort(left, sortType);
    right = PlantInfoListMergeSort(right, sortType);
    return PlantinfoListMergeInPlace(left, right, sortType);
}

void PlantInfoListDivide(plantInfo first, plantInfo *left, plantInfo
*right) {
```



```
    plantInfo slow = first, fast = first->next;
    while (fast != NULL) {
        fast = fast->next;
        if (fast != NULL) {
            fast = fast->next;
            slow = slow->next;
        }
    }
    *left = first;
    *right = slow->next;
    slow->next = NULL;
}

plantInfo PlantinfoListMergeInPlace (plantInfo left, plantInfo right,
int sortType) {
    plantInfo tmp head = (plantInfo)malloc(sizeof(struct
plant info));
    plantInfo first = tmp head;
    for (; left != NULL && right != NULL; first = first->next) {
        if (PlantInfoCompare(left, right, sortType)) {
            first->next = left;
            left = left->next;
        }
        else {
            first->next = right;
            right = right->next;
        }
    }
    first->next = left != NULL ? left : right;
    first = tmp head->next;
    free(tmp head);
    return first;
}

#ifdef 0
/*方法2*/
if (left == NULL) {
    return right;
}
if (right == NULL) {
    return left;
}
plantInfo first;
if (left->area <= right->area) {
    first = left;
    first->next = PlantinfoListMergeInPlaceByArea(left->next,
right);
}
else {
    first = right;
    first->next = PlantinfoListMergeInPlaceByArea(left,
right->next);
}
return first;
#endif
}
/*
 * @usage: 按照指定的排序条件判断两结构体位置先后
 * @param: sortType:排序类型

```





```
* @return: 0-- a 在 b 后 1-- a 在 b 前
*/
int PlantInfoCompare (plantInfo a, plantInfo b, int sortType) {
    switch (sortType) {
        case SORTAREA:
            if (a->area <= b->area) {
                return 1;
            }
            else return 0;
        case SORTAREAR:
            if (a->area <= b->area) {
                return 0;
            }
            else return 1;
        case SORTWEIGHT:
            if (a->weight <= b->weight) {
                return 1;
            }
            else return 0;
        case SORTWEIGHTR:
            if (a->weight <= b->weight) {
                return 0;
            }
            else return 1;
        case SORTYEAR:
            if (getIntFromString(a->year) <= getIntFromString(b->year)) {
                return 1;
            }
            else return 0;
        case SORTYEARR:
            if (getIntFromString(a->year) <= getIntFromString(b->year)) {
                return 0;
            }
            else return 1;
        case SORTTYPE:
            if (a->father->vegType <= b->father->vegType) {
                return 1;
            }
            else return 0;
        case SORTDEFAULT:
            if (a->id <= b->id) {
                return 1;
            }
            else return 0;
    }
    return 0;
}
```

./Csource/ PlantInfoListSort.h

```
//
// Created by hunto on 2017/8/31 0031.
//
```

```
#ifndef CSOURCE_LISTSORT_H
#define CSOURCE_LISTSORT_H

#include "PlantingInfo.h"
```



```
/*
 * @usage: 将蔬菜种植信息链表按条件归并排序
 * @params: first--蔬菜种植信息链表表头    sortType--排序类型
 * @return: 排序后的蔬菜种植信息链表表头
 */
plantInfo PlantInfoListMergeSort (plantInfo first, int sortType);

/*
 * @usage: 将蔬菜种植信息链表复制一份
 * @param: 蔬菜种植信息链表表头
 * @return: 复制后的蔬菜种植信息链表表头
 */
plantInfo PlantInfoListCopy(plantInfo info);

/* 排序种类的枚举, 从前至后分别为, 按面积升降序排列、
 * 按重量升降序排列、按年份升降序排列、按类型排列、默认排序*/
enum sort type {
    SORTAREA, SORTAREAR, SORTWEIGHT, SORTWEIGHTR,
    SORTYEAR, SORTYEARR, SORTTYPE, SORTDEFAULT
};

#endif //CSOURCE_LISTSORT_H

./Csource/PlantingInfo.c

//
// Created by 黄涛 on 2017/7/17.
//

#include "PlantingInfo.h"

plantStructInfo loadPlantInfoFromFile(void) {
    FILE * plantInfoFile = fopen(PLANT_INFO_PATH, "r");
    plantStructInfo result = (plantStructInfo) malloc(sizeof(struct
plant struct info));
    plantInfo head = getPlantItemFromFile(plantInfoFile);
    plantInfo current = head;
    head->prev = NULL;
    while (1) {
        plantInfo info = getPlantItemFromFile(plantInfoFile);
        if (info == NULL) {
            current->next = NULL;
            fclose(plantInfoFile);
            result->head = head;
            result->tail = current;
            result->num = current->id;
            return result;
        }
        current->next = info;
        info->prev = current;
        current = info;
    }
}

void savePlantInfoToFile(plantInfo head) {
    plantInfo current = head;
```



```
FILE * plantInfoFile = fopen(PLANT_INFO_PATH, "w");
while (1) {
    /*种植作物 ID*/
    fputs("{\\", plantInfoFile);
    fputs(getStringFromInt(current->id), plantInfoFile);
    fputs("\\',\\'", plantInfoFile);
    /*蔬菜编号*/
    fputs(getStringFromInt(current->vegCode), plantInfoFile);
    fputs("\\',\\'", plantInfoFile);
    /*种植面积*/
    fputs(getStringFromInt(current->area), plantInfoFile);
    fputs("\\',\\'", plantInfoFile);
    /*收获重量*/
    fputs(getStringFromFloat(current->weight), plantInfoFile);
    fputs("\\',\\'", plantInfoFile);
    /*种植年份*/
    fputs(current->year, plantInfoFile);
    fputs("\\'}\\n", plantInfoFile);

    current = current->next;
    if (current == NULL) {
        break;
    }
}
fclose(plantInfoFile);
}

plantInfo getPlantItemFromFile(FILE * plantInfoFile) {
    if (fgetc(plantInfoFile) != '{') {
        return NULL;
    }
    while (fgetc(plantInfoFile) != '\\');
    char c;
    char *buffer = (char *)malloc(sizeof(char) * 20);
    char *bufferHead = buffer;
    plantInfo result = (plantInfo) malloc(sizeof(struct plant info));
    /*种植作物 id*/
    while((c = fgetc(plantInfoFile)) != '\\') {
        * (buffer++) = c;
    }
    *buffer = '\\0';
    result->id = getIntFromString(bufferHead);
    /*蔬菜编号*/
    fgetc(plantInfoFile);
    fgetc(plantInfoFile);
    buffer = bufferHead;
    while((c = fgetc(plantInfoFile)) != '\\') {
        * (buffer++) = c;
    }
    *buffer = '\\0';
    result->vegCode = getIntFromString(bufferHead);
    /*种植面积*/
    fgetc(plantInfoFile);
    fgetc(plantInfoFile);
    buffer = bufferHead;
    while((c = fgetc(plantInfoFile)) != '\\') {
        * (buffer++) = c;
```



```
    }
    *buffer = '\\0';
    result->area = getIntFromString(bufferHead);
    /*收获重量*/
    fgetc(plantInfoFile);
    fgetc(plantInfoFile);
    buffer = bufferHead;
    while((c = fgetc(plantInfoFile)) != '\\') {
        * (buffer++) = c;
    }
    *buffer = '\\0';
    result->weight = getFloatFromString(bufferHead);
    /*种植年份*/
    fgetc(plantInfoFile);
    fgetc(plantInfoFile);
    buffer = bufferHead;
    while((c = fgetc(plantInfoFile)) != '\\') {
        * (buffer++) = c;
    }
    *buffer = '\\0';
    strcpy(result->year, bufferHead);
    fgetc(plantInfoFile);
    fgetc(plantInfoFile);
    result->next = NULL;
    return result;
}

plantInfo createPlantInfoItem(int id, char * vegCode, char * area,
char *weight, char * year) {
    plantInfo info = (plantInfo) malloc(sizeof(struct plant info));
    info->id = id;
    if(*vegCode >= '0' && *vegCode <= '9')
        info->vegCode = getIntFromString(vegCode);
    else {
        vegStructInfo veginfo = loadVegInfoFromFile();
        info->vegCode = getVegInfoByName(veginfo->head,
vegCode)->vegCode;
        /*释放内存*/
        freeVegInfo(veginfo->head);
        free(veginfo);
    }
    info->area = getIntFromString(area);
    info->weight = getFloatFromString(weight);
    strcpy(info->year, year);
    return info;
}

void addPlantInfoItem(plantInfo tail,plantInfo addItem) {
    tail->next = addItem;
    addItem->prev = tail;
    addItem->next = NULL;
}

void removePlantInfoItem(plantInfo removeItem) {
    plantInfo current;
    current = removeItem->next;
    for(int i = removeItem->id; current != NULL; i++, current =
```



```
current->next){
    current->id = i;
}
if (removeItem->id == 1) {
    /*删除第一项--直接把第二项复制到第一项*/
    plantInfo next = removeItem->next;
    *removeItem = *next;
    free(next);
}
else {
    removeItem->prev->next = removeItem->next;
    removeItem->next->prev = removeItem->prev;
    free(removeItem);
}
}

void removePlantInfoItems(plantInfo removeItem) {
    if (removeItem->id == 1) {
        /*删除第一项--直接把第二项复制到第一项*/
        plantInfo next = removeItem->next;
        *removeItem = *next;
        free(next);
    }
    else {
        removeItem->prev->next = removeItem->next;
        removeItem->next->prev = removeItem->prev;
        free(removeItem);
    }
}

void renamePlantInfoID(plantInfo head) {
    plantInfo current = head;
    for(int i = 1; current != NULL; i++, current = current->next){
        current->id = i;
    }
}

plantInfo getPlantInfoByID(plantInfo head, int id) {
    plantInfo result = head;
    while (1) {
        if (result->id == id){
            return result;
        }
        result = result->next;
        if (result == NULL) {
            printf("未找到该种植 ID: %d", id);
            return NULL;
        }
    }
}

plantInfo getPlantInfoByVegCode(plantInfo head, int vegCode) {
    plantInfo result = head;
    while (1) {
        if (result->id == vegCode){
            return result;
        }
    }
}
```



```
        result = result->next;
    if (result == NULL) {
        printf("未种植该蔬菜");
        return NULL;
    }
}

int getPlantAreaAll(plantInfo head) {
    plantInfo current = head;
    int area = 0;
    for (; current != NULL; current = current->next) {
        area += current->area;
    }
    return area;
}

int getPlantAreaYear(plantInfo head, int year) {
    plantInfo current = head;
    int area = 0;
    for (; current != NULL; current = current->next) {
        if (getIntFromString(current->year) == year)
            area += current->area;
    }
    return area;
}

int getPlantAreaVegType(plantInfo head, vegInfo vegHead, int VegType)
{
    plantInfo current = head;
    int area = 0;
    for (; current != NULL; current = current->next) {
        if (getVegInfoByCode(vegHead, current->vegCode)->vegType -
'0' == VegType)
            area += current->area;
    }
    return area;
}

float getPlantWeightAll(plantInfo head) {
    plantInfo current = head;
    float weight = 0;
    for (; current != NULL; current = current->next) {
        weight += current->weight;
    }
    return weight;
}

float getPlantWeightYear(plantInfo head, int year) {
    plantInfo current = head;
    float weight = 0;
    for (; current != NULL; current = current->next) {
        if (getIntFromString(current->year) == year)
            weight += current->weight;
    }
    return weight;
}
```



```
}

float getPlantWeightVegType(plantInfo head, vegInfo vegHead, int
VegType) {
    plantInfo current = head;
    float weight = 0;
    for (; current != NULL; current = current->next) {
        if (getVegInfoByCode(vegHead, current->vegCode) ->vegType -
'0' == VegType)
            weight += current->weight;
    }
    return weight;
}

void freePlantInfo(plantInfo head) {
    plantInfo current = head, prev = head;
    while (1) {
        if (current->next == NULL) {
            free(current);
            break;
        }
        prev = current;
        current = current->next;
        free(prev);
    }
}
```

### ./Csource/PlantingInfo.h

```
//
// Created by 黄涛 on 2017/7/17.
//

#ifndef CHOMEWORK_PLANTINGINFO_H
#define CHOMEWORK_PLANTINGINFO_H

#include "sys.h"
#include "stdio.h"
#include "stdlib.h"
#include "MyStringUtil.h"
#include "string.h"
#include "VegInfo.h"
#include "ListAll.h"

/*plantinfo 数据路径 debug 时用绝对路径*/
#if 0
#define PLANT_INFO_PATH "plantinfo.info"
#else
#define PLANT_INFO_PATH
"C:\\\\Users\\\\hunto\\\\Documents\\\\CHomework\\\\database\\\\plantinfo.info"
#endif

/*
 * 种植信息结构体
 * @id: 种植编号 @vegCode: 蔬菜编号 @area:面积 @weight:收成 @year:年
份
 */
```



```
typedef struct veg info *vegInfo;

typedef struct plant info {
    struct plant info *prev, *next;
    vegInfo father;
    int id;
    int vegCode;
    int area;
    float weight;
    char year[5];
}*plantInfo;

/*
 * 存储种植信息链表的结构体
 * @head:链表表头 @tail:链表尾部 @num:链表成员数量
 */
typedef struct plant struct info {
    plantInfo head, tail;
    int num;
}*plantStructInfo;

/*
 * @usage: 从文件中读取种植信息存储至链表中
 * @return: 链表信息
 */
plantStructInfo loadPlantInfoFromFile(void);

/*
 * @usage: 将种植信息链表保存到文件
 * @param: 链表表头
 */
void savePlantInfoToFile(plantInfo head);

/*
 * @usage: 从文件指针处开始读取一项种植信息 用于loadPlantInfoFromFile 中
 * @param: 文件指针
 * @return: 种植信息结构体
 */
plantInfo getPlantItemFromFile(FILE * plantInfoFile);

/*
 * @usage: 增加一项种植信息到链表尾部
 * @params: tail:链表尾部 additem:要增加的种植信息结构体
 */
void addPlantInfoItem(plantInfo tail,plantInfo addItem);

/*
 * @usage: 从链表移除一项种植信息
 * @param: 要移除的种植信息结构体
 */
void removePlantInfoItem(plantInfo removeItem);

/*
 * @usage: 从链表移除一项种植信息 用于需要同时删除多项的情况--不会重排编号
 * @param: 要移除的种植信息结构体
 */
```





```
*/
void removePlantInfoItems (plantInfo removeItem);

/*
 * @usage: 种植信息重新编号--表头为1 从小到大编号
 * @param: 种植信息链表表头
 */
void renamePlantInfoID (plantInfo head);

/*
 * @usage: 通过种植编号获取种植信息结构体
 * @params: head:种植信息链表表头 id:要获取的编号
 * @return: 获取的种植信息结构体 NULL 为未找到
 */
plantInfo getPlantInfoByID (plantInfo head, int id);

/*
 * @usage: 通过蔬菜编号获取种植信息结构体
 * @params: head:种植信息链表表头 vegCode:要获取的蔬菜编号
 * @return: 获取的种植信息结构体 NULL 为未找到
 */
plantInfo getPlantInfoByVegCode (plantInfo head, int vegCode);

/*
 * @usage: 创建种植信息结构体 用于与C++的UI 交互
 * @params: 同结构体成员变量
 * @return: 创建的种植信息结构体 NULL 为未创建
 */
plantInfo createPlantInfoItem (int id, char * vegCode, char * area,
char * weight, char * year);

/*
 * @usage: 释放链表
 * @param: 链表表头
 */
void freePlantInfo (plantInfo head);

/*
 * @usage: 统计种植面积总和
 * @param: 链表表头
 * @return: 面积总和
 */
int getPlantAreaAll (plantInfo head);

/*
 * @usage: 统计一年的种植面积总和
 * @params: head:链表表头 year:要统计的年份
 * @return: 该年面积总和
 */
int getPlantAreaYear (plantInfo head, int year);

/*
 * @usage: 统计一年的种植面积总和
 * @params: head:链表表头 vegHead:蔬菜种类信息的链表表头 vegType:蔬菜种类
 * @return: 该种类面积总和
 */
int getPlantAreaVegType (plantInfo head, vegInfo vegHead, int
```



```
VegType);

/*
 * @usage: 统计收成总和
 * @param: 链表表头
 * @return: 收成总和
 */
float getPlantWeightAll(plantInfo head);

/*
 * @usage: 统计一年的收成总和
 * @params: head:链表表头 year:要统计的年份
 * @return: 该年收成总和
 */
float getPlantWeightYear(plantInfo head, int year);

/*
 * @usage: 统计一年的收成总和
 * @params: head:链表表头 vegHead:蔬菜种类信息的链表表头 vegType:蔬菜种类
 * @return: 该种类收成总和
 */
float getPlantWeightVegType(plantInfo head, vegInfo vegHead, int
VegType);

#endif //CHOMEWORK_PLANTINGINFO_H
```

### ./Csource/TypeInfo.c

```
//
// Created by hunto on 2017/8/31 0031.
//

#include "TypeInfo.h"

typeInfo getItemFromFile(FILE * typeInfoFile);

typeInfo loadTypeInfoFromFile (void) {
    FILE * typeInfoFile = fopen(TYPE_INFO_PATH, "r");
    typeInfo head = getItemFromFile(typeInfoFile);
    typeInfo current = head;
    while (1) {
        current->next = getItemFromFile(typeInfoFile);
        if (current->next == NULL) {
            current->next=NULL;
            fclose(typeInfoFile);
            return head;
        }
        current = current->next;
    }
}

void saveTypeInfoToFile(typeInfo head) {
    typeInfo current = head;
    FILE * typeInfoFile = fopen(TYPE_INFO_PATH, "w");
    while (1) {
        /*种类编号*/
        fputs("\\", typeInfoFile);
    }
}
```



```
fputc(current->typeCode, typeInfoFile);
fputs("\',\'", typeInfoFile);
/*种类名称*/
fputs(current->typeName, typeInfoFile);
fputs("\'}\n", typeInfoFile);

if (current->next == NULL) {
    fputs("\n", typeInfoFile);
    break;
}
current = current->next;
}
fclose(typeInfoFile);
}

typeInfo getItemFromFile(FILE * typeInfoFile) {
    if (fgetc(typeInfoFile) != '{') {
        return NULL;
    }
    while (fgetc(typeInfoFile) != '\\');
    char c;
    char *buffer = (char *)malloc(sizeof(char) * 20);
    char *bufferHead = buffer;
    typeInfo result = (typeInfo) malloc(sizeof(struct type info));
    /*种类编号*/
    result->typeCode = fgetc(typeInfoFile);
    /*种类名*/
    fgetc(typeInfoFile);
    fgetc(typeInfoFile);
    fgetc(typeInfoFile);
    buffer = bufferHead;
    while ((c = fgetc(typeInfoFile)) != '\\') {
        * (buffer++) = c;
    }
    *buffer = '\\0';
    strcpy(result->typeName, bufferHead);
    free(bufferHead);
    fgetc(typeInfoFile); //'}'
    fgetc(typeInfoFile); //'\\n'
    result->next = NULL;
    result->vInfoList = NULL;
    return result;
}

void addTypeInfoItem (typeInfo head, char *addItemName) {
    typeInfo current = head;
    while (current->next != NULL) {
        current = current->next;
    }
    typeInfo addItem = (typeInfo)malloc(sizeof(struct type info));
    addItem->next = NULL;
    addItem->vInfoList = NULL;
    addItem->typeCode = current->typeCode + 1;
    strcpy(addItem->typeName, addItemName);
    current->next = addItem;
}
```



```
int deleteTypeInfo (TypeInfo head, char typeCode) {
    TypeInfo current = head;
    TypeInfo prev = NULL;
    for (; current != NULL; prev = current, current = current->next)
    {
        if (current->typeCode == typeCode) {
            if (prev == NULL) {
                prev = current->next;
                *current = *(current->next);
                free (prev);
            }
            else {
                prev->next = current->next;
                free(current);
            }
            current = head;
            for (char code = '1'; current != NULL; code++, current =
current->next) {
                current->typeCode = code;
            }
            return 1;
        }
    }
    return 0;
}

int editTypeInfo(TypeInfo head, char typeCode, char *typeName) {
    TypeInfo current = head;
    for (; current != NULL; current = current->next) {
        if (current->typeCode == typeCode) {
            strcpy(current->typeName, typeName);
            return 1;
        }
    }
    return 0;
}
```

### ./Csource/TypeInfo.h

```
//
// Created by hunto on 2017/8/31 0031.
//

#ifndef CSOURCE_TYPEINFO_H
#define CSOURCE_TYPEINFO_H

#include "VegInfo.h"
#include "ListAll.h"
#include "PlantingInfo.h"

/*蔬菜种类数据路径 debug 时用绝对路径*/
#if 0
#define TYPE_INFO_PATH "typeinfo.info"
#else
#define TYPE_INFO_PATH
"C:\\Users\\hunto\\Documents\\CHomework\\database\\typeinfo.info"
#endif

/*互相嵌套调用结构体时声明*/
```



```
typedef struct veg info *vegInfo;

/*
 * 蔬菜种类信息结构体
 * next -- 下一节点
 * vInfoList -- 子蔬菜基本信息链表
 * typeCode -- 种类编号
 * typeName -- 种类名称
 */
typedef struct type info {
    struct type info *next;
    vegInfo vInfoList;
    char typeCode;
    char typeName[10];
}*typeInfo;

/*
 * @usage: 从文件中读取蔬菜种类信息单向链表
 */
typeInfo loadTypeInfoFromFile (void);

/*
 * @usage : 保存蔬菜种类信息链表至文件
 */
void saveTypeInfoToFile(typeInfo head);

/*
 * @usage : 新增蔬菜种类信息链接至链表表尾
 */
void addTypeInfoItem (typeInfo head ,char *addItemName);

/*
 * @usage: 删除蔬菜种类信息
 * @param: typeCode -- 要删除的种类编号
 * @return: 0--删除失败 1--删除成功
 */
int deleteTypeInfo (typeInfo head, char typeCode);

/*
 * @usage ; 修改蔬菜种类信息
 */
int editTypeInfo(typeInfo head, char typeCode, char *typeName);
#endif //CSOURCE_TYPEINFO_H

./Csource/VegDataBackupUtil.c

//
// Created by hunto on 2017/9/6 0006.
//

#include "VegDataBackupUtil.h"
#include "VegInfo.h"
#include "PlantingInfo.h"
#include "TypeInfo.h"

int backupDataFileTo (char *filePath) {
    char ch;
```



```
FILE *backupFile = fopen(filePath, "w");
if(backupFile == NULL) {
    //文件打开失败
    return 0;
}
//typeInfo
//文件识别码
fputs("#\n", backupFile);
FILE * typeInfoFile = fopen(TYPE_INFO_PATH, "r");
while ((ch = fgetc(typeInfoFile)) != EOF) {
    fputc(ch, backupFile);
}
fclose(typeInfoFile);
//vegInfo
//文件识别码
fputs("#\n", backupFile);
FILE * vegInfoFile = fopen(VEG_INFO_PATH, "r");
while ((ch = fgetc(vegInfoFile)) != EOF) {
    fputc(ch, backupFile);
}
fclose(vegInfoFile);
//plantInfo
//文件识别码
fputs("#\n", backupFile);
FILE * plantInfoFile = fopen(PLANT_INFO_PATH, "r");
while ((ch = fgetc(plantInfoFile)) != EOF) {
    fputc(ch, backupFile);
}
fclose(plantInfoFile);
fclose(backupFile);
return 1;
}

int getDataFileFromBackupFile (char *filePath) {
    char ch;
    FILE *backupFile = fopen(filePath, "r");
    if ((ch = fgetc(backupFile)) != '#') {
        /*标志不对, 不是备份文件*/
        return 0;
    }
    fgetc(backupFile); // '\n'
    //typeInfo
    FILE *typeInfoFile = fopen(TYPE_INFO_PATH, "w");
    while ((ch = fgetc(backupFile)) != '#') {
        fputc(ch, typeInfoFile);
    }
    fclose(typeInfoFile);
    fgetc(backupFile); // '\n'
    //vegInfo
    FILE *vegInfoFile = fopen(VEG_INFO_PATH, "w");
    while ((ch = fgetc(backupFile)) != '#') {
        fputc(ch, vegInfoFile);
    }
    fclose(vegInfoFile);
    fgetc(backupFile); // '\n'
    //plantInfo
    FILE *plantInfoFile = fopen(PLANT_INFO_PATH, "w");
```



```
    while ((ch = fgetc(backupFile)) != EOF) {
        fputc(ch, plantInfoFile);
    }
    fclose(plantInfoFile);
    fclose(backupFile);
    return 1;
}
```

#### ./Csource/VegDataBackupUtil.h

```
//
// Created by hunto on 2017/9/6 0006.
//

#ifndef CSOURCE_VEGDATABACKUPUTIL_H
#define CSOURCE_VEGDATABACKUPUTIL_H

/*
 * @usage: 将数据文件备份至指定路径
 * @return: 1--保存成功 0--保存失败
 */
int backupDataFileTo (char *filePath);

/*
 * @usage: 从指定路径读取备份文件还原至数据库
 * @return: 1--读取成功 0--读取失败
 */
int getDataFileFromBackupFile (char *filePath);

#endif //CSOURCE_VEGDATABACKUPUTIL_H
```

#### ./Csource/VegInfo.c

```
//
// Created by 黄涛 on 2017/7/17.
//
/*
 * usage : 蔬菜信息链表
 */

#include "VegInfo.h"

/*
 * @params : 链表表头, 查找的蔬菜 ID
 * @return : 蔬菜 ID 对应的蔬菜信息结构体
 */
vegInfo getVegInfoByCode(vegInfo head, int vegCode) {
    vegInfo result = head;
    while (1) {
        if (result->vegCode == vegCode){
            return result;
        }
        result = result->next;
        if (result == NULL) {
            printf("未找到该蔬菜 ID: %d", vegCode);
            return NULL;
        }
    }
}
```



```
}

/*
 * @params : 链表表头, 查找的蔬菜名
 * @return : 蔬菜名对应的蔬菜信息结构体
 */
vegInfo getVegInfoByName(vegInfo head, char *vegName) {
    vegInfo result = head;
    while (1) {
        if (strcmp(result->vegName, vegName)==0) {
            return result;
        }
        result = result->next;
        if (result == NULL) {
            printf("未找到该蔬菜: %s", vegName);
            return NULL;
        }
    }
}

/*
 * @return : 蔬菜信息链表的头尾及长度
 */
vegStructInfo loadVegInfoFromFile(void) {
    FILE * vegInfoFile = fopen(VEG_INFO_PATH, "r");
    vegStructInfo result = (vegStructInfo) malloc(sizeof(struct
veg struct info));
    vegInfo head = getItemFromFile(vegInfoFile);
    head->prev = NULL;
    vegInfo current = head;
    while (1) {
        vegInfo info = getItemFromFile(vegInfoFile);
        if (info == NULL) {
            current->next=NULL;
            result->head = head;
            result->tail = current;
            result->itemNum = current->vegCode;
            fclose(vegInfoFile);
            return result;
        }
        current->next = info;
        info->prev = current;
        current = info;
    }
}

void saveVegInfoToFile(vegInfo head) {
    vegInfo current = head;
    FILE * vegInfoFile = fopen(VEG_INFO_PATH, "w");
    while (1) {
        /*蔬菜ID*/
        fputs("{\\", vegInfoFile);
        fputs(getStringFromInt(current->vegCode), vegInfoFile);
        fputs("\\',\\'", vegInfoFile);
        /*蔬菜名*/
    }
}
```





```
    fputs(current->vegName, vegInfoFile);
    fputs("\',\'", vegInfoFile);
    /*蔬菜种类*/
    fputc(current->vegType, vegInfoFile);
    fputs("\',\'", vegInfoFile);
    /*蔬菜营养成分*/
    fputs(current->nutritionFacts, vegInfoFile);
    fputs("\'}\n", vegInfoFile);

    if (current->next == NULL) {
        fputs("\n", vegInfoFile);
        break;
    }
    current = current->next;
}
fclose(vegInfoFile);
}

/*
 * @param : 存储蔬菜信息文件的 FILE 指针
 * @return : 一项蔬菜信息
 */
vegInfo getItemFromFile(FILE * vegInfoFile) {
    if (fgetc(vegInfoFile) != '{'){
        return NULL;
    }
    while (fgetc(vegInfoFile) != '\n');
    char c;
    char *buffer = (char *)malloc(sizeof(char) * 200);
    char *bufferHead = buffer;
    vegInfo result = (vegInfo) malloc(sizeof(struct veg info));
    /*蔬菜编号*/
    while ((c = fgetc(vegInfoFile)) != '\n') {
        * (buffer++) = c;
    }
    *buffer = '\0';
    result->vegCode = getIntFromString(bufferHead);
    /*蔬菜名*/
    fgetc(vegInfoFile);
    fgetc(vegInfoFile);
    buffer = bufferHead;
    while ((c = fgetc(vegInfoFile)) != '\n') {
        * (buffer++) = c;
    }
    *buffer = '\0';
    char *vegName = result->vegName;
    strcpy(vegName, bufferHead);
    /*蔬菜种类编号*/
    fgetc(vegInfoFile);
    fgetc(vegInfoFile); //移2位以忽略,"
    result->vegType = fgetc(vegInfoFile);
    fgetc(vegInfoFile);
    /*蔬菜营养成分*/
    fgetc(vegInfoFile);
    fgetc(vegInfoFile); //移2位以忽略,"
    buffer = bufferHead;
```



```
while ((c = fgetc(vegInfoFile))!='\n') {
    * (buffer++) = c;
}
*buffer = '\0';
char *nutritionFacts = result->nutritionFacts;
strcpy(nutritionFacts, bufferHead);
free(bufferHead);
fgetc(vegInfoFile); //'}'
fgetc(vegInfoFile); //'\\n'
result->next = NULL;
result->pInfoList = NULL;
return result;
}

void addVegInfoItem(vegInfo tail, vegInfo addItem) {
    tail->next = addItem;
    addItem->prev = tail;
    addItem->next = NULL;
}

vegInfo createVegInfoItem(int vegCode, char * vegName, char *
vegType, char * nutritionFacts) {
    vegInfo info = (vegInfo) malloc(sizeof(struct veg info));
    info->vegCode = vegCode;
    strcpy(info->vegName, vegName);
    if(*vegType >= '0' && *vegType <= '9')
        info->vegType = *vegType;
    else {
        info->vegType = getTypeCodeByName(vegType) + '0';
    }
    strcpy(info->nutritionFacts, nutritionFacts);
    return info;
}

void freeVegInfo(vegInfo head) {
    vegInfo current = head, prev;
    while (1) {
        if (current->next == NULL){
            free(current);
            break;
        }
        prev = current;
        current = current->next;
        free(prev);
    }
}

searchResult searchVegName(vegInfo veginfo, char *searchData) {
    int resultNum = 0;
    int *result;
    vegInfo current = veginfo;
    do {
        if(strstr(current->vegName, searchData) != NULL) {
            if (resultNum == 0){
```



```
        result = (int *)malloc(sizeof(int));
        resultNum = 1;
        result[0] = current->vegCode;
    }
    else {
        resultNum++;
        result = realloc(result, resultNum * sizeof(int));
        result[resultNum - 1] = current->vegCode;
    }
}
current = current->next;
} while(current != NULL);
if(resultNum == 0) {
    return NULL;
}
else {
    searchResult searchresult =
(searchResult)malloc(sizeof(struct search result));
    searchresult->resultNum = resultNum;
    searchresult->vegCodeArray = result;
    return searchresult;
}
}

searchResult searchVegFacts(vegInfo veginfo, char *searchData) {
    int resultNum = 0;
    int *result;
    vegInfo current = veginfo;
    do {
        if(strstr(current->nutritionFacts, searchData)!=NULL) {
            if (resultNum == 0){
                result = (int *)malloc(sizeof(int));
                resultNum = 1;
                result[0] = current->vegCode;
            }
            else {
                resultNum++;
                result = realloc(result, resultNum * sizeof(int));
                result[resultNum - 1] = current->vegCode;
            }
        }
        current = current->next;
    } while(current != NULL);
    if(resultNum == 0) {
        return NULL;
    }
    else {
        searchResult searchresult =
(searchResult)malloc(sizeof(struct search result));
        searchresult->resultNum = resultNum;
        searchresult->vegCodeArray = result;
        return searchresult;
    }
}

void removeVegInfoItem(vegInfo removeItem) {
```



```
/*删除的是最后一项*/
if(removeItem->next == NULL) {
    removeItem->prev->next = NULL;
    free(removeItem);
    return ;
}
vegInfo current;
current = removeItem->next;
for(int i = removeItem->vegCode; current != NULL; i++, current =
current->next){
    current->vegCode = i;
}
if (removeItem->vegCode == 1) {
    /*删除第一项--直接把第二项复制到第一项*/
    vegInfo next = removeItem->next;
    *removeItem = *next;
    free(next);
}
else {
    removeItem->prev->next = removeItem->next;
    removeItem->next->prev = removeItem->prev;
    free(removeItem);
}
}

void removeVegInfoItems(vegInfo removeItem) {
    if (removeItem->vegCode == 1) {
        /*删除第一项--直接把第二项复制到第一项*/
        vegInfo next = removeItem->next;
        *removeItem = *next;
        free(next);
    }
    else {
        removeItem->prev->next = removeItem->next;
        removeItem->next->prev = removeItem->prev;
        free(removeItem);
    }
}

void renameVegInfoID(vegInfo head) {
    vegInfo current = head;
    for(int i = 1; current != NULL; i++, current = current->next){
        current->vegCode = i;
    }
}
```

./Csource/VegInfo.h

```
//
// Created by 黄涛 on 2017/7/17.
//
```

```
#ifndef CHOMEWORK_VEGINFO_H
#define CHOMEWORK_VEGINFO_H

#include "sys.h"
#include "stdio.h"
#include "stdlib.h"
```



```
#include "MyStringUtil.h"
#include "string.h"
#include "VegTypeEnum.h"
#include "PlantingInfo.h"
#include "ListAll.h"

/*veginfo 数据路径 debug 时用绝对路径*/
#if 0
#define VEG_INFO_PATH "veginfo.info"
#else
#define VEG_INFO_PATH
"C:\\Users\\hunto\\Documents\\CHomework\\database\\veginfo.info"
#endif

/*
 * 蔬菜信息结构体
 * @vegCode:蔬菜编号 @vegName:蔬菜名称 @vegType:蔬菜种类
 @nutritionFacts:营养成分
 */
typedef struct veg info {
    struct type info *father;
    plantInfo pInfoList;
    int vegCode;
    char vegName[20];
    char vegType;
    char nutritionFacts[200];
    struct veg info *prev, *next;
}*vegInfo;

/*
 * 蔬菜信息链表的结构体 用于从文件读取链表返回
 * @head:链表表头 @tail:链表尾部 @itemNum:链表长度
 */
typedef struct veg struct info {
    vegInfo head;
    vegInfo tail;
    int itemNum;
}*vegStructInfo;

/*
 * 搜索结果结构体
 * @vegCode:蔬菜编号 @resultNum:符合条件的结果数
 */
typedef struct search result {
    int resultNum;
    int *vegCodeArray;
}*searchResult;

/*
 * @usage: 通过蔬菜编号获取蔬菜信息结构体
 * @param: head:链表表头 vegCode:查找的蔬菜编号
 * @return: 蔬菜信息结构体
 */
vegInfo getVegInfoByCode(vegInfo head, int vegCode);
```



```
/*
 * @usage: 通过蔬菜名称获取蔬菜信息结构体
 * @param: head:链表表头  vegCode:查找的蔬菜名称
 * @return: 蔬菜信息结构体
 */
vegInfo getVegInfoByName(vegInfo head, char *vegName);

/*
 * @usage: 从文件中读取蔬菜信息存储为链表
 * @return: 链表信息结构体
 */
vegStructInfo loadVegInfoFromFile(void);

/*
 * @usage:将蔬菜信息链表存储到文件中
 */
void saveVegInfoToFile(vegInfo head);

/*
 * @usage: 从文件指针处开始读取一项蔬菜信息 用于loadVegInfoFromFile 中
 * @param: 文件指针
 * @return: 种植信息结构体
 */
vegInfo getItemFromFile(FILE * vegInfoFile);

/*
 * @usage: 添加一项蔬菜信息至链表尾部
 * @params: tail:链表表尾  addItem:要添加的蔬菜信息结构体
 */
void addVegInfoItem(vegInfo tail,vegInfo addItem);

/*
 * @usage: 新建一个蔬菜信息结构体 用于与Qt 界面交互
 * @return: 新建的蔬菜信息结构体
 */
vegInfo createVegInfoItem(int vegCode, char * vegName, char *
vegType, char * nutritionFacts);

/*
 * @usage: 释放链表内存
 * @param: 链表表头
 */
void freeVegInfo(vegInfo head);

/*
 * @usage : 搜索菜名包含关键字的蔬菜信息
 */
searchResult searchVegName(vegInfo veginfo, char *searchData);

/*
 * @usage : 搜索营养成分包含关键字的蔬菜信息
 */
searchResult searchVegFacts(vegInfo veginfo, char *searchData);

/*删除单项蔬菜基本信息*/
void removeVegInfoItem(vegInfo removeItem);
/*删除多项蔬菜基本信息*/
```



```
void removeVegInfoItems(vegInfo removeItem);
/*删除多项后要重新编号*/
void renameVegInfoID(vegInfo head);

#endif //CHOMEWORK_VEGINFO_H

./Csource/VegInfoListSort.c

//
// Created by hunto on 2017/8/31 0031.
//

#include "VegInfoListSort.h"
#include "VegInfo.h"

void VegInfoListDivide(vegInfo first, vegInfo *left, vegInfo *right);
vegInfo VegInfoListMergeInPlace (vegInfo left, vegInfo right, int
sortType);
int VegInfoCompare (vegInfo a, vegInfo b, int sortType);

vegInfo VegInfoListCopy(vegInfo info) {
    vegInfo current, copy = NULL, prev;
    vegInfo result = (vegInfo)malloc(sizeof(struct veg info));
    *result = *info;
    prev = result;
    current = info->next;
    while (current != NULL) {
        copy = (vegInfo)malloc(sizeof(struct veg info));
        *copy = *current;
        prev->next = copy;
        prev = copy;
        current = current->next;
    }
    copy->next = NULL;
    return result;
}

vegInfo VegInfoListMergeSort(vegInfo first, int sortType) {
    vegInfo left, right;
    if (first == NULL || first->next == NULL) {
        return first;
    }
    VegInfoListDivide(first, &left, &right);
    left = VegInfoListMergeSort(left, sortType);
    right = VegInfoListMergeSort(right, sortType);
    return VegInfoListMergeInPlace(left, right, sortType);
}

void VegInfoListDivide(vegInfo first, vegInfo *left, vegInfo *right)
{
    vegInfo slow = first, fast = first->next;
    while (fast != NULL) {
        fast = fast->next;
        if (fast != NULL) {
            fast = fast->next;
            slow = slow->next;
        }
    }
    *left = first;
    *right = slow->next;
    slow->next = NULL;
}
```



```
    }
}
*left = first;
*right = slow->next;
slow->next = NULL;
}

vegInfo VegInfoListMergeInPlace (vegInfo left, vegInfo right, int
sortType) {
    vegInfo tmp head = (vegInfo)malloc(sizeof(struct veg info));
    vegInfo first = tmp head;
    for (; left != NULL && right != NULL; first = first->next) {
        if (VegInfoCompare(left, right, sortType)) {
            first->next = left;
            left = left->next;
        }
        else {
            first->next = right;
            right = right->next;
        }
    }
    first->next = left != NULL ? left : right;
    first = tmp head->next;
    free(tmp head);
    return first;
#ifdef 0
    /*方法2*/
    if (left == NULL) {
        return right;
    }
    if (right == NULL) {
        return left;
    }
    vegInfo first;
    if (left->area <= right->area) {
        first = left;
        first->next = vegInfoListMergeInPlaceByArea(left->next,
right);
    }
    else {
        first = right;
        first->next = vegInfoListMergeInPlaceByArea(left,
right->next);
    }
    return first;
#endif
}

int VegInfoCompare (vegInfo a, vegInfo b, int sortType) {
    switch (sortType) {
        case VEGSORTDEFAULT:
            if (a->vegCode <= b->vegCode) {
                return 1;
            }
            else return 0;
        case VEGSORTTYPE:
            if (a->vegType <= b->vegType) {
```





```
        return 1;
    }
    else return 0;
    case VEGSORTFACTSLENTH:
        if (strlen(a->nutritionFacts) <=
strlen(b->nutritionFacts)) {
            return 0;
        }
        else return 1;
    }
    return 0;
}
```

### ./Csource/VegInfoListSort.h

```
//
// Created by hunto on 2017/8/31 0031.
//

#ifndef CSOURCE_VEGINFOLISTSORT_H
#define CSOURCE_VEGINFOLISTSORT_H

#include "VegInfo.h"

/*
 * 蔬菜基本信息的归并排序
 * 基本等同蔬菜种植信息的归并排序
 * 本处不再赘述
 */

enum veg sort type {
    VEGSORTTYPE, VEGSORTDEFAULT, VEGSORTFACTSLENTH
};

vegInfo VegInfoListCopy(vegInfo info);
vegInfo VegInfoListMergeSort(vegInfo first, int sortType);

#endif //CSOURCE_VEGINFOLISTSORT_H
```

### ./CppSource/add\_plantinfo\_dialog.cpp

```
#include "add_plantinfo_dialog.h"
#include "ui_add_plantinfo_dialog.h"
#include "Csource/cppincludes.h"

AddPlantInfoDialog::AddPlantInfoDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::AddPlantInfoDialog)
```



```
{
    ui->setupUi(this);
}

AddPlantInfoDialog::~AddPlantInfoDialog()
{
    delete ui;
}

/*选择 OK */
void AddPlantInfoDialog::on_buttonBox_accepted()
{
    /*读取蔬菜种植信息单链表*/
    plantStructInfo plantStruct = loadPlantInfoFromFile();
    /*将输入框内容发至核心层处理*/
    addPlantInfoItem(plantStruct->tail,
                     createPlantInfoItem(plantStruct->tail->id + 1,

ui->textVegCode->text().toLatin1().data(),
ui->textArea->text().toLatin1().data(),

ui->textWeight->text().toLatin1().data(),
ui->textYear->text().toLatin1().data()));
    /*保存修改后的链表*/
    savePlantInfoToFile(plantStruct->head);
    /*释放内存*/
    freePlantInfo(plantStruct->head);
    free(plantStruct);
    /*发送刷新表格信号 signal*/
    emit sendBoolean(true);
}
```

#### **./CppSource/ add\_plantinfo\_dialog.h**

```
#ifndef ADDPLANTINFODIALOG_H
#define ADDPLANTINFODIALOG_H

#include <QDialog>

/*新增蔬菜种植信息窗口*/

namespace Ui {
class AddPlantInfoDialog;
}

class AddPlantInfoDialog : public QDialog
{
    Q_OBJECT

public:
    explicit AddPlantInfoDialog(QWidget *parent = 0);
    ~AddPlantInfoDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool isChanged);
private slots:
    /*OK 按钮点击槽函数*/
    void on_buttonBox_accepted();

private:
```



```
    Ui::AddPlantInfoDialog *ui;  
};
```

```
#endif // ADDPLANTINFODIALOG_H
```

### ./CppSource/add\_type\_dialog.cpp

```
#include "add_type_dialog.h"  
#include "ui_add_type_dialog.h"  
#include "Csource/cppincludes.h"  
  
AddTypeDialog::AddTypeDialog(QWidget *parent) :  
    QDialog(parent),  
    ui(new Ui::AddTypeDialog)  
{  
    ui->setupUi(this);  
}  
  
AddTypeDialog::~AddTypeDialog()  
{  
    delete ui;  
}  
  
void AddTypeDialog::on_buttonBox_accepted()  
{  
    /*从文件读取蔬菜种类信息*/  
    typeInfo tInfo = loadFarmListFromFile();  
    /*增加蔬菜种类信息到链表*/  
    addTypeInfoItem(tInfo, ui->lineEdit->text().toUtf8().data());  
    /*保存蔬菜种类信息到文件*/  
    saveTypeInfoToFile(tInfo);  
    /*发送刷新表格信号量 signal*/  
    emit sendBoolean(true);  
}
```

### ./CppSource/add\_type\_dialog.h

```
#ifndef ADD_TYPE_DIALOG_H  
#define ADD_TYPE_DIALOG_H  
  
#include <QDialog>  
  
/*新增蔬菜种类信息窗口*/  
  
namespace Ui {  
class AddTypeDialog;  
}  
  
class AddTypeDialog : public QDialog  
{  
    Q_OBJECT  
  
public:  
    explicit AddTypeDialog(QWidget *parent = 0);  
    ~AddTypeDialog();  
signals:  
    /*信号量函数，用于发送 boolean 给槽函数*/  
    void sendBoolean(bool isChanged);  
private slots:  
    /*点击 OK 按钮槽函数*/
```



```
void on_buttonBox_accepted();

private:
    Ui::AddTypeDialog *ui;
};

#endif // ADD_TYPE_DIALOG_H

./CppSource/add_vegtype_dialog.cpp

#include "add_vegtype_dialog.h"
#include "ui_add_vegtype_dialog.h"

AddVegTypeDialog::AddVegTypeDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::AddVegTypeDialog)
{
    ui->setupUi(this);
}

AddVegTypeDialog::~AddVegTypeDialog()
{
    delete ui;
}

void AddVegTypeDialog::on_buttonBox_accepted()
{
    /*从文件中读取蔬菜基本信息*/
    vegStructInfo info = loadVegInfoFromFile();
    /*新增蔬菜基本信息至链表*/
    addVegInfoItem(info->tail,
                   createVegInfoItem(info->tail->vegCode + 1,

ui->vegName->text().toUtf8().data(),

ui->vegType->text().toUtf8().data(),

ui->vegN->toPlainText().toUtf8().data()));
    /*保存蔬菜基本信息至文件*/
    saveVegInfoToFile(info->head);
    /*发送刷新表格信号 signal*/
    emit sendBoolean(true);
    /*释放内存*/
    freeVegInfo(info->head);
}
```

### ./CppSource/add\_vegtype\_dialog.h

```
#ifndef ADDVEGTYPEDIALOG_H
#define ADDVEGTYPEDIALOG_H

#include <QDialog>
#include "Csource/cppincludes.h"

/*新增蔬菜基本信息窗口*/

namespace Ui {
class AddVegTypeDialog;
}
```



```
class AddVegTypeDialog : public QDialog
{
    Q_OBJECT

public:
    explicit AddVegTypeDialog(QWidget *parent = 0);
    ~AddVegTypeDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool isChanged);
private slots:
    /*OK 按钮点击槽函数*/
    void on_buttonBox_accepted();

private:
    Ui::AddVegTypeDialog *ui;
};

#endif // ADDVEGTYPEDIALOG_H
```

### **./CppSource/delete\_plantinfo\_dialog.cpp**

```
#include "delete_plantinfo_dialog.h"
#include "ui_delete_plantinfo_dialog.h"
#include "notfound_dialog.h"
#include "Csource/cppincludes.h"

DeletePlantInfoDialog::DeletePlantInfoDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DeletePlantInfoDialog)
{
    ui->setupUi(this);
}

DeletePlantInfoDialog::~DeletePlantInfoDialog()
{
    delete ui;
}

void DeletePlantInfoDialog::on_buttonBox_accepted()
{
    QString str = ui->lineEdit->text();
    /*读取蔬菜种植信息*/
    plantStructInfo info = loadPlantInfoFromFile();
    /*删除多项*/
    if (str.contains(',')){ //如果是删除多项
        for(int i = 0; i < str.count(',') + 1; i++) { //一项项删除
            plantInfo plantinfo = getPlantInfoByID(info->head,
str.section(',', i, i).toInt());
            if (plantinfo == NULL){
                NotFoundDialog *dlg = new NotFoundDialog(this);
                dlg->show();
            }
            else
                removePlantInfoItems(plantinfo);
        }
        /*删除完之后把编号重新编一次*/
        renamePlantInfoID(info->head);
    }
}
```



```
/*删除一项*/
else {
    plantInfo plantinfo = getPlantInfoByID(info->head,
str.toInt());
    if (plantinfo == NULL){
        NotFoundDialog *dlg = new NotFoundDialog(this);
        dlg->show();
    }
    else
        removePlantInfoItem(plantinfo);
}
/*保存到文件*/
savePlantInfoToFile(info->head);
/*发送刷新表格信号 signal*/
emit sendBoolean(true);
/*释放内存*/
freePlantInfo(info->head);
}
```

### **./CppSource/delete\_plantinfo\_dialog.h**

```
#ifndef DELETE_PLANTINFO_DIALOG_H
#define DELETE_PLANTINFO_DIALOG_H

#include <QDialog>
#include "Csource/cppincludes.h"

/*删除蔬菜种植信息窗口*/

namespace Ui {
class DeletePlantInfoDialog;
}

class DeletePlantInfoDialog : public QDialog
{
    Q_OBJECT

public:
    explicit DeletePlantInfoDialog(QWidget *parent = 0);
    ~DeletePlantInfoDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool isChanged);
private slots:
    /*点击 OK 槽函数*/
    void on_buttonBox_accepted();

private:
    Ui::DeletePlantInfoDialog *ui;
};

#endif // DELETE_PLANTINFO_DIALOG_H
```

### **./CppSource/delete\_type\_dialog.cpp**

```
#include "delete_type_dialog.h"
#include "ui_delete_type_dialog.h"
#include "Csource/cppincludes.h"
#include "notfound_dialog.h"
```



```
DeleteTypeDialog::DeleteTypeDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DeleteTypeDialog)
{
    ui->setupUi(this);
}

DeleteTypeDialog::~DeleteTypeDialog()
{
    delete ui;
}

void DeleteTypeDialog::on_buttonBox_accepted()
{
    /*从文件中读取蔬菜信息十字交叉链表*/
    typeInfo tInfo = loadFarmListFromFile();
    /*删除并判断返回值*/
    if (deleteTypeInfo(tInfo,
ui->lineEdit->text().toUtf8().data()[0])) {
        /*成功 -- 保存文件并刷新*/
        saveTypeInfoToFile(tInfo);
        emit sendBoolean(true);
    }
    else {
        /*失败 -- 只能说是没有这个信息 -- 显示未找到窗口*/
        NotFoundDialog *dlg = new NotFoundDialog(this);
        dlg->show();
    }
}
```

### **./CppSource/delete\_type\_dialog.h**

```
#ifndef DELETE_TYPE_DIALOG_H
#define DELETE_TYPE_DIALOG_H

#include <QDialog>

/*删除蔬菜种类信息窗口*/

namespace Ui {
class DeleteTypeDialog;
}

class DeleteTypeDialog : public QDialog
{
    Q_OBJECT

public:
    explicit DeleteTypeDialog(QWidget *parent = 0);
    ~DeleteTypeDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean (bool isChanged);
private slots:
    /*点击 OK 槽函数*/
    void on_buttonBox_accepted();

private:
    Ui::DeleteTypeDialog *ui;
};
```



```
#endif // DELETE_TYPE_DIALOG_H
```

### **./CppSource/delete\_veginfo\_dialog.cpp**

```
#include "delete_veginfo_dialog.h"
#include "ui_delete_veginfo_dialog.h"
#include "Csource/cppincludes.h"
#include "notfound_dialog.h"

DeleteVegInfoDialog::DeleteVegInfoDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::DeleteVegInfoDialog)
{
    ui->setupUi(this);
}

DeleteVegInfoDialog::~DeleteVegInfoDialog()
{
    delete ui;
}

void DeleteVegInfoDialog::on_buttonBox_accepted()
{
    QString str = ui->lineEdit->text();
    vegStructInfo info = loadVegInfoFromFile();
    //vegInfo current = info->head;
    /*删除多项*/
    if (str.contains(',')) {
        for(int i = 0; i < str.count(',') + 1; i++) {
            vegInfo veginfo = getVegInfoByCode(info->head,
str.section(',', i, i).toInt());
            if (veginfo == NULL) {
                NotFoundDialog *dlg = new NotFoundDialog(this);
                dlg->show();
            }
            else
                removeVegInfoItems(veginfo);
        }
        renameVegInfoID(info->head);
    }
    /*删除一项*/
    else {
        vegInfo veginfo = getVegInfoByCode(info->head, str.toInt());
        if (veginfo == NULL) {
            NotFoundDialog *dlg = new NotFoundDialog(this);
            dlg->show();
        }
        else
            removeVegInfoItem(veginfo);
    }
    saveVegInfoToFile(info->head);
    emit sendBoolean(true);
    freeVegInfo(info->head);
}
```

### **./CppSource/delete\_veginfo\_dialog.h**

```
#ifndef DELETE_VEGINFO_DIALOG_H
#define DELETE_VEGINFO_DIALOG_H
```





```
#include <QDialog>

/*删除蔬菜基本信息窗口---注释同前面的删除*/

namespace Ui {
class DeleteVegInfoDialog;
}

class DeleteVegInfoDialog : public QDialog
{
    Q_OBJECT

public:
    explicit DeleteVegInfoDialog(QWidget *parent = 0);
    ~DeleteVegInfoDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    sendBoolean(bool);
private slots:
    void on_buttonBox_accepted();

private:
    Ui::DeleteVegInfoDialog *ui;
};

#endif // DELETE_VEGINFO_DIALOG_H
```

### **./CppSource/edit\_plantinfo\_dialog.cpp**

```
#include "edit_plantinfo_dialog.h"
#include "ui_edit_plantinfo_dialog.h"
#include "Csource/cppincludes.h"
#include "notfound_dialog.h"
#include "mainwindow.h"

EditPlantInfoDialog::EditPlantInfoDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::EditPlantInfoDialog)
{
    ui->setupUi(this);
}

EditPlantInfoDialog::~EditPlantInfoDialog()
{
    delete ui;
}

void EditPlantInfoDialog::on_buttonBox_accepted()
{
    /*读取种植信息*/
    plantStructInfo plantSInfo = loadPlantInfoFromFile();
    /*获取到符合的种植信息*/
    plantInfo info =
    getPlantInfoByVegCode(plantSInfo->head ,ui->vegCode->text().toInt());
    /*如果存在，一个个修改*/
    if (info != NULL){
        if (!ui->weight->text().isEmpty()){
            info->weight = ui->weight->text().toFloat();
        }
    }
}
```



```
        if (!ui->area->text().isEmpty()) {
            info->area = ui->area->text().toInt();
        }
        if (!ui->year->text().isEmpty()) {
            strcpy(info->year, ui->year->text().toUtf8().data());
        }
        /*保存到文件*/
        savePlantInfoToFile(plantSInfo->head);
        /*刷新表格*/
        emit sendBoolean(true);
    }
    else {
        /*不存在弹出未找到窗口*/
        NotFoundDialog *dialog = new NotFoundDialog(this);
        dialog->show();
    }
    /*释放内存*/
    freePlantInfo(plantSInfo->head);
}
```

### **./CppSource/edit\_plantinfo\_dialog.h**

```
#ifndef EDITPLANTINFODIALOG_H
#define EDITPLANTINFODIALOG_H

#include <QDialog>
#include "mainwindow.h"

/*修改蔬菜种植信息窗口*/

namespace Ui {
class EditPlantInfoDialog;
}

class EditPlantInfoDialog : public QDialog
{
    Q_OBJECT

public:
    explicit EditPlantInfoDialog(QWidget *parent = 0);
    ~EditPlantInfoDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool isChanged);
private slots:
    /*点击 OK 槽函数*/
    void on_buttonBox_accepted();

private:
    Ui::EditPlantInfoDialog *ui;
};

#endif // EDITPLANTINFODIALOG_H
```

### **./CppSource/edit\_type\_dialog.cpp**

```
#include "edit_type_dialog.h"
#include "ui_edit_type_dialog.h"
```



```
#include "Csource/cppincludes.h"
#include "notfound_dialog.h"

EditTypeDialog::EditTypeDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::EditTypeDialog)
{
    ui->setupUi(this);
}

EditTypeDialog::~EditTypeDialog()
{
    delete ui;
}

void EditTypeDialog::on_buttonBox_accepted()
{
    typeInfo tInfo = loadFarmListFromFile();
    if (editTypeInfo(tInfo, ui->lineEdit->text().toUtf8().data()[0],
ui->lineEdit_2->text().toUtf8().data())) {
        /*修改成功--保存--刷新*/
        saveTypeInfoToFile(tInfo);
        emit sendBoolean(true);
    }
    else {
        /*没找到*/
        NotFoundDialog *dlg = new NotFoundDialog(this);
        dlg->show();
    }
}
```

### **./CppSource/edit\_type\_dialog.h**

```
#ifndef EDIT_TYPE_DIALOG_H
#define EDIT_TYPE_DIALOG_H

#include <QDialog>

/*修改蔬菜种类信息窗口*/

namespace Ui {
class EditTypeDialog;
}

class EditTypeDialog : public QDialog
{
    Q_OBJECT

public:
    explicit EditTypeDialog(QWidget *parent = 0);
    ~EditTypeDialog();

private slots:
    /*点击 OK 槽*/
    void on_buttonBox_accepted();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool);
private:
```



```
    Ui::EditTypeDialog *ui;
};

#endif // EDIT_TYPE_DIALOG_H
```

### **./CppSource/edit\_veginfo\_dialog.cpp**

```
#include "edit_veginfo_dialog.h"
#include "ui_edit_veginfo_dialog.h"
#include "Csource/cppincludes.h"
#include "notfound_dialog.h"

EditVegInfoDialog::EditVegInfoDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::EditVegInfoDialog)
{
    ui->setupUi(this);
}

EditVegInfoDialog::~EditVegInfoDialog()
{
    delete ui;
}

void EditVegInfoDialog::on_buttonBox_accepted()
{
    /*读取蔬菜基本信息*/
    vegStructInfo vegSInfo = loadVegInfoFromFile();
    /*找到对应基本信息结构体*/
    vegInfo info =
getVegInfoByCode(vegSInfo->head, ui->vegCode->text().toInt());
    if (info != NULL) {
        /*存在--一个个修改*/
        if (!ui->vegType->text().isEmpty()) {
            info->vegType = ui->vegType->text().toInt() + '0';
        }
        if (!ui->textEdit->toPlainText().isEmpty()) {
            strcpy(info->nutritionFacts,
ui->textEdit->toPlainText().toUtf8().data());
        }
        /*保存文件*/
        saveVegInfoToFile(vegSInfo->head);
        /*刷新表格*/
        emit sendBoolean(true);
    }
    else {
        /*不存在--显示未找到窗口*/
        NotFoundDialog *dialog = new NotFoundDialog(this);
        dialog->show();
    }
    /*释放内存*/
    freeVegInfo(vegSInfo->head);
}
```

**./CppSource/edit\_veginfo\_dialog.h**

```
#ifndef EDITVEGINFODIALOG_H
#define EDITVEGINFODIALOG_H

#include <QDialog>

/*修改蔬菜基本信息窗口*/

namespace Ui {
class EditVegInfoDialog;
}

class EditVegInfoDialog : public QDialog
{
    Q_OBJECT

public:
    explicit EditVegInfoDialog(QWidget *parent = 0);
    ~EditVegInfoDialog();
signals:
    /*信号量函数，用于发送 boolean 给槽函数*/
    void sendBoolean(bool isChanged);
private slots:
    /*点击 OK 槽函数*/
    void on_buttonBox_accepted();

private:
    Ui::EditVegInfoDialog *ui;
};

#endif // EDITVEGINFODIALOG_H
```

**./CppSource/imformation\_dialog.cpp**

```
#include "imformation_dialog.h"
#include "ui_imformation_dialog.h"

InformationDialog::InformationDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::ImformationDialog)
{
    ui->setupUi(this);
}

void InformationDialog::prepareShow() {
    /*显示之前调用，只是修改 label 显示文本*/
    ui->label->setText(imformation);
}

InformationDialog::~ImformationDialog()
{
    delete ui;
}
```

**./CppSource/imformation\_dialog.h**

```
#ifndef INFORMATIONDIALOG_H
#define INFORMATIONDIALOG_H
```



```
#include <QDialog>

/*显示信息窗口*/

namespace Ui {
class ImformationDialog;
}

class ImformationDialog : public QDialog
{
    Q_OBJECT

public:
    explicit ImformationDialog(QWidget *parent = 0);
    QString information;
    ~ImformationDialog();
    /*show 前调用, 准备显示信息内容*/
    void prepareShow(void);
private:
    Ui::ImformationDialog *ui;
};

#endif // INFORMATIONDIALOG_H
```

### **./CppSource/main.cpp**

```
#include "mainwindow.h"
#include <QApplication>
#include "Csource/cppincludes.h"

/*
 * 形式上的 main 函数
 * 启动 main 窗口就没用了
 */

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}
```

### **./CppSource/mainwindow.cpp**

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "add_plantinfo_dialog.h"
#include "add_vegtype_dialog.h"
#include "edit_plantinfo_dialog.h"
#include "yearinput_dialog.h"
#include "imformation_dialog.h"
#include "Csource/cppincludes.h"
#include "edit_veginfo_dialog.h"
#include "delete_plantinfo_dialog.h"
#include "delete_veginfo_dialog.h"
#include "searchinput_dialog.h"
```



```
#include "add_type_dialog.h"
#include "delete_type_dialog.h"
#include "edit_type_dialog.h"
#include "search2_input_dialog.h"
#include "range_input_dialog.h"
#include "vegcode_input_dialog.h"
#include "QFileDialog"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    //plantSInfo = loadPlantInfoFromFile();
    //vegSInfo = loadVegInfoFromFile();
    tInfo = loadFarmListFromFile();
    vInfo = VegInfoListMergeSort(getVegInfoFromFarmList(tInfo),
    VEGSORTDEFAULT);
    pInfo =
    PlantInfoListMergeSort(getPlantInfoListFromFarmList(tInfo),
    SORTDEFAULT);
    showPlantInfo(pInfo);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_actionAddPlantInfo_triggered()
{
    AddPlantInfoDialog *dlg = new AddPlantInfoDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
    SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_action_Quit_triggered()
{
    exit(0);
}

void MainWindow::on_actionAddVegType_triggered()
{
    AddVegTypeDialog *dlg = new AddVegTypeDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
    SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actionPlantInfoAll_triggered()
{
    showPlantInfo(pInfo);
}

void MainWindow::showPlantInfo(plantInfo head)
{
    int i;
    currentTable = PLANTALL;
```



```
QStandardItemModel *model = new QStandardItemModel();
model->setColumnCount(7);
model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("种植编号"));
model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜编号"));
model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜种类"));
model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("蔬菜名称"));
model->setHeaderData(4, Qt::Horizontal, QString::fromUtf8("种植面积/分"));
model->setHeaderData(5, Qt::Horizontal, QString::fromUtf8("种植年份"));
model->setHeaderData(6, Qt::Horizontal, QString::fromUtf8("收成/kg"));
ui->tableView->setModel(model);
plantInfo current = head;
for (i = 0; current != NULL; i++, current = current->next) {
    model->setItem(i, 0, new
QStandardItem(QString::number(current->id, 10)));
    model->setItem(i, 1, new
QStandardItem(QString::number(current->vegCode, 10)));
    model->setItem(i, 2,
        new
QStandardItem(QString(current->father->father->typeName)));
    model->setItem(i, 3,
        new
QStandardItem(QString(current->father->vegName)));
    model->setItem(i, 4, new
QStandardItem(QString::number(current->area, 10)));
    model->setItem(i, 5, new QStandardItem(current->year));
    model->setItem(i, 6, new
QStandardItem(QString("%1").arg(current->weight)));
    for (int j = 0; j < 7; j++){
        model->item(i, j)->setTextAlignment(Qt::AlignCenter);
    }
}
ui->statusBar->showMessage(QString("菜园种植数据读取完成, 当前种植数目:%1").arg(i), 10000);
}

void MainWindow::showVegInfo(vegInfo info)
{
    int i;
    currentTable = VEGALL;
    QStandardItemModel *model = new QStandardItemModel();
    model->setColumnCount(4);
    model->setHeaderData(0, Qt::Horizontal, QString("蔬菜编号"));
    model->setHeaderData(1, Qt::Horizontal, QString("蔬菜种类"));
    model->setHeaderData(2, Qt::Horizontal, QString("蔬菜名称"));
    model->setHeaderData(3, Qt::Horizontal, QString("营养成分"));
    ui->tableView->setModel(model);
    vegInfo current = info;
    for (i = 0; current != NULL; i++, current = current->next) {
        model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->vegCode)));
        model->setItem(i, 1, new
```





```
QStandardItem(QString(current->father->typeName)));
    model->setItem(i, 2,
        new QStandardItem(QString(current->vegName)));
    model->setItem(i, 3, new
QStandardItem(QString(current->nutritionFacts)));
    for (int j = 0; j < 4; j++){
        model->item(i, j)->setTextAlignment(Qt::AlignCenter);
    }
}
ui->statusBar->showMessage(QString("蔬菜信息数据读取完成, 当前存储蔬菜
数目:%1").arg(i),10000);
}

void MainWindow::showTypeInfo(typeInfo info) {
    int i;
    currentTable = TYPEALL;
    QStandardItemModel *model = new QStandardItemModel();
    model->setColumnCount(3);
    model->setHeaderData(0, Qt::Horizontal, QString("种类编号"));
    model->setHeaderData(1, Qt::Horizontal, QString("种类名称"));
    model->setHeaderData(2, Qt::Horizontal, QString("蔬菜"));
    ui->tableView->setModel(model);
    typeInfo current = info;
    for (i = 0; current != NULL; i++, current = current->next) {
        model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->typeCode)));
        model->setItem(i, 1, new
QStandardItem(QString(current->typeName)));
        vegInfo vInfo = current->vInfoList;
        QString vegAll;
        if (vInfo != NULL) {
            vegAll = QString("%1").arg(vInfo->vegName);
            vInfo = vInfo->next;
            for (; vInfo != NULL; vInfo = vInfo->next) {
                vegAll += QString("、%1").arg(vInfo->vegName);
            }
        }
        else {
            vegAll = QString("[该种类下没有蔬菜]");
        }
        model->setItem(i, 2,
            new QStandardItem(vegAll));
        for (int j = 0; j < 3; j++){
            model->item(i, j)->setTextAlignment(Qt::AlignCenter);
        }
    }
    ui->statusBar->showMessage(QString("蔬菜种类数据读取完成, 当前存储种类
数目:%1").arg(i),10000);
}

void MainWindow::on_actionVegTypeAll_triggered()
{
    showVegInfo(vInfo);
}

void MainWindow::on_actionEditPlantInfo_triggered()
{
    EditPlantInfoDialog *dialog = new EditPlantInfoDialog(this);
```



```
connect(dialog, SIGNAL(sendBoolean(bool)), this,
SLOT(receiveBoolean(bool)));
dialog->show();
}

void MainWindow::receiveBoolean(bool isChanged) {
    if (isChanged) {
        //Sleep(1500);
        /*释放内存*/
        freePlantInfo(pInfo);
        freeVegInfo(vInfo);
        /*读取数据*/
        tInfo = loadFarmListFromFile();
        vInfo = VegInfoListMergeSort(getVegInfoFromFarmList(tInfo),
VEGSORTDEFAULT);
        pInfo =
PlantInfoListMergeSort(getPlantInfoListFromFarmList(tInfo),
SORTDEFAULT);
        switch (currentTable) {
            case PLANTALL:
                showPlantInfo(pInfo);
                break;
            case VEGALL:
                showVegInfo(vInfo);
                break;
            case TYPEALL:
                showTypeInfo(tInfo);
                break;
            default:
                showPlantInfo(pInfo);
                break;
        }
    }
}

void MainWindow::receiveYear(int year) {
    switch (currentYearType) {
        case SHOWPLANTINFO:
            showYearPlantInfo(year);
            break;
        case AREA:
            {
                ImformationDialog *dlg = new ImformationDialog(this);
                dlg->imformation = QString("%1 年蔬菜种植面积总和为:  %2 分地
").arg(year).arg(getPlantAreaYear(pInfo, year));
                dlg->prepareShow();
                dlg->show();
                break;
            }
        case WEIGHT:
            {
                ImformationDialog *dlg2 = new ImformationDialog(this);
                dlg2->imformation = QString("%1 年所有蔬菜收成总和为:  %2
kg").arg(year)
                .arg(getPlantWeightYear(pInfo, year));
                dlg2->prepareShow();
                dlg2->show();
                break;
            }
    }
}
```



```
}

void MainWindow::on_actionSaveAll_triggered()
{
    savePlantInfoToFile(pInfo);
    saveVegInfoToFile(vInfo);
}

void MainWindow::on_actionSavePlantInfo_triggered()
{
    savePlantInfoToFile(pInfo);
}

void MainWindow::on_actionSaveVegType_triggered()
{
    saveVegInfoToFile(vInfo);
}

void MainWindow::on_actionThisYear_3_triggered()
{
    showYearPlantInfo(2017);
}

void MainWindow::showYearPlantInfo (int year) {
    int i;
    currentTable = PLANTALL;
    QStandardItemModel *model = new QStandardItemModel();
    model->setColumnCount(7);
    model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("种植编
号"));
    model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜编
号"));
    model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜种
类"));
    model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("蔬菜名
称"));
    model->setHeaderData(4, Qt::Horizontal, QString::fromUtf8("种植面
积/分"));
    model->setHeaderData(5, Qt::Horizontal, QString::fromUtf8("种植年
份"));
    model->setHeaderData(6, Qt::Horizontal, QString::fromUtf8("收成
/kg"));
    ui->tableView->setModel(model);
    plantInfo current = pInfo;
    for (i = 0; current != NULL; i++, current = current->next) {
        while (current != NULL) {
            if (QString::fromUtf8(current->year).toInt() == year)
                break;
            current = current->next;
        }
        if (current == NULL) {
            break;
        }
        model->setItem(i, 0, new
QStandardItem(QString::number(current->id, 10)));
        model->setItem(i, 1, new
QStandardItem(QString::number(current->vegCode, 10)));
        model->setItem(i, 2, new
QStandardItem(QString(current->father->father->typeName)));
    }
}
```



```
        model->setItem(i, 3, new
QStandardItem(QString(current->father->vegName)));
        model->setItem(i, 4, new
QStandardItem(QString::number(current->area, 10)));
        model->setItem(i, 5, new QStandardItem(current->year));
        model->setItem(i, 6, new
QStandardItem(QString("%1").arg(current->weight)));
        for (int j = 0; j < 7; j++){
            model->item(i, j)->setTextAlignment(Qt::AlignCenter);
        }
    }
    ui->statusBar->showMessage(QString::fromUtf8("菜园种植数据读取完成,
该年份种植数目:") + QString("%1").arg(i), 10000);
}

void MainWindow::showTypePlantInfo(int type) {
    int i;
    currentTable = PLANTTYPE;
    QStandardItemModel *model = new QStandardItemModel();
    model->setColumnCount(7);
    model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("种植编
号"));
    model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜编
号"));
    model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜种
类"));
    model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("蔬菜名
称"));
    model->setHeaderData(4, Qt::Horizontal, QString::fromUtf8("种植面
积/分"));
    model->setHeaderData(5, Qt::Horizontal, QString::fromUtf8("种植年
份"));
    model->setHeaderData(6, Qt::Horizontal, QString::fromUtf8("收成
/kg"));
    ui->tableView->setModel(model);
    plantInfo current = pInfo;
    for (i = 0; current != NULL; i++, current = current->next) {
        while (current != NULL) {
            if (getVegInfoByCode(vInfo, current->vegCode)->vegType -
'0' == type)
                break;
            current = current->next;
        }
        if (current == NULL) {
            break;
        }
        model->setItem(i, 0, new
QStandardItem(QString::number(current->id, 10)));
        model->setItem(i, 1, new
QStandardItem(QString::number(current->vegCode, 10)));
        model->setItem(i, 2, new
QStandardItem(QString(current->father->father->typeName));
        model->setItem(i, 3, new
QStandardItem(QString(current->father->vegName));
        model->setItem(i, 4, new
QStandardItem(QString::number(current->area, 10)));
        model->setItem(i, 5, new QStandardItem(current->year));
        model->setItem(i, 6, new
QStandardItem(QString("%1").arg(current->weight)));
    }
```



```
        for (int j = 0; j < 7; j++){
            model->item(i, j)->setTextAlignment(Qt::AlignCenter);
        }
    }
    ui->statusBar->showMessage(QString::fromUtf8("菜园种植数据读取完成,
该种类蔬菜种植数目:") + QString("%1").arg(i), 10000);
}

void MainWindow::on_actionInputYear_3_triggered()
{
    YearInputDialog *dlg = new YearInputDialog(this);
    currentYearType = SHOWPLANTINFO;
    connect(dlg, SIGNAL(sendYear(int)), this,
    SLOT(receiveYear(int)));
    dlg->show();
}

void MainWindow::on_actionAreaAllVeg_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("当前蔬菜种植面积总和为:  %1 分地
").arg(getPlantAreaAll(pInfo));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionInputYear_2_triggered()
{
    YearInputDialog *dlg = new YearInputDialog(this);
    currentYearType = AREA;
    connect(dlg, SIGNAL(sendYear(int)), this,
    SLOT(receiveYear(int)));
    dlg->show();
}

void MainWindow::on_actionThisYear_2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("%1 年蔬菜种植面积总和为:  %2 分地
").arg(2017).arg(getPlantAreaYear(pInfo, 2017));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType1_2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("根茎类蔬菜种植面积总和为:  %1 分地")
        .arg(getPlantAreaVegType(pInfo, vInfo, 1));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType2_2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("果菜类蔬菜种植面积总和为:  %1 分地")
        .arg(getPlantAreaVegType(pInfo, vInfo, 2));
    dlg->prepareShow();
    dlg->show();
}
```



```
}

void MainWindow::on_actionType3_2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("瓜类蔬菜种植面积总和为:  %1 分地")
        .arg(getPlantAreaVegType(pInfo, vInfo, 3));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionTyp34_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("叶菜类蔬菜种植面积总和为:  %1 分地")
        .arg(getPlantAreaVegType(pInfo, vInfo, 4));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType5_2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("菌类蔬菜种植面积总和为:  %1 分地")
        .arg(getPlantAreaVegType(pInfo, vInfo, 5));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionWeightAll_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("所有蔬菜收成总和为:  %1 kg")
        .arg(getPlantWeightAll(pInfo));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionThisYear_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("%1 年所有蔬菜收成总和为:  %2
kg").arg(2017)
        .arg(getPlantWeightYear(pInfo, 2017));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionInputYear_triggered()
{
    YearInputDialog *dlg = new YearInputDialog(this);
    currentYearType = WEIGHT;
    connect(dlg, SIGNAL(sendYear(int)), this,
        SLOT(receiveYear(int)));
    dlg->show();
}

void MainWindow::on_actionType1_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
```



```
        dlg->imformation = QString("根茎类蔬菜收成总和为:  %1 分地")
            .arg(getPlantWeightVegType(pInfo, vInfo, 1));
        dlg->prepareShow();
        dlg->show();
    }

void MainWindow::on_actionType2_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("果菜类蔬菜收成总和为:  %1 分地")
        .arg(getPlantWeightVegType(pInfo, vInfo, 2));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType3_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("瓜类蔬菜收成总和为:  %1 分地")
        .arg(getPlantWeightVegType(pInfo, vInfo, 3));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType4_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("叶菜类蔬菜收成总和为:  %1 分地")
        .arg(getPlantWeightVegType(pInfo, vInfo, 4));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionType5_triggered()
{
    ImformationDialog *dlg = new ImformationDialog(this);
    dlg->imformation = QString("菌类蔬菜收成总和为:  %1 分地")
        .arg(getPlantWeightVegType(pInfo, vInfo, 5));
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::savePlantInfoToFileFromTable() {

}

void MainWindow::on_actionEditVegInfo_triggered()
{
    EditVegInfoDialog *dlg = new EditVegInfoDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
        SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actiondeletePlantInfo_triggered()
{
    DeletePlantInfoDialog *dlg = new DeletePlantInfoDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
        SLOT(receiveBoolean(bool)));
    dlg->show();
}
```



```
void MainWindow::on_actionPIType1_triggered()
{
    showTypePlantInfo(1);
}

void MainWindow::on_actionPIType2_triggered()
{
    showTypePlantInfo(2);
}

void MainWindow::on_actionPIType3_triggered()
{
    showTypePlantInfo(3);
}

void MainWindow::on_actionPIType4_triggered()
{
    showTypePlantInfo(4);
}

void MainWindow::on_actionPIType5_triggered()
{
    showTypePlantInfo(5);
}

void MainWindow::on_actionReloadPlantInfo_triggered()
{
    /*释放内存*/
    freePlantInfo(pInfo);
    /*读取数据*/
    typeInfo tInfo = loadFarmListFromFile();
    pInfo = getPlantInfoListFromFarmList(tInfo);
    pInfo = PlantInfoListMergeSort(pInfo, SORTDEFAULT);
    if (currentTable == PLANTALL){
        showPlantInfo(pInfo);
    }
}

void MainWindow::on_actionReloadVegInfo_triggered()
{
    freeVegInfo(vInfo);
    typeInfo tInfo = loadFarmListFromFile();
    vInfo = getVegInfoFromFarmList(tInfo);
    if (currentTable == VEGALL){
        showVegInfo(vInfo);
    }
}

void MainWindow::on_actionPlantInfoBySearch_triggered()
{
    currentSearchType = VEGNAME;
    SearchInputDialog *dlg = new SearchInputDialog(this);
    connect(dlg, SIGNAL(sendSearch(QString)), this,
    SLOT(receiveSearch(QString)));
    dlg->searchLabel = QString("搜索蔬菜名");
    dlg->prepareShow();
    dlg->show();
}
```





```
}

void MainWindow::receiveSearch(QString data) {
    int i;
    searchResult result = NULL;
    switch (currentSearchType) {
    case VEGNAME:
        result = searchVegName(vInfo, data.toUtf8().data());
        break;
    case VEGFACTS:
        result = searchVegFacts(vInfo, data.toUtf8().data());
        break;
    case VEGINFONAME:
        result = searchVegName(vInfo, data.toUtf8().data());
        break;
    case VEGINFOFACTS:
        currentTable = VEGALL;
        QStandardItemModel *model = new QStandardItemModel();
        model->setColumnCount(4);
        model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("蔬菜编号"));
        model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜种类"));
        model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜名称"));
        model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("营养成分"));
        ui->tableView->setModel(model);
        vegInfo current = vInfo;
        for (i = 0; current != NULL; i++, current = current->next) {
            while (current != NULL) {
                if (strstr(current->nutritionFacts,
data.toUtf8().data()) != NULL)
                    break;
                current = current->next;
            }
            if (current == NULL) {
                break;
            }
            model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->vegCode)));
            model->setItem(i, 1, new
QStandardItem(QString(current->father->typeName)));
            model->setItem(i, 2, new
QStandardItem(QString(current->vegName)));
            model->setItem(i, 3, new
QStandardItem(QString(current->nutritionFacts)));
            for (int j = 0; j < 4; j++){
                model->item(i, j)->setTextAlignment(Qt::AlignCenter);
            }
        }
        ui->statusBar->showMessage(QString("蔬菜品种数据读取完成, 当前存储品种数目:%1").arg(i),10000);
        return ;
    }
    if (result == NULL) {
        ImformationDialog *dlg = new ImformationDialog(this);
        dlg->imformation = QString("无法找到: %1").arg(data);
    }
}
```



```
        dlg->prepareShow();
        dlg->show();
    }
    else {
        if (currentSearchType == VEGINFONAME) {
            currentTable = VEGALL;
            QStandardItemModel *model = new QStandardItemModel();
            model->setColumnCount(4);
            model->setHeaderData(0, Qt::Horizontal,
QString::fromUtf8("蔬菜编号"));
            model->setHeaderData(1, Qt::Horizontal,
QString::fromUtf8("蔬菜种类"));
            model->setHeaderData(2, Qt::Horizontal,
QString::fromUtf8("蔬菜名称"));
            model->setHeaderData(3, Qt::Horizontal,
QString::fromUtf8("营养成分"));
            ui->tableView->setModel(model);
            vegInfo current = vInfo;
            for (i = 0; current != NULL; i++, current =
current->next) {
                int j = 0;
                while(1) {
                    if (current == NULL) {
                        ui->statusBar->showMessage(QString("蔬菜品种数
数据读取完成, 当前存储品种数目:%1").arg(i),10000);
                        return ;
                    }
                    for(j = 0; j < result->resultNum; j++) {
                        if(current->vegCode ==
result->vegCodeArray[j]) {
                            break;
                        }
                    }
                    if (j == result->resultNum) {
                        current = current->next;
                    }
                    else {
                        break;
                    }
                }
                model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->vegCode)));
                model->setItem(i, 1, new
QStandardItem(QString(current->father->typeName)));
                model->setItem(i, 2, new
QStandardItem(QString(current->vegName)));
                model->setItem(i, 3, new
QStandardItem(QString(current->nutritionFacts)));
                for (int j = 0; j < 4; j++){
                    model->item(i,
j)->setTextAlignment(Qt::AlignCenter);
                }
            }
            ui->statusBar->showMessage(QString("蔬菜品种数据读取完成, 当
前存储品种数目:%1").arg(i),10000);
            return ;
        }
        currentTable = PLANTSEARCH;
        QStandardItemModel *model = new QStandardItemModel();
        model->setColumnCount(7);
```



```
        model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("种植编号"));
        model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜编号"));
        model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜种类"));
        model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("蔬菜名称"));
        model->setHeaderData(4, Qt::Horizontal, QString::fromUtf8("种植面积/分"));
        model->setHeaderData(5, Qt::Horizontal, QString::fromUtf8("种植年份"));
        model->setHeaderData(6, Qt::Horizontal, QString::fromUtf8("收成/kg"));
        ui->tableView->setModel(model);
        plantInfo current = pInfo;
        for (i = 0; current != NULL; i++, current = current->next) {
            int j = 0;
            while(1) {
                if (current == NULL) {
                    ui->statusBar->showMessage(QString("菜园种植数据搜索完成, 搜索到的种植数目:") + QString("%1").arg(i), 10000);
                    return ;
                }
                for(j = 0; j < result->resultNum; j++) {
                    if(current->vegCode == result->vegCodeArray[j]) {
                        break;
                    }
                }
                if (j == result->resultNum) {
                    current = current->next;
                }
                else {
                    break;
                }
            }
            model->setItem(i, 0, new
QStandardItem(QString::number(current->id, 10)));
            model->setItem(i, 1, new
QStandardItem(QString::number(current->vegCode, 10)));
            model->setItem(i, 2, new
QStandardItem(QString(current->father->father->typeName)));
            model->setItem(i, 3, new
QStandardItem(QString(current->father->vegName)));
            model->setItem(i, 4, new
QStandardItem(QString::number(current->area, 10)));
            model->setItem(i, 5, new QStandardItem(current->year));
            model->setItem(i, 6, new
QStandardItem(QString("%1").arg(current->weight)));
            for (int j = 0; j < 7; j++){
                model->item(i, j)->setTextAlignment(Qt::AlignCenter);
            }
        }
        ui->statusBar->showMessage(QString("菜园种植数据搜索完成, 搜索到的种植数目:") + QString("%1").arg(i), 10000);
    }
}
```



```
void MainWindow::receiveSearchTwo(searchData data) {
    int i;
    searchResult result = NULL;
    switch (currentSearchType) {
        case PLANTTYPEFACTS:
            result = searchVegFacts(vInfo, data.data2.toUtf8().data());

            break;
        case PLANTYEARFACTS:
            result = searchVegFacts(vInfo, data.data2.toUtf8().data());
            break;
        case VEGTYPEFACTS:
            result = searchVegFacts(vInfo, data.data2.toUtf8().data());
            break;
    }

    if (result == NULL) {
        ImformationDialog *dlg = new ImformationDialog(this);
        dlg->imformation = QString("无法找到: %1").arg(data.data2);
        dlg->prepareShow();
        dlg->show();
        return ;
    }
    if (currentSearchType == VEGTYPEFACTS) {
        currentTable = VEGALL;
        QStandardItemModel *model = new QStandardItemModel();
        model->setColumnCount(4);
        model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("蔬菜编号"));
        model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜种类"));
        model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜名称"));
        model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("营养成分"));
        ui->tableView->setModel(model);
        vegInfo current = vInfo;
        for (i = 0; current != NULL; i++, current = current->next) {
            int j = 0;
            while(1) {
                if (current == NULL) {
                    ui->statusBar->showMessage(QString("蔬菜信息搜索完成, 搜索到的蔬菜数目:") + QString("%1").arg(i), 10000);
                    return ;
                }
                for(j = 0; j < result->resultNum; j++) {
                    if(current->vegCode == result->vegCodeArray[j]
                        && current->vegType ==
data.data1.toUtf8().data()[0]) {
                        break;
                    }
                }
                if (j == result->resultNum) {
                    current = current->next;
                }
                else {
                    break;
                }
            }
        }
    }
}
```



```

        }
    }
    model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->vegCode)));
    model->setItem(i, 1, new
QStandardItem(QString(current->father->typeName)));
    model->setItem(i, 2, new
QStandardItem(QString(current->vegName)));
    model->setItem(i, 3, new
QStandardItem(QString(current->nutritionFacts)));
    for (int j = 0; j < 4; j++){
        model->item(i, j)->setTextAlignment(Qt::AlignCenter);
    }
}
ui->statusBar->showMessage(QString("蔬菜品种数据读取完成, 当前存储
品种数目:%1").arg(i),10000);
return ;
}
currentTable = PLANTSEARCH;
QStandardItemModel *model = new QStandardItemModel();
model->setColumnCount(7);
model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("种植编
号"));
model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜编
号"));
model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜种
类"));
model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("蔬菜名
称"));
model->setHeaderData(4, Qt::Horizontal, QString::fromUtf8("种植面
积/分"));
model->setHeaderData(5, Qt::Horizontal, QString::fromUtf8("种植年
份"));
model->setHeaderData(6, Qt::Horizontal, QString::fromUtf8("收成
/kg"));
ui->tableView->setModel(model);
plantInfo current = pInfo;
for (i = 0; current != NULL; i++, current = current->next) {
    int j = 0;
    while(1) {
        if (current == NULL) {
            ui->statusBar->showMessage(QString("菜园种植数据搜索完
成, 搜索到的种植数目:") + QString("%1").arg(i),10000);
            return ;
        }
        for(j = 0; j < result->resultNum; j++) {
            if(current->vegCode == result->vegCodeArray[j]) {
                if (currentSearchType == PLANTTYPEFACTS) {
                    if (current->father->vegType ==
data.data1.toUtf8().data()[0])
                        break;
                }
                else if (currentSearchType == PLANTYEARFACTS) {
                    if (strstr(current->year,
data.data1.toUtf8().data()) != NULL)
                        break;
                }
            }
        }
    }
}
}

```



```
        }
        if (j == result->resultNum) {
            current = current->next;
        }
        else {
            break;
        }
    }
    model->setItem(i, 0, new
QStandardItem(QString::number(current->id, 10)));
    model->setItem(i, 1, new
QStandardItem(QString::number(current->vegCode, 10)));
    model->setItem(i, 2, new
QStandardItem(QString(current->father->father->typeName)));
    model->setItem(i, 3, new
QStandardItem(QString(current->father->vegName)));
    model->setItem(i, 4, new
QStandardItem(QString::number(current->area, 10)));
    model->setItem(i, 5, new QStandardItem(current->year));
    model->setItem(i, 6, new
QStandardItem(QString("%1").arg(current->weight)));
    for (int j = 0; j < 7; j++){
        model->item(i, j)->setTextAlignment(Qt::AlignCenter);
    }
}
ui->statusBar->showMessage(QString("菜园种植数据搜索完成, 搜索到的种植
数目:") + QString("%1").arg(i), 10000);

}

void MainWindow::on_actionSearchFacts_triggered()
{
    currentSearchType = VEGFACTS;
    SearchInputDialog *dlg = new SearchInputDialog(this);
    connect(dlg, SIGNAL(sendSearch(QString)), this,
SLOT(receiveSearch(QString)));
    dlg->searchLabel = QString("搜索蔬菜营养成分");
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionSearchFactsVeginfo_triggered()
{
    currentSearchType = VEGINFOFACTS;
    SearchInputDialog *dlg = new SearchInputDialog(this);
    connect(dlg, SIGNAL(sendSearch(QString)), this,
SLOT(receiveSearch(QString)));
    dlg->searchLabel = QString("搜索蔬菜营养成分\n 成分间用, 隔开");
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::showTypeVegInfo(int type) {
    int i;
    currentTable = VEGALL;
```



```
QStandardItemModel *model = new QStandardItemModel();
model->setColumnCount(4);
model->setHeaderData(0, Qt::Horizontal, QString::fromUtf8("蔬菜编
号"));
model->setHeaderData(1, Qt::Horizontal, QString::fromUtf8("蔬菜种
类"));
model->setHeaderData(2, Qt::Horizontal, QString::fromUtf8("蔬菜名
称"));
model->setHeaderData(3, Qt::Horizontal, QString::fromUtf8("营养成
分"));
ui->tableView->setModel(model);
vegInfo current = vInfo;
for (i = 0; current != NULL; i++, current = current->next) {
    while (current != NULL) {
        if (current->vegType - '0' == type)
            break;
        current = current->next;
    }
    if (current == NULL) {
        break;
    }
    model->setItem(i, 0, new
QStandardItem(QString("%1").arg(current->vegCode)));
    model->setItem(i, 1, new
QStandardItem(QString(current->father->typeName)));
    model->setItem(i, 2, new
QStandardItem(QString(current->vegName)));
    model->setItem(i, 3, new
QStandardItem(QString(current->nutritionFacts)));
    for (int j = 0; j < 4; j++){
        model->item(i, j)->setTextAlignment(Qt::AlignCenter);
    }
}
ui->statusBar->showMessage(QString::fromUtf8("蔬菜品种数据读取完成,
当前存储品种数目:") + QString("%1").arg(i), 10000);
}

void MainWindow::on_actionShowVeginfoType1_triggered()
{
    showTypeVegInfo(1);
}

void MainWindow::on_actionShowVeginfoType2_triggered()
{
    showTypeVegInfo(2);
}

void MainWindow::on_actionShowVeginfoType3_triggered()
{
    showTypeVegInfo(3);
}

void MainWindow::on_actionShowVeginfoType2_3_triggered()
{
    showTypeVegInfo(4);
}

void MainWindow::on_actionShowVeginfoType5_triggered()
{

```



```
        showTypeVegInfo(5);
    }

void MainWindow::on_actiondeleteVegInfo_triggered()
{
    DeleteVegInfoDialog *dlg = new DeleteVegInfoDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
        SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actionSortBySArea_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTAREA);
    showPlantInfo(head);
}

void MainWindow::on_actionSortByYear_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTYEAR);
    showPlantInfo(head);
}

void MainWindow::on_actionSortByWeight_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTWEIGHT);
    showPlantInfo(head);
}

void MainWindow::on_actionSortByType_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTTYPE);
    showPlantInfo(head);
}

void MainWindow::on_actionSortByAreaR_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTAREAR);
    showPlantInfo(head);
}

void MainWindow::on_actionSortByYearR_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTYEARR);
    showPlantInfo(head);
}

void MainWindow::on_actionSortWeightR_triggered()
{
    plantInfo info = PlantInfoListCopy(pInfo);
    plantInfo head = PlantInfoListMergeSort(info, SORTWEIGHTR);
    showPlantInfo(head);
}
```





```
void MainWindow::on_actionSortDefault_triggered()
{
    showPlantInfo(pInfo);
}

void MainWindow::on_actionVegSortCode_triggered()
{
    vegInfo info = VegInfoListCopy(vInfo);
    showVegInfo(VegInfoListMergeSort(info, VEGSORTDEFAULT));
}

void MainWindow::on_actionVegSortType_triggered()
{
    vegInfo info = VegInfoListCopy(vInfo);
    showVegInfo(VegInfoListMergeSort(info, VEGSORTDEFAULT));
}

void MainWindow::on_actionVegSortLenth_triggered()
{
    vegInfo info = VegInfoListCopy(vInfo);
    showVegInfo(VegInfoListMergeSort(info, VEGSORTFACTSLENTH));
}

void MainWindow::on_actionShowVegType_triggered()
{
    showTypeInfo(tInfo);
}

void MainWindow::on_actionAddVegType_2_triggered()
{
    AddTypeDialog *dlg = new AddTypeDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
    SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actionDeleteType_triggered()
{
    DeleteTypeDialog *dlg = new DeleteTypeDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
    SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actionEditTypeInfo_triggered()
{
    EditTypeDialog *dlg = new EditTypeDialog(this);
    connect(dlg, SIGNAL(sendBoolean(bool)), this,
    SLOT(receiveBoolean(bool)));
    dlg->show();
}

void MainWindow::on_actionSearchPlantByTypeFacts_triggered()
{
    currentSearchType = PLANTTYPEFACTS;
    Search2InputDialog *dlg = new Search2InputDialog(this);
    connect(dlg, SIGNAL(sendSearchData(searchData)), this,
    SLOT(receiveSearchTwo(searchData)));
    dlg->PrepareShow(QString("蔬菜种类编号"), QString("营养成分关键字"));
    dlg->show();
}
```



```
}

void MainWindow::on_actionSearchPlantByYearName_triggered()
{
    currentSearchType = PLANTYEARFACTS;
    Search2InputDialog *dlg = new Search2InputDialog(this);
    connect(dlg, SIGNAL(sendSearchData(searchData)), this,
    SLOT(receiveSearchTwo(searchData)));
    dlg->PrepareShow(QString("蔬菜种植年份"), QString("营养成分关键字"));
    dlg->show();
}

void MainWindow::on_actionSearchVegInfoTypeFacts_triggered()
{
    currentSearchType = VEGTYPEFACTS;
    Search2InputDialog *dlg = new Search2InputDialog(this);
    connect(dlg, SIGNAL(sendSearchData(searchData)), this,
    SLOT(receiveSearchTwo(searchData)));
    dlg->PrepareShow(QString("蔬菜种类编号"), QString("营养成分关键字"));
    dlg->show();
}

void MainWindow::on_actionSearchVegInfoName_triggered()
{
    currentSearchType = VEGINFONAME;
    SearchInputDialog *dlg = new SearchInputDialog(this);
    connect(dlg, SIGNAL(sendSearch(QString)), this,
    SLOT(receiveSearch(QString)));
    dlg->searchLabel = QString("搜索蔬菜名");
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionShowPlantByYearArea_triggered()
{
    currentRangeType = YEARRANGE;
    RangeInputDialog *dlg = new RangeInputDialog(this);
    connect(dlg, SIGNAL(sendRange(RangeInput)), this,
    SLOT(receiveRange(RangeInput)));
    dlg->PrepareShow(QString("起始年份"), QString("终止年份"));
    dlg->show();
}

void MainWindow::receiveRange(RangeInput range) {
    switch (currentRangeType) {
        case YEARRANGE: {
            plantInfo current = pInfo, head = NULL, currentNew = NULL;
            for (; current != NULL; current = current->next) {
                if (getIntFromString(current->year) >= range.from &&
                getIntFromString(current->year) <= range.to) {
                    if (head == NULL) {
                        head = (plantInfo)malloc(sizeof(struct
plant_info));
                        *head = *current;
                        currentNew = head;
                    }
                    else {
                        currentNew->next =
(plantInfo)malloc(sizeof(struct plant_info));

```



```
        currentNew = currentNew->next;
        *currentNew = *current;
    }
}
currentNew->next = NULL;
showPlantInfo(PlantInfoListMergeSort(head, SORTWEIGHTR));
freePlantInfo(head);
break;
}
case AREARANGE: {
    plantInfo current = pInfo, head = NULL, currentNew = NULL;
    for (; current != NULL; current = current->next) {
        if (current->area >= range.from && current->area <=
range.to) {
            if (head == NULL) {
                head = (plantInfo)malloc(sizeof(struct
plant_info));
                *head = *current;
                currentNew = head;
            }
            else {
                currentNew->next =
(plantInfo)malloc(sizeof(struct plant_info));
                currentNew = currentNew->next;
                *currentNew = *current;
            }
        }
        currentNew->next = NULL;
        showPlantInfo(PlantInfoListMergeSort(head, SORTAREA));
        freePlantInfo(head);
        break;
    }
    case WEIGHTRANGE: {
        plantInfo current = pInfo, head = NULL, currentNew = NULL;
        for (; current != NULL; current = current->next) {
            if (current->weight >= range.from && current->weight <=
range.to) {
                if (head == NULL) {
                    head = (plantInfo)malloc(sizeof(struct
plant_info));
                    *head = *current;
                    currentNew = head;
                }
                else {
                    currentNew->next =
(plantInfo)malloc(sizeof(struct plant_info));
                    currentNew = currentNew->next;
                    *currentNew = *current;
                }
            }
            currentNew->next = NULL;
            showPlantInfo(PlantInfoListMergeSort(head, SORTWEIGHT));
            freePlantInfo(head);
            break;
        }
    }
}
```



```

void MainWindow::on_actionShowPlantByAreaRange_triggered()
{
    currentRangeType = AREARANGE;
    RangeInputDialog *dlg = new RangeInputDialog(this);
    connect(dlg, SIGNAL(sendRange(RangeInput)), this,
    SLOT(receiveRange(RangeInput)));
    dlg->PrepareShow(QString("起始面积"), QString("终止面积"));
    dlg->show();
}

void MainWindow::on_actionShowPlantByWeightRange_triggered()
{
    currentRangeType = WEIGHTRANGE;
    RangeInputDialog *dlg = new RangeInputDialog(this);
    connect(dlg, SIGNAL(sendRange(RangeInput)), this,
    SLOT(receiveRange(RangeInput)));
    dlg->PrepareShow(QString("起始收成"), QString("终止收成"));
    dlg->show();
}

void MainWindow::on_actionCountVegPlantNum_triggered()
{
    VegCodeInputDialog *dlg = new VegCodeInputDialog(this);
    connect(dlg, SIGNAL(sendVegCode(int)), this,
    SLOT(receiveVegCode(int)));
    dlg->show();
}

void MainWindow::receiveVegCode(int vegCode) {
    vegInfo current = vInfo;
    for (; current != NULL; current = current->next) {
        if (current->vegCode == vegCode) {
            plantInfo currentPInfo = current->pInfoList;
            int plantNum = 0;
            for (; currentPInfo != 0; currentPInfo =
currentPInfo->next, plantNum++) ;
            InformationDialog *dlg = new InformationDialog(this);
            dlg->imformation = QString("目前种植蔬菜 %1 的个数
为: %2").arg(current->vegName).arg(plantNum);
            dlg->prepareShow();
            dlg->show();
            return ;
        }
    }
    InformationDialog *dlg = new InformationDialog(this);
    dlg->imformation = QString("无法找到编号为 %1 的蔬菜").arg(vegCode);
    dlg->prepareShow();
    dlg->show();
}

void MainWindow::on_actionloadFromBackup_triggered()
{
    QString fileName = QFileDialog::getOpenFileName(this,
    tr("打开蔬菜数据备份文件"),
    "",
    tr("蔬菜数据备份文件 (*.bak);;所有文件 (*.*)")
    );
}

```



```
        if (getDataFileFromBackupFile(fileName.toUtf8().data())) {
            ui->statusBar->showMessage(QString("读取成功"), 10000);
        }
    }

void MainWindow::on_actionbackupData_triggered()
{
    QString fileName = QFileDialog::getSaveFileName(this,
                                                    tr("蔬菜数据备份文
件要保存到"),
                                                    " ",
                                                    tr("蔬菜数据备份文
件 (*.bak);;所有文件 (*.*)"),
                                                    );
    if (backupDataFileTo(fileName.toUtf8().data())) {
        ui->statusBar->showMessage(QString("保存成功"), 10000);
    }
}
```

### ./CppSource/mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QStandardItemModel>
#include <QtSerialPort/QtSerialPort>
#include "Csource/cppincludes.h"
#include <QtCore/QtGlobal>
#include "search2_input_dialog.h"
#include "range_input_dialog.h"

#define __RUN_IN_QT

/*主窗口---最复杂最重要的窗口*/

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private slots:
    /*
     * @type: 自定义槽函数
     * @param: 是否改变了数据库
     * @true: 刷新 QTable 显示内容
     */
    void receiveBoolean(bool isChanged);
```



```
/*
 * @type: 自定义槽函数
 * @param: 年份
 * @usage: 处理指定年份数据
 * 关联:year_type
 */
void receiveYear(int year);

/*
 * @type: 自定义槽函数
 * @param: 搜索关键词
 * @usage: 搜索
 * 关联:search_type
 */
void receiveSearch(QString data);

/*
 * @type: 自定义槽函数
 * @param: 双项搜索关键词类的对象
 * @usage: 双项搜索
 * 关联:search_type
 */
void receiveSearchTwo(searchData data);

/*
 * @type: 自定义槽函数
 * @param: 输入的范围类的对象
 * @usage: 接受用户输入的范围
 * 关联:range_type
 */
void receiveRange(RangeInput range);

/*
 * @type: 自定义槽函数
 * @param: 蔬菜基本信息编号
 * @usage: 对输入蔬菜基本信息窗口回调进行处理
 */
void receiveVegCode (int vegCode);

/*
 * @type: QAction
 * @usage: 显示增加种植信息 dialog
 */
void on_actionAddPlantInfo_triggered();

/*
 * @type: QAction
 * @usage: 退出程序
 */
void on_action_Quit_triggered();

/*
 * @type: QAction
 * @usage: 显示增加蔬菜品种 dialog
 */
void on_actionAddVegType_triggered();

/*
```



```
* @type: QAction
* @usage: QTable 显示所有种植信息
*/
void on_actionPlantInfoAll_triggered();

/*
* @type: QAction
* @usage: QTable 显示所有蔬菜品种信息
*/
void on_actionVegTypeAll_triggered();

/*
* @type: QAction
* @usage: 显示修改种植信息 dialog
*/
void on_actionEditPlantInfo_triggered();

/*
* @type: QAction
* @usage: 显示修改植物品种信息 dialog
*/
void on_actionEditVegInfo_triggered();

/*
* @type: QAction
* @usage: 保存当前所有数据到文件中
*/
void on_actionSaveAll_triggered();

/*
* @type: QAction
* @usage: 保存种植信息到文件中
*/
void on_actionSavePlantInfo_triggered();

/*
* @type: QAction
* @usage: 保存蔬菜品种信息到文件中
*/
void on_actionSaveVegType_triggered();

/*
* @type: QAction
* @usage: QTable 显示今年[2017]种植信息
*/
void on_actionThisYear_3_triggered();

/*
* @type: QAction
* @usage: 显示输入年份对话框、返回年份信号、
*         QTable 显示指定年份种植信息
*/
void on_actionInputYear_3_triggered();

/*
* @type: QAction
* @usage: 显示 统计总占地的信息 dialog
*/
void on_actionAreaAllVeg_triggered();
```



```
/*
 * @type: QAction
 * @usage: 显示输入年份对话框、返回年份信号、
 *         显示 统计指定年份占地的信息 dialog
 */
void on_actionInputYear_2_triggered();

/*
 * @type: QAction
 * @usage: 显示输入年份对话框、返回年份信号、
 *         显示 今年[2017]占地的信息 dialog
 */
void on_actionThisYear_2_triggered();

/*
 * @type: QAction
 * @usage: 显示 种类占地的信息 dialog
 */
void on_actionType1_2_triggered();
void on_actionType2_2_triggered();
void on_actionType3_2_triggered();
void on_actionType4_2_triggered();
void on_actionType5_2_triggered();

/*
 * @type: QAction
 * @usage: 显示 所有收成的信息 dialog
 */
void on_actionWeightAll_triggered();

/*
 * @type: QAction
 * @usage: 显示 今年[2017]占地的信息 dialog
 */
void on_actionThisYear_triggered();

/*
 * @type: QAction
 * @usage: 显示 输入年份的 dialog
 */
void on_actionInputYear_triggered();

/*
 * @type: QAction
 * @usage: 显示 种类收成的信息 dialog
 */
void on_actionType1_triggered();
void on_actionType2_triggered();
void on_actionType3_triggered();
void on_actionType4_triggered();
void on_actionType5_triggered();

/*
 * 以下是各种菜单项对应的槽函数
 * 报告中有详细功能表
 * 本处不再赘述
 */
```





```
void on_actiondeletePlantInfo_triggered();

void on_actiondeleteVegInfo_triggered();

void on_actionPIType1_triggered();

void on_actionPIType2_triggered();

void on_actionPIType3_triggered();

void on_actionPIType4_triggered();

void on_actionPIType5_triggered();

void on_actionReloadPlantInfo_triggered();

void on_actionReloadVegInfo_triggered();

void on_actionPlantInfoBySearch_triggered();

void on_actionSearchFacts_triggered();

void on_actionSearchFactsVeginfo_triggered();

void on_actionShowVeginfoType1_triggered();

void on_actionShowVeginfoType2_triggered();

void on_actionShowVeginfoType3_triggered();

void on_actionShowVeginfoType2_3_triggered();

void on_actionShowVeginfoType5_triggered();

void on_actionSortBySArea_triggered();

void on_actionSortByYear_triggered();

void on_actionSortByWeight_triggered();

void on_actionSortByType_triggered();

void on_actionSortByAreaR_triggered();

void on_actionSortByYearR_triggered();

void on_actionSortWeightR_triggered();

void on_actionSortDefault_triggered();

void on_actionVegSortCode_triggered();

void on_actionVegSortType_triggered();

void on_actionVegSortLenth_triggered();

void on_actionShowVegType_triggered();

void on_actionAddVegType_2_triggered();

void on_actionDeleteType_triggered();
```



```
void on_actionEditTypeInfo_triggered();

void on_actionSearchPlantByTypeFacts_triggered();

void on_actionSearchPlantByYearName_triggered();

void on_actionSearchVegInfoTypeFacts_triggered();

void on_actionSearchVegInfoName_triggered();

void on_actionShowPlantByYearArea_triggered();

void on_actionShowPlantByAreaRange_triggered();

void on_actionShowPlantByWeightRange_triggered();

void on_actionCountVegPlantNum_triggered();

void on_actionloadFromBackup_triggered();

void on_actionbackupData_triggered();

private:
    Ui::MainWindow *ui;

    /*蔬菜场信息总链表*/
    typeInfo tInfo;
    /*种植信息*/
    plantInfo pInfo;
    /*蔬菜信息*/
    vegInfo vInfo;
    /*QTable 展示内容类型的枚举*/
    enum tableType {
        PLANTALL, PLANTYEAR, PLANTTYPE, PLANTSEARCH, VEGALL,
        VEGYEAR, VEGTYPE, TYPEALL
    };
    /*当前 QTable 展示内容类型*/
    int currentTable = PLANTALL;
    /*当前 QTable 展示的年份__YEAR 时使用*/
    int currentYear;
    /*当前 QTable 展示蔬菜种类__TYPE 时使用*/
    int currentType;

    /*输入年份种类枚举*/
    enum year_type {
        SHOWPLANTINFO, AREA, WEIGHT, AREAWEIGHT
    };
    /*是谁要的输入年份__输入年份 dialog 回调用*/
    int currentYearType;

    /*搜索种类枚举*/
    enum search_type {
        VEGNAME, VEGFACTS, VEGINFOFACTS, VEGINFONAME, PLANTTYPEFACTS,
        PLANTYEARFACTS, VEGTYPEFACTS
    };
    /*搜索类型*/
    int currentSearchType;
```



```
/*范围种类枚举*/
enum range_type {
    YEARRANGE, AREARANGE, WEIGHTRANGE
};
/*范围类型*/
int currentRangeType;

/*QTable 显示所有种植信息*/
void showPlantInfo(plantInfo head);
/*QTable 显示所有蔬菜信息*/
void showVegInfo(vegInfo info);
/*QTable 显示种类信息*/
void showTypeInfo(typeInfo info);
/*QTable 显示指定年份种植信息*/
void showYearPlantInfo (int year);
/*QTable 显示一类蔬菜种植信息*/
void showTypePlantInfo (int type);
/*QTable 显示一类蔬菜信息*/
void showTypeVegInfo(int type);
/*从 QTable 读取种植信息保存到文件*/
void savePlantInfoToFileFromTable();
};
#endif // MAINWINDOW_H
```

### **./CppSource/notfound\_dialog.cpp**

```
#include "notfound_dialog.h"
#include "ui_notfound_dialog.h"

NotFoundDialog::NotFoundDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::NotFoundDialog)
{
    ui->setupUi(this);
}

NotFoundDialog::~NotFoundDialog()
{
    delete ui;
}
```

### **./CppSource/notfound\_dialog.h**

```
#ifndef NOTFOUNDIALOG_H
#define NOTFOUNDIALOG_H

#include <QDialog>

/*未找到窗口 -- 没有任何东西，就是显示*/

namespace Ui {
class NotFoundDialog;
}

class NotFoundDialog : public QDialog
```



```
{
    Q_OBJECT

public:
    explicit NotFoundDialog(QWidget *parent = 0);
    ~NotFoundDialog();

private:
    Ui::NotFoundDialog *ui;
};

#endif // NOTFOUNDDIALOG_H
```

### **./CppSource/range\_input\_dialog.cpp**

```
#include "range_input_dialog.h"
#include "ui_range_input_dialog.h"

RangeInputDialog::RangeInputDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::RangeInputDialog)
{
    ui->setupUi(this);
}

RangeInputDialog::~RangeInputDialog()
{
    delete ui;
}

void RangeInputDialog::PrepareShow(QString rangName1, QString
rangeName2) {
    /*显示前调用，配置两个范围名称*/
    ui->label->setText(rangName1);
    ui->label_2->setText(rangeName2);
}

void RangeInputDialog::on_buttonBox_accepted()
{
    float num1 = ui->lineEdit->text().toFloat();
    float num2 = ui->lineEdit_2->text().toFloat();
    /*根据大小新建范围对象*/
    RangeInput range = RangeInput(num1 < num2 ? num1:num2, num1 <
num2 ? num2:num1);
    /*发送范围信号*/
    emit sendRange(range);
}
```

### **./CppSource/range\_input\_dialog.h**

```
#ifndef RANGE_INPUT_DIALOG_H
#define RANGE_INPUT_DIALOG_H

#include <QDialog>

/*储存输入的范围的类*/
class RangeInput {
public:
```



```
float from, to;
/*简单构造函数*/
RangeInput (float from, float to) {
    this->from = from;
    this->to = to;
}
};

namespace Ui {
class RangeInputDialog;
}

/*输入范围窗口*/

class RangeInputDialog : public QDialog
{
    Q_OBJECT

public:
    explicit RangeInputDialog(QWidget *parent = 0);
    ~RangeInputDialog();
    /*显示前准备*/
    void PrepareShow (QString rangName1, QString rangeName2);
signals:
    /*发送给 MainWindow 的信号*/
    void sendRange (RangeInput);
private slots:
    void on_buttonBox_accepted();

private:
    Ui::RangeInputDialog *ui;
};

#endif // RANGE_INPUT_DIALOG_H
```

### **./CppSource/search2\_input\_dialog.cpp**

```
#include "search2_input_dialog.h"
#include "ui_search2_input_dialog.h"

Search2InputDialog::Search2InputDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::Search2InputDialog)
{
    ui->setupUi(this);
}

Search2InputDialog::~Search2InputDialog()
{
    delete ui;
}

void Search2InputDialog::PrepareShow(QString labelOne, QString
labelTwo) {
    ui->label->setText(labelOne);
    ui->label_2->setText(labelTwo);
}
```



```
void Search2InputDialog::on_buttonBox_accepted()
{
    searchData data = searchData(ui->lineEdit->text(),
    ui->lineEdit_2->text());
    emit sendSearchData(data);
}
```

### **./CppSource/search2\_input\_dialog.h**

```
#ifndef SEARCH2_INPUT_DIALOG_H
#define SEARCH2_INPUT_DIALOG_H

#include <QDialog>

/*双项搜索数据类--存储数据而已*/
class searchData {
public:
    QString data1, data2;
    searchData(QString data1, QString data2) {
        this->data1 = data1;
        this->data2 = data2;
    }
};

namespace Ui {
class Search2InputDialog;
}

/*双项搜索窗口*/

class Search2InputDialog : public QDialog
{
    Q_OBJECT

public:
    explicit Search2InputDialog(QWidget *parent = 0);
    ~Search2InputDialog();
    /*显示前调用*/
    void PrepareShow (QString labelOne, QString labelTwo);

signals:
    /*发送数据给主窗口*/
    void sendSearchData (searchData data);

private slots:
    void on_buttonBox_accepted();

private:
    Ui::Search2InputDialog *ui;
};

#endif // SEARCH2_INPUT_DIALOG_H
```

### **./CppSource/searchinput\_dialog.cpp**

```
#include "searchinput_dialog.h"
```



```
#include "ui_searchinput_dialog.h"

SearchInputDialog::SearchInputDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::SearchInputDialog)
{
    ui->setupUi(this);
}

void SearchInputDialog::prepareShow() {
    ui->label->setText(searchLabel);
}

SearchInputDialog::~SearchInputDialog()
{
    delete ui;
}

void SearchInputDialog::on_buttonBox_accepted()
{
    emit sendSearch(ui->lineEdit->text());
}
```

#### **./CppSource/searchinput\_dialog.h**

```
#ifndef SEARCHINPUT_DIALOG_H
#define SEARCHINPUT_DIALOG_H

#include <QDialog>

namespace Ui {
class SearchInputDialog;
}

/*输入搜索数据窗口*/

class SearchInputDialog : public QDialog
{
    Q_OBJECT

public:
    explicit SearchInputDialog(QWidget *parent = 0);
    ~SearchInputDialog();
    QString searchLabel;
    void prepareShow(void);
signals:
    void sendSearch(QString);
private slots:
    void on_buttonBox_accepted();

private:
    Ui::SearchInputDialog *ui;
};

#endif // SEARCHINPUT_DIALOG_H
```

#### **./CppSource/vegcode\_input\_dialog.cpp**

```
#include "vegcode_input_dialog.h"
#include "ui_vegcode_input_dialog.h"
```



```
VegCodeInputDialog::VegCodeInputDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::VegCodeInputDialog)
{
    ui->setupUi(this);
}

VegCodeInputDialog::~VegCodeInputDialog()
{
    delete ui;
}

void VegCodeInputDialog::on_buttonBox_accepted()
{
    emit sendVegCode(ui->lineEdit->text().toInt());
}
```

### **./CppSource/vegcode\_input\_dialog.h**

```
#ifndef VEGCODE_INPUT_DIALOG_H
#define VEGCODE_INPUT_DIALOG_H

#include <QDialog>

namespace Ui {
class VegCodeInputDialog;
}

/*输入蔬菜基本信息编号窗口*/

class VegCodeInputDialog : public QDialog
{
    Q_OBJECT

public:
    explicit VegCodeInputDialog(QWidget *parent = 0);
    ~VegCodeInputDialog();
signals:
    /*发送结果给主窗口*/
    void sendVegCode(int vegCode);
private slots:
    void on_buttonBox_accepted();

private:
    Ui::VegCodeInputDialog *ui;
};

#endif // VEGCODE_INPUT_DIALOG_H
```

### **./CppSource/yearinput\_dialog.cpp**

```
#include "yearinput_dialog.h"
#include "ui_yearinput_dialog.h"

YearInputDialog::YearInputDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::YearInputDialog)
```





```
{
    ui->setupUi(this);
}

YearInputDialog::~YearInputDialog()
{
    delete ui;
}

void YearInputDialog::on_buttonBox_accepted()
{
    emit sendYear(ui->lineEdit->text().toInt());
}
```

### **./CppSource/yearinput\_dialog.h**

```
#ifndef YEARINPUTDIALOG_H
#define YEARINPUTDIALOG_H

#include <QDialog>

namespace Ui {
class YearInputDialog;
}

/*输入年份的窗口*/

class YearInputDialog : public QDialog
{
    Q_OBJECT

public:
    explicit YearInputDialog(QWidget *parent = 0);
    ~YearInputDialog();
signals:
    /*发送结果给主窗口*/
    void sendYear(int year);

private slots:
    void on_buttonBox_accepted();

private:
    Ui::YearInputDialog *ui;
};

#endif // YEARINPUTDIALOG_H
```



### **./UI/add\_plantinfo\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>AddPlantInfoDialog</class>
  <widget class="QDialog" name="AddPlantInfoDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>300</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>新增种植作物信息</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
```



```
<rect>
  <x>30</x>
  <y>240</y>
  <width>341</width>
  <height>32</height>
</rect>
</property>
<property name="orientation">
  <enum>Qt::Horizontal</enum>
</property>
<property name="standardButtons">
  <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
</property>
<property name="centerButtons">
  <bool>true</bool>
</property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>30</y>
      <width>321</width>
      <height>171</height>
    </rect>
  </property>
  <layout class="QGridLayout" name="gridLayout">
    <item row="0" column="1">
      <widget class="QLineEdit" name="textVegCode"/>
    </item>
    <item row="1" column="1">
      <widget class="QLineEdit" name="textArea"/>
    </item>
    <item row="2" column="1">
      <widget class="QLineEdit" name="textWeight"/>
    </item>
    <item row="3" column="1">
      <widget class="QLineEdit" name="textYear"/>
    </item>
    <item row="0" column="0">
      <widget class="QLabel" name="label">
        <property name="text">
          <string>蔬菜编号</string>
        </property>
      </widget>
    </item>
    <item row="1" column="0">
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>种植面积</string>
        </property>
      </widget>
    </item>
    <item row="2" column="0">
      <widget class="QLabel" name="label_3">
        <property name="text">
          <string>收获重量</string>
        </property>
      </widget>
    </item>
```



```

    <item row="3" column="0">
        <widget class="QLabel" name="label_4">
            <property name="text">
                <string>种植年份</string>
            </property>
        </widget>
    </item>
</layout>
</widget>
</widget>
<resources/>
<connections>
    <connection>
        <sender>buttonBox</sender>
        <signal>accepted()</signal>
        <receiver>AddPlantInfoDialog</receiver>
        <slot>accept()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>248</x>
                <y>254</y>
            </hint>
            <hint type="destinationlabel">
                <x>157</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
    <connection>
        <sender>buttonBox</sender>
        <signal>rejected()</signal>
        <receiver>AddPlantInfoDialog</receiver>
        <slot>reject()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>316</x>
                <y>260</y>
            </hint>
            <hint type="destinationlabel">
                <x>286</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
</connections>
</ui>

```

### ./UI/add\_type\_dialog.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>AddTypeDialog</class>
    <widget class="QDialog" name="AddTypeDialog">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>400</width>
                <height>150</height>
            </rect>

```



```
</property>
<property name="minimumSize">
  <size>
    <width>400</width>
    <height>150</height>
  </size>
</property>
<property name="maximumSize">
  <size>
    <width>400</width>
    <height>150</height>
  </size>
</property>
<property name="windowTitle">
  <string notr="true">新增蔬菜种类信息</string>
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>100</y>
      <width>341</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>10</y>
      <width>341</width>
      <height>80</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label">
        <property name="text">
          <string>种类名称</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
```



```

<sender>buttonBox</sender>
<signal>accepted()</signal>
<receiver>AddTypeDialog</receiver>
<slot>accept()</slot>
<hints>
  <hint type="sourcelabel">
    <x>248</x>
    <y>254</y>
  </hint>
  <hint type="destinationlabel">
    <x>157</x>
    <y>274</y>
  </hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>AddTypeDialog</receiver>
  <slot>reject()</slot>
  <hints>
    <hint type="sourcelabel">
      <x>316</x>
      <y>260</y>
    </hint>
    <hint type="destinationlabel">
      <x>286</x>
      <y>274</y>
    </hint>
  </hints>
</connection>
</connections>
</ui>

```

### ./UI/add\_vegtype\_dialog.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>AddVegTypeDialog</class>
  <widget class="QDialog" name="AddVegTypeDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>300</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>新增蔬菜品种</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>240</y>
          <width>341</width>
          <height>32</height>
        </rect>
      </property>
    </widget>
  </widget>
</ui>

```



```
<property name="orientation">
  <enum>Qt::Horizontal</enum>
</property>
<property name="standardButtons">
  <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
</property>
<property name="centerButtons">
  <bool>true</bool>
</property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>10</y>
      <width>351</width>
      <height>117</height>
    </rect>
  </property>
  <layout class="QGridLayout" name="gridLayout">
    <item row="1" column="1">
      <widget class="QLineEdit" name="vegName"/>
    </item>
    <item row="1" column="0">
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>蔬菜名称</string>
        </property>
      </widget>
    </item>
    <item row="2" column="0">
      <widget class="QLabel" name="label">
        <property name="minimumSize">
          <size>
            <width>70</width>
            <height>0</height>
          </size>
        </property>
        <property name="text">
          <string>蔬菜种类</string>
        </property>
      </widget>
    </item>
    <item row="2" column="1">
      <widget class="QLineEdit" name="vegType"/>
    </item>
  </layout>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>130</y>
      <width>351</width>
      <height>91</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label_3">
        <property name="minimumSize">
```



```

        <size>
            <width>70</width>
            <height>0</height>
        </size>
    </property>
    <property name="text">
        <string>营养成分</string>
    </property>
</widget>
</item>
<item>
    <widget class="QTextEdit" name="vegN"/>
</item>
</layout>
</widget>
</widget>
<resources/>
<connections>
    <connection>
        <sender>buttonBox</sender>
        <signal>accepted()</signal>
        <receiver>AddVegTypeDialog</receiver>
        <slot>accept()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>248</x>
                <y>254</y>
            </hint>
            <hint type="destinationlabel">
                <x>157</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
    <connection>
        <sender>buttonBox</sender>
        <signal>rejected()</signal>
        <receiver>AddVegTypeDialog</receiver>
        <slot>reject()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>316</x>
                <y>260</y>
            </hint>
            <hint type="destinationlabel">
                <x>286</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
</connections>
</ui>

```

### **./UI/delete\_plantinfo\_dialog.ui**

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>DeletePlantInfoDialog</class>
    <widget class="QDialog" name="DeletePlantInfoDialog">
        <property name="geometry">

```





```
<rect>
  <x>0</x>
  <y>0</y>
  <width>400</width>
  <height>200</height>
</rect>
</property>
<property name="minimumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="maximumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="windowTitle">
  <string>删除种植信息</string>
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>140</y>
      <width>341</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>110</x>
      <y>10</y>
      <width>191</width>
      <height>51</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>Microsoft YaHei</family>
      <pointsize>10</pointsize>
    </font>
  </property>
  <property name="text">
    <string>删除一项或多项种植信息
删除多项使用英文','分隔</string>
  </property>
```



```
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>39</x>
      <y>70</y>
      <width>321</width>
      <height>51</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>种植编号</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>DeletePlantInfoDialog</receiver>
    <slot>accept()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>248</x>
        <y>254</y>
      </hint>
      <hint type="destinationlabel">
        <x>157</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
  <connection>
    <sender>buttonBox</sender>
    <signal>rejected()</signal>
    <receiver>DeletePlantInfoDialog</receiver>
    <slot>reject()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>316</x>
        <y>260</y>
      </hint>
      <hint type="destinationlabel">
        <x>286</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
</connections>
</ui>
```

**./UI/delete\_type\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>DeleteTypeDialog</class>
  <widget class="QDialog" name="DeleteTypeDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>150</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="windowTitle">
      <string notr="true">删除蔬菜种类信息</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>100</y>
          <width>341</width>
          <height>32</height>
        </rect>
      </property>
      <property name="orientation">
        <enum>Qt::Horizontal</enum>
      </property>
      <property name="standardButtons">
        <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
      </property>
      <property name="centerButtons">
        <bool>true</bool>
      </property>
    </widget>
    <widget class="QWidget" name="horizontalLayoutWidget">
      <property name="geometry">
        <rect>
          <x>20</x>
          <y>10</y>
          <width>361</width>
          <height>80</height>
        </rect>
      </property>
      <layout class="QHBoxLayout" name="horizontalLayout">
        <item>
          <widget class="QLabel" name="label">
```



```

        <property name="text">
            <string>蔬菜种类编号</string>
        </property>
    </widget>
</item>
<item>
    <widget class="QLineEdit" name="lineEdit"/>
</item>
</layout>
</widget>
</widget>
<resources/>
<connections>
    <connection>
        <sender>buttonBox</sender>
        <signal>accepted()</signal>
        <receiver>DeleteTypeDialog</receiver>
        <slot>accept()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>248</x>
                <y>254</y>
            </hint>
            <hint type="destinationlabel">
                <x>157</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
    <connection>
        <sender>buttonBox</sender>
        <signal>rejected()</signal>
        <receiver>DeleteTypeDialog</receiver>
        <slot>reject()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>316</x>
                <y>260</y>
            </hint>
            <hint type="destinationlabel">
                <x>286</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
</connections>
</ui>

```

### **./UI/delete\_veginfo\_dialog.ui**

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>DeleteVegInfoDialog</class>
    <widget class="QDialog" name="DeleteVegInfoDialog">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>400</width>
                <height>200</height>
            </rect>
        </property>
    </widget>
</ui>

```



```
</rect>
</property>
<property name="minimumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="maximumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="acceptDrops">
  <bool>false</bool>
</property>
<property name="windowTitle">
  <string notr="true">删除蔬菜信息</string>
</property>
<property name="whatsThis">
  <string notr="true"/>
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>150</y>
      <width>341</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>100</x>
      <y>0</y>
      <width>201</width>
      <height>81</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>Microsoft YaHei</family>
      <pointsize>10</pointsize>
    </font>
  </property>
  <property name="text">
    <string>删除一项或多项蔬菜信息
    删除多项使用英文','分隔</string>
```



```
</property>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>80</y>
      <width>341</width>
      <height>51</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>蔬菜编号</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>DeleteVegInfoDialog</receiver>
    <slot>accept()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>248</x>
        <y>254</y>
      </hint>
      <hint type="destinationlabel">
        <x>157</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
  <connection>
    <sender>buttonBox</sender>
    <signal>rejected()</signal>
    <receiver>DeleteVegInfoDialog</receiver>
    <slot>reject()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>316</x>
        <y>260</y>
      </hint>
      <hint type="destinationlabel">
        <x>286</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
</connections>
</ui>
```

**./UI/edit\_plantinfo\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>EditPlantInfoDialog</class>
  <widget class="QDialog" name="EditPlantInfoDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>300</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>修改已添加的种植蔬菜信息</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>240</y>
          <width>341</width>
          <height>32</height>
        </rect>
      </property>
      <property name="orientation">
        <enum>Qt::Horizontal</enum>
      </property>
      <property name="standardButtons">
        <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
      </property>
      <property name="centerButtons">
        <bool>true</bool>
      </property>
    </widget>
    <widget class="QWidget" name="gridLayoutWidget">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>60</y>
          <width>331</width>
          <height>171</height>
        </rect>
      </property>
      <layout class="QGridLayout" name="gridLayout">
        <item row="3" column="0">
          <widget class="QLabel" name="label_4">
            <property name="font">
              <font>
                <family>Microsoft YaHei</family>
              </font>
            </property>
            <property name="text">
              <string>种植年份</string>
            </property>
          </widget>
        </item>
        <item row="0" column="0">
```



```
<widget class="QLabel" name="label">
  <property name="font">
    <font>
      <family>Microsoft YaHei</family>
    </font>
  </property>
  <property name="text">
    <string>蔬菜名称/编号</string>
  </property>
</widget>
</item>
<item row="1" column="0">
  <widget class="QLabel" name="label_2">
    <property name="font">
      <font>
        <family>Microsoft YaHei</family>
      </font>
    </property>
    <property name="text">
      <string>种植面积</string>
    </property>
  </widget>
</item>
<item row="2" column="0">
  <widget class="QLabel" name="label_3">
    <property name="font">
      <font>
        <family>Microsoft YaHei</family>
      </font>
    </property>
    <property name="text">
      <string>收成/kg</string>
    </property>
  </widget>
</item>
<item row="0" column="1">
  <widget class="QLineEdit" name="vegCode"/>
</item>
<item row="1" column="1">
  <widget class="QLineEdit" name="area"/>
</item>
<item row="2" column="1">
  <widget class="QLineEdit" name="weight"/>
</item>
<item row="3" column="1">
  <widget class="QLineEdit" name="year"/>
</item>
</layout>
</widget>
<widget class="QLabel" name="label_5">
  <property name="geometry">
    <rect>
      <x>140</x>
      <y>20</y>
      <width>121</width>
      <height>31</height>
    </rect>
  </property>
  <property name="font">
    <font>
```





```

        <family>Microsoft YaHei</family>
        <pointsize>10</pointsize>
    </font>
</property>
<property name="text">
    <string>留空代表不修改</string>
</property>
</widget>
</widget>
</resources/>
<connections>
    <connection>
        <sender>buttonBox</sender>
        <signal>accepted()</signal>
        <receiver>EditPlantInfoDialog</receiver>
        <slot>accept()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>248</x>
                <y>254</y>
            </hint>
            <hint type="destinationlabel">
                <x>157</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
    <connection>
        <sender>buttonBox</sender>
        <signal>rejected()</signal>
        <receiver>EditPlantInfoDialog</receiver>
        <slot>reject()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>316</x>
                <y>260</y>
            </hint>
            <hint type="destinationlabel">
                <x>286</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
</connections>
</ui>

```

### ./UI/edit\_type\_dialog.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>EditTypeDialog</class>
    <widget class="QDialog" name="EditTypeDialog">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>400</width>
                <height>200</height>
            </rect>
        </property>
    </widget>
</ui>

```



```
<property name="minimumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="maximumSize">
  <size>
    <width>400</width>
    <height>200</height>
  </size>
</property>
<property name="windowTitle">
  <string>修改蔬菜种类信息</string>
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>150</y>
      <width>400</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>20</y>
      <width>361</width>
      <height>111</height>
    </rect>
  </property>
  <layout class="QGridLayout" name="gridLayout">
    <item row="1" column="1">
      <widget class="QLineEdit" name="lineEdit_2"/>
    </item>
    <item row="0" column="1">
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
    <item row="0" column="0">
      <widget class="QLabel" name="label">
        <property name="text">
          <string>种类编号</string>
        </property>
      </widget>
    </item>
    <item row="1" column="0">
      <widget class="QLabel" name="label_2">
        <property name="text">
```



```

        <string>种类名称</string>
    </property>
</widget>
</item>
</layout>
</widget>
</widget>
</resources/>
<connections>
<connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>EditTypeDialog</receiver>
    <slot>accept()</slot>
    <hints>
        <hint type="sourcelabel">
            <x>248</x>
            <y>254</y>
        </hint>
        <hint type="destinationlabel">
            <x>157</x>
            <y>274</y>
        </hint>
    </hints>
</connection>
<connection>
    <sender>buttonBox</sender>
    <signal>rejected()</signal>
    <receiver>EditTypeDialog</receiver>
    <slot>reject()</slot>
    <hints>
        <hint type="sourcelabel">
            <x>316</x>
            <y>260</y>
        </hint>
        <hint type="destinationlabel">
            <x>286</x>
            <y>274</y>
        </hint>
    </hints>
</connection>
</connections>
</ui>

```

### ./UI/edit\_veginfo\_dialog.ui

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>EditVegInfoDialog</class>
    <widget class="QDialog" name="EditVegInfoDialog">
        <property name="geometry">
            <rect>
                <x>0</x>
                <y>0</y>
                <width>400</width>
                <height>300</height>
            </rect>
        </property>
        <property name="windowTitle">
            <string>修改蔬菜品种信息</string>

```



```
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>240</y>
      <width>341</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>60</y>
      <width>331</width>
      <height>171</height>
    </rect>
  </property>
  <layout class="QGridLayout" name="gridLayout">
    <property name="topMargin">
      <number>0</number>
    </property>
    <property name="horizontalSpacing">
      <number>7</number>
    </property>
    <property name="verticalSpacing">
      <number>20</number>
    </property>
    <item row="1" column="0">
      <widget class="QLabel" name="label_2">
        <property name="font">
          <font>
            <family>Microsoft YaHei</family>
          </font>
        </property>
        <property name="text">
          <string>蔬菜种类</string>
        </property>
        <property name="alignment">
          <set>Qt::AlignCenter</set>
        </property>
      </widget>
    </item>
    <item row="0" column="0">
      <widget class="QLabel" name="label">
        <property name="font">
          <font>
            <family>Microsoft YaHei</family>
          </font>
        </property>
      </widget>
    </item>
  </layout>
</widget>
```



```
<property name="text">
  <string>蔬菜编号</string>
</property>
<property name="alignment">
  <set>Qt::AlignCenter</set>
</property>
</widget>
</item>
<item row="2" column="0">
  <widget class="QLabel" name="label_3">
    <property name="font">
      <font>
        <family>Microsoft YaHei</family>
      </font>
    </property>
    <property name="text">
      <string>营养成分</string>
    </property>
    <property name="alignment">
      <set>Qt::AlignCenter</set>
    </property>
  </widget>
</item>
<item row="0" column="1">
  <widget class="QLineEdit" name="vegCode"/>
</item>
<item row="1" column="1">
  <widget class="QLineEdit" name="vegType"/>
</item>
<item row="2" column="1">
  <widget class="QTextEdit" name="textEdit"/>
</item>
</layout>
</widget>
<widget class="QLabel" name="label_4">
  <property name="geometry">
    <rect>
      <x>140</x>
      <y>20</y>
      <width>121</width>
      <height>21</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>Microsoft YaHei UI</family>
      <pointsize>10</pointsize>
    </font>
  </property>
  <property name="text">
    <string>留空代表不输入</string>
  </property>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>EditVegInfoDialog</receiver>
```



```
<slot>accept ()</slot>
<hints>
  <hint type="sourcelabel">
    <x>248</x>
    <y>254</y>
  </hint>
  <hint type="destinationlabel">
    <x>157</x>
    <y>274</y>
  </hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>EditVegInfoDialog</receiver>
  <slot>reject()</slot>
  <hints>
    <hint type="sourcelabel">
      <x>316</x>
      <y>260</y>
    </hint>
    <hint type="destinationlabel">
      <x>286</x>
      <y>274</y>
    </hint>
  </hints>
</connection>
</connections>
</ui>
```

### **./UI/imformation\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>ImformationDialog</class>
  <widget class="QDialog" name="ImformationDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>500</width>
        <height>200</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>500</width>
        <height>200</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>500</width>
        <height>200</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>提示</string>
    </property>
```



```
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>10</y>
      <width>461</width>
      <height>171</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>Microsoft YaHei</family>
      <pointsize>11</pointsize>
    </font>
  </property>
  <property name="text">
    <string>TextLabel</string>
  </property>
  <property name="alignment">
    <set>Qt::AlignLeading|Qt::AlignLeft|Qt::AlignVCenter</set>
  </property>
</widget>
</widget>
<resources/>
<connections/>
</ui>
```

### **./UI/mainwindow.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>MainWindow</class>
  <widget class="QMainWindow" name="MainWindow">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>1000</width>
        <height>600</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>1000</width>
        <height>600</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>1000</width>
        <height>600</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>蔬菜种植管理系统</string>
    </property>
    <widget class="QWidget" name="centralWidget">
      <widget class="QTableView" name="tableView">
        <property name="geometry">
          <rect>
```



```
<x>40</x>
<y>110</y>
<width>921</width>
<height>421</height>
</rect>
</property>
<property name="textElideMode">
  <enum>Qt::ElideLeft</enum>
</property>
<attribute name="horizontalHeaderCascadingSectionResizes">
  <bool>>false</bool>
</attribute>
<attribute name="horizontalHeaderDefaultSectionSize">
  <number>120</number>
</attribute>
<attribute name="horizontalHeaderMinimumSectionSize">
  <number>80</number>
</attribute>
<attribute name="horizontalHeaderShowSortIndicator" stdset="0">
  <bool>>false</bool>
</attribute>
<attribute name="horizontalHeaderStretchLastSection">
  <bool>>true</bool>
</attribute>
<attribute name="verticalHeaderVisible">
  <bool>>false</bool>
</attribute>
</widget>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>50</x>
      <y>20</y>
      <width>901</width>
      <height>51</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
      <pointsize>10</pointsize>
    </font>
  </property>
  <property name="text">
    <string>欢迎使用 HT 菜农种植信息管理系统 v1.0
  </string>
  </property>
</widget>
<widget class="QLabel" name="tableType">
  <property name="geometry">
    <rect>
      <x>50</x>
      <y>80</y>
      <width>231</width>
      <height>21</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
```





```
<pointsiz>10</pointsiz>
</font>
</property>
<property name="text">
  <string>当前显示的是菜园种植概况</string>
</property>
</widget>
</widget>
<widget class="QMenuBar" name="menuBar">
  <property name="geometry">
    <rect>
      <x>0</x>
      <y>0</y>
      <width>1000</width>
      <height>26</height>
    </rect>
  </property>
  <widget class="QMenu" name="menu">
    <property name="title">
      <string>新增 (&N)</string>
    </property>
    <addaction name="actionAddPlantInfo"/>
    <addaction name="actionAddVegType"/>
    <addaction name="actionAddVegType_2"/>
  </widget>
  <widget class="QMenu" name="show">
    <property name="title">
      <string>查看 (&W)</string>
    </property>
    <widget class="QMenu" name="menu_4">
      <property name="title">
        <string>按年份查看种植信息</string>
      </property>
      <addaction name="actionThisYear_3"/>
      <addaction name="actionInputYear_3"/>
      <addaction name="actionShowPlantByYearArea"/>
    </widget>
    <widget class="QMenu" name="menuPlantInfoYear">
      <property name="title">
        <string>按种类查看种植信息</string>
      </property>
      <addaction name="actionPIType1"/>
      <addaction name="actionPIType2"/>
      <addaction name="actionPIType3"/>
      <addaction name="actionPIType4"/>
      <addaction name="actionPIType5"/>
    </widget>
    <widget class="QMenu" name="menu_6">
      <property name="title">
        <string>搜索</string>
      </property>
      <addaction name="actionPlantInfoBySearch"/>
      <addaction name="actionSearchFacts"/>
      <addaction name="actionSearchPlantByTypeFacts"/>
      <addaction name="actionSearchPlantByYearName"/>
      <addaction name="separator"/>
      <addaction name="actionSearchFactsVeginfo"/>
      <addaction name="actionSearchVegInfoName"/>
      <addaction name="actionSearchVegInfoTypeFacts"/>
    </widget>
```



```
<widget class="QMenu" name="menuShowVegInfoType">
  <property name="title">
    <string>按分类查看蔬菜信息</string>
  </property>
  <addaction name="actionShowVeginfoType1"/>
  <addaction name="actionShowVeginfoType2"/>
  <addaction name="actionShowVeginfoType3"/>
  <addaction name="actionShowVeginfoType2_3"/>
  <addaction name="actionShowVeginfoType5"/>
</widget>
<addaction name="actionPlantInfoAll"/>
<addaction name="actionVegTypeAll"/>
<addaction name="actionShowVegType"/>
<addaction name="separator"/>
<addaction name="menu_4"/>
<addaction name="menuPlantInfoYear"/>
<addaction name="actionShowPlantByAreaRange"/>
<addaction name="actionShowPlantByWeightRange"/>
<addaction name="separator"/>
<addaction name="menuShowVegInfoType"/>
<addaction name="separator"/>
<addaction name="menu_6"/>
</widget>
<widget class="QMenu" name="menu_F">
  <property name="title">
    <string>文件(&F)</string>
  </property>
  <addaction name="actionReloadPlantInfo"/>
  <addaction name="actionReloadVegInfo"/>
  <addaction name="actionSaveAll"/>
  <addaction name="actionSavePlantInfo"/>
  <addaction name="actionSaveVegType"/>
  <addaction name="separator"/>
  <addaction name="actionbackupData"/>
  <addaction name="actionloadFromBackup"/>
  <addaction name="separator"/>
  <addaction name="action_Quit"/>
</widget>
<widget class="QMenu" name="menu_3">
  <property name="title">
    <string>修改(&M)</string>
  </property>
  <addaction name="actionEditPlantInfo"/>
  <addaction name="actionEditVegInfo"/>
  <addaction name="actionEditTypeInfo"/>
</widget>
<widget class="QMenu" name="menu_C">
  <property name="title">
    <string>统计(&C)</string>
  </property>
  <widget class="QMenu" name="menuArea">
    <property name="title">
      <string>蔬菜种植面积总和</string>
    </property>
    <widget class="QMenu" name="menuAreaYear">
      <property name="title">
        <string>按年份</string>
      </property>
      <addaction name="actionThisYear_2"/>
      <addaction name="actionInputYear_2"/>
    </widget>
  </widget>
</widget>
```



```
</widget>
<widget class="QMenu" name="menu_5">
  <property name="title">
    <string>按蔬菜种类</string>
  </property>
  <addaction name="actionType1_2"/>
  <addaction name="actionType2_2"/>
  <addaction name="actionType3_2"/>
  <addaction name="actionType34"/>
  <addaction name="actionType5_2"/>
</widget>
<addaction name="actionAreaAllVeg"/>
<addaction name="separator"/>
<addaction name="menuAreaYear"/>
<addaction name="menu_5"/>
</widget>
<widget class="QMenu" name="menuWeight">
  <property name="title">
    <string>蔬菜收成总和</string>
  </property>
  <widget class="QMenu" name="menuWeightType">
    <property name="title">
      <string>按蔬菜种类</string>
    </property>
    <addaction name="actionType1"/>
    <addaction name="actionType2"/>
    <addaction name="actionType3"/>
    <addaction name="actionType4"/>
    <addaction name="actionType5"/>
  </widget>
  <widget class="QMenu" name="menuWeightYear">
    <property name="title">
      <string>按年份</string>
    </property>
    <addaction name="actionThisYear"/>
    <addaction name="actionInputYear"/>
  </widget>
  <addaction name="actionWeightAll"/>
  <addaction name="separator"/>
  <addaction name="menuWeightYear"/>
  <addaction name="menuWeightType"/>
</widget>
<addaction name="menuArea"/>
<addaction name="menuWeight"/>
<addaction name="actionCountVegPlantNum"/>
</widget>
<widget class="QMenu" name="menu_2">
  <property name="title">
    <string>排序(&P)</string>
  </property>
  <widget class="QMenu" name="menusa">
    <property name="title">
      <string>按面积排序种植信息</string>
    </property>
    <addaction name="actionSortBySArea"/>
    <addaction name="actionSortByAreaR"/>
  </widget>
  <widget class="QMenu" name="menua">
    <property name="title">
      <string>按年份排序种植信息</string>
    </property>
```



```
</property>
<addaction name="actionSortByYear"/>
<addaction name="actionSortByYearR"/>
</widget>
<widget class="QMenu" name="menusa_2">
<property name="title">
<string>接收成排序种植信息</string>
</property>
<addaction name="actionSortByWeight"/>
<addaction name="actionSortWeightR"/>
</widget>
<addaction name="actionSortDefault"/>
<addaction name="actionSortByType"/>
<addaction name="menusa"/>
<addaction name="menua"/>
<addaction name="menusa_2"/>
<addaction name="separator"/>
<addaction name="actionVegSortType"/>
<addaction name="actionVegSortCode"/>
<addaction name="actionVegSortLenth"/>
</widget>
<widget class="QMenu" name="menu_D">
<property name="title">
<string>删除 (&D)</string>
</property>
<addaction name="actiondeletePlantInfo"/>
<addaction name="actiondeleteVegInfo"/>
<addaction name="actionDeleteType"/>
</widget>
<addaction name="menu_F"/>
<addaction name="menu"/>
<addaction name="menu_3"/>
<addaction name="menu_D"/>
<addaction name="show"/>
<addaction name="menu_C"/>
<addaction name="menu_2"/>
</widget>
<widget class="QStatusBar" name="statusBar">
<property name="autoFillBackground">
<bool>>false</bool>
</property>
<property name="styleSheet">
<string notr="true">background-color: rgb(222, 222, 222);color:
rgb(0,0,0);</string>
</property>
</widget>
<action name="actionNew">
<property name="icon">
<iconset>
<normaloff>../../Desktop/QQ 图片
20170709210854.png</normaloff>../../Desktop/QQ 图片
20170709210854.png</iconset>
</property>
<property name="text">
<string>新增 (&N)</string>
</property>
</action>
<action name="actionAddVegType">
<property name="text">
<string>新增蔬菜基本信息</string>
```



```
</property>
<property name="toolTip">
  <string>新增蔬菜种类</string>
</property>
<property name="font">
  <font>
    <family>微软雅黑</family>
  </font>
</property>
</action>
<action name="actionAddPlantInfo">
  <property name="text">
    <string>新增蔬菜种植信息</string>
  </property>
  <property name="toolTip">
    <string>新增种植蔬菜信息</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
</action>
<action name="actionPlantInfoAll">
  <property name="text">
    <string>蔬菜种植信息</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
</action>
<action name="actionVegTypeAll">
  <property name="text">
    <string>蔬菜基本信息</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
</action>
<action name="actionSaveVegType">
  <property name="text">
    <string>保存蔬菜品种信息</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
</action>
<action name="actionSavePlantInfo">
  <property name="text">
    <string>保存种植信息</string>
  </property>
  <property name="font">
    <font>
```



```
<family>微软雅黑</family>
</font>
</property>
</action>
<action name="actionSaveAll">
  <property name="text">
    <string>保存所有 (&S)</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
  <property name="shortcut">
    <string>Ctrl+S</string>
  </property>
</action>
<action name="action_Quit">
  <property name="text">
    <string>退出 (&Q)</string>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
    </font>
  </property>
</action>
<action name="actionEditPlantInfo">
  <property name="text">
    <string>修改蔬菜种植信息</string>
  </property>
</action>
<action name="actionEditVegInfo">
  <property name="text">
    <string>修改蔬菜基本信息</string>
  </property>
</action>
<action name="actionAreaAllVeg">
  <property name="text">
    <string>所有蔬菜</string>
  </property>
</action>
<action name="actionWeightAll">
  <property name="text">
    <string>所有蔬菜</string>
  </property>
</action>
<action name="actionType1">
  <property name="text">
    <string>根茎类</string>
  </property>
</action>
<action name="actionType2">
  <property name="text">
    <string>果菜类</string>
  </property>
</action>
<action name="actionType3">
  <property name="text">
```



```
<string>瓜类</string>
</property>
</action>
<action name="actionType4">
  <property name="text">
    <string>叶菜类</string>
  </property>
</action>
<action name="actionType5">
  <property name="text">
    <string>菌类</string>
  </property>
</action>
<action name="actionThisYear">
  <property name="text">
    <string>今年</string>
  </property>
</action>
<action name="actionInputYear">
  <property name="text">
    <string>输入年份</string>
  </property>
</action>
<action name="actionThisYear_2">
  <property name="text">
    <string>今年</string>
  </property>
</action>
<action name="actionInputYear_2">
  <property name="text">
    <string>输入年份</string>
  </property>
</action>
<action name="actionThisYear_3">
  <property name="text">
    <string>今年</string>
  </property>
</action>
<action name="actionInputYear_3">
  <property name="text">
    <string>输入年份</string>
  </property>
</action>
<action name="actionType1_2">
  <property name="text">
    <string>根茎类</string>
  </property>
</action>
<action name="actionType2_2">
  <property name="text">
    <string>果菜类</string>
  </property>
</action>
<action name="actionType3_2">
  <property name="text">
    <string>瓜类</string>
  </property>
</action>
<action name="actionTyp34">
```



```
<property name="text">
  <string>叶菜类</string>
</property>
</action>
<action name="actionType5_2">
  <property name="text">
    <string>菌类</string>
  </property>
</action>
<action name="actionSortBySArea">
  <property name="text">
    <string>从小到大</string>
  </property>
</action>
<action name="actionSortByYear">
  <property name="text">
    <string>从小到大</string>
  </property>
</action>
<action name="actionSortByWeight">
  <property name="text">
    <string>从小到大</string>
  </property>
</action>
<action name="actionSortByType">
  <property name="text">
    <string>按种类排序种植信息</string>
  </property>
</action>
<action name="actionPlantInfoBySearch">
  <property name="text">
    <string>菜名关键字</string>
  </property>
</action>
<action name="actionPIType1">
  <property name="text">
    <string>根茎类</string>
  </property>
</action>
<action name="actionPIType2">
  <property name="text">
    <string>果菜类</string>
  </property>
</action>
<action name="actionPIType3">
  <property name="text">
    <string>瓜类</string>
  </property>
</action>
<action name="actionPIType4">
  <property name="text">
    <string>叶菜类</string>
  </property>
</action>
<action name="actionPIType5">
  <property name="text">
    <string>菌类</string>
  </property>
</action>
```





```
<action name="actiondeletePlantInfo">
  <property name="text">
    <string>删除蔬菜种植信息</string>
  </property>
</action>
<action name="actiondeleteVegInfo">
  <property name="text">
    <string>删除蔬菜基本信息</string>
  </property>
</action>
<action name="actionReloadPlantInfo">
  <property name="text">
    <string>重新读取种植数据</string>
  </property>
</action>
<action name="actionReloadVegInfo">
  <property name="text">
    <string>重新读取蔬菜基本数据</string>
  </property>
</action>
<action name="actionSearchFacts">
  <property name="text">
    <string>含营养成分的种植信息</string>
  </property>
</action>
<action name="actionSearchFactsVeginfo">
  <property name="text">
    <string>含营养成分的蔬菜信息</string>
  </property>
</action>
<action name="actionShowVeginfoType1">
  <property name="text">
    <string>根茎类</string>
  </property>
</action>
<action name="actionShowVeginfoType2">
  <property name="text">
    <string>果菜类</string>
  </property>
</action>
<action name="actionShowVeginfoType3">
  <property name="text">
    <string>瓜类</string>
  </property>
</action>
<action name="actionShowVeginfoType2_3">
  <property name="text">
    <string>叶菜类</string>
  </property>
</action>
<action name="actionShowVeginfoType5">
  <property name="text">
    <string>菌类</string>
  </property>
</action>
<action name="actionSortByAreaR">
  <property name="text">
    <string>从大到小</string>
  </property>
```



```
</action>
<action name="actionSortByYearR">
  <property name="text">
    <string>从大到小</string>
  </property>
</action>
<action name="actionSortWeightR">
  <property name="text">
    <string>从大到小</string>
  </property>
</action>
<action name="actionSortDefault">
  <property name="text">
    <string>按编号排序种植信息</string>
  </property>
</action>
<action name="actionVegSortType">
  <property name="text">
    <string>按种类排序蔬菜信息</string>
  </property>
</action>
<action name="actionVegSortCode">
  <property name="text">
    <string>按编号排序蔬菜信息</string>
  </property>
</action>
<action name="actionVegSortLenth">
  <property name="text">
    <string>按营养成分长度排序</string>
  </property>
</action>
<action name="actionShowVegType">
  <property name="text">
    <string>蔬菜种类信息</string>
  </property>
</action>
<action name="actionAddVegType_2">
  <property name="text">
    <string>新增蔬菜种类信息</string>
  </property>
</action>
<action name="actionDeleteType">
  <property name="text">
    <string>删除蔬菜种类信息</string>
  </property>
</action>
<action name="actionEditTypeInfo">
  <property name="text">
    <string>修改蔬菜种类信息</string>
  </property>
</action>
<action name="actionSearchPlantByTypeFacts">
  <property name="text">
    <string>含分类及营养成分的种植信息</string>
  </property>
</action>
<action name="actionSearchPlantByYearName">
  <property name="text">
    <string>某年含菜名关键字的种植信息</string>
  </property>
</action>
```



```

    </property>
</action>
<action name="actionSearchVegInfoName">
    <property name="text">
        <string>蔬菜名关键字查找蔬菜信息</string>
    </property>
</action>
<action name="actionSearchVegInfoTypeFacts">
    <property name="text">
        <string>含分类及营养成分的蔬菜信息</string>
    </property>
</action>
<action name="actionShowPlantByYearArea">
    <property name="text">
        <string>按年份范围并以收成降序排序</string>
    </property>
</action>
<action name="actionShowPlantByAreaRange">
    <property name="text">
        <string>按面积范围查看种植信息</string>
    </property>
</action>
<action name="actionShowPlantByWeightRange">
    <property name="text">
        <string>按收成范围查看种植信息</string>
    </property>
</action>
<action name="actionCountVegPlantNum">
    <property name="text">
        <string>某种蔬菜已种植数量</string>
    </property>
</action>
<action name="actionloadFromBackup">
    <property name="text">
        <string>从备份文件读取所有蔬菜数据</string>
    </property>
    <property name="toolTip">
        <string>从备份文件读取数据</string>
    </property>
</action>
<action name="actionbackupData">
    <property name="text">
        <string>保存所有数据到备份文件</string>
    </property>
</action>
</widget>
<layoutdefault spacing="6" margin="11"/>
<resources/>
<connections/>
</ui>

```

### **./UI/notfound\_dialog.ui**

```

<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
    <class>NotFoundDialog</class>
    <widget class="QDialog" name="NotFoundDialog">
        <property name="geometry">

```



```
<rect>
  <x>0</x>
  <y>0</y>
  <width>300</width>
  <height>100</height>
</rect>
</property>
<property name="maximumSize">
  <size>
    <width>300</width>
    <height>100</height>
  </size>
</property>
<property name="windowTitle">
  <string>提示</string>
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>60</y>
      <width>231</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QLabel" name="label">
  <property name="geometry">
    <rect>
      <x>90</x>
      <y>20</y>
      <width>111</width>
      <height>21</height>
    </rect>
  </property>
  <property name="font">
    <font>
      <family>微软雅黑</family>
      <pointsize>10</pointsize>
    </font>
  </property>
  <property name="text">
    <string>无法找到该项</string>
  </property>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
```



```
<receiver>NotFoundDialog</receiver>
<slot>accept()</slot>
<hints>
  <hint type="sourcelabel">
    <x>248</x>
    <y>254</y>
  </hint>
  <hint type="destinationlabel">
    <x>157</x>
    <y>274</y>
  </hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>NotFoundDialog</receiver>
  <slot>reject()</slot>
  <hints>
    <hint type="sourcelabel">
      <x>316</x>
      <y>260</y>
    </hint>
    <hint type="destinationlabel">
      <x>286</x>
      <y>274</y>
    </hint>
  </hints>
</connection>
</connections>
</ui>
```

### **./UI/range\_input\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>RangeInputDialog</class>
  <widget class="QDialog" name="RangeInputDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>150</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>输入范围</string>
    </property>
  </widget>
</ui>
```



```
</property>
<widget class="QDialogButtonBox" name="buttonBox">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>110</y>
      <width>341</width>
      <height>32</height>
    </rect>
  </property>
  <property name="orientation">
    <enum>Qt::Horizontal</enum>
  </property>
  <property name="standardButtons">
    <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
  </property>
  <property name="centerButtons">
    <bool>true</bool>
  </property>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>10</y>
      <width>361</width>
      <height>91</height>
    </rect>
  </property>
  <layout class="HBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label">
        <property name="text">
          <string>TextLabel</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
    <item>
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>TextLabel</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit_2"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>RangeInputDialog</receiver>
    <slot>accept()</slot>
  </connection>
</connections>
<hints>
```



```
<hint type="sourcelabel">
  <x>248</x>
  <y>254</y>
</hint>
<hint type="destinationlabel">
  <x>157</x>
  <y>274</y>
</hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>RangeInputDialog</receiver>
  <slot>reject()</slot>
  <hints>
    <hint type="sourcelabel">
      <x>316</x>
      <y>260</y>
    </hint>
    <hint type="destinationlabel">
      <x>286</x>
      <y>274</y>
    </hint>
  </hints>
</connection>
</connections>
</ui>
```

### **./UI/search2\_input\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>Search2InputDialog</class>
  <widget class="QDialog" name="Search2InputDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>200</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>400</width>
        <height>200</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>400</width>
        <height>200</height>
      </size>
    </property>
    <property name="windowTitle">
      <string notr="true">搜索</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
```



```
<rect>
  <x>30</x>
  <y>150</y>
  <width>341</width>
  <height>32</height>
</rect>
</property>
<property name="orientation">
  <enum>Qt::Horizontal</enum>
</property>
<property name="standardButtons">
  <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
</property>
<property name="centerButtons">
  <bool>true</bool>
</property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>
      <x>20</x>
      <y>20</y>
      <width>361</width>
      <height>111</height>
    </rect>
  </property>
  <layout class="QGridLayout" name="gridLayout">
    <item row="0" column="1">
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
    <item row="1" column="1">
      <widget class="QLineEdit" name="lineEdit_2"/>
    </item>
    <item row="0" column="0">
      <widget class="QLabel" name="label">
        <property name="text">
          <string>TextLabel</string>
        </property>
      </widget>
    </item>
    <item row="1" column="0">
      <widget class="QLabel" name="label_2">
        <property name="text">
          <string>TextLabel</string>
        </property>
      </widget>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>Search2InputDialog</receiver>
    <slot>accept()</slot>
  </connection>
</connections>
<hints>
  <hint type="sourcelabel">
    <x>248</x>
    <y>254</y>
  </hint>
</hints>
```





```
</hint>
<hint type="destinationlabel">
  <x>157</x>
  <y>274</y>
</hint>
</hints>
</connection>
<connection>
  <sender>buttonBox</sender>
  <signal>rejected()</signal>
  <receiver>Search2InputDialog</receiver>
  <slot>reject()</slot>
  <hints>
    <hint type="sourcelabel">
      <x>316</x>
      <y>260</y>
    </hint>
    <hint type="destinationlabel">
      <x>286</x>
      <y>274</y>
    </hint>
  </hints>
</connection>
</connections>
</ui>
```

### **./UI/searchinput\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>SearchInputDialog</class>
  <widget class="QDialog" name="SearchInputDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>500</width>
        <height>150</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>500</width>
        <height>150</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>500</width>
        <height>150</height>
      </size>
    </property>
    <property name="windowTitle">
      <string notr="true">搜索</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>100</y>
```



```
<width>341</width>
<height>32</height>
</rect>
</property>
<property name="orientation">
  <enum>Qt::Horizontal</enum>
</property>
<property name="standardButtons">
  <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
</property>
<property name="centerButtons">
  <bool>true</bool>
</property>
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>10</y>
      <width>441</width>
      <height>80</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label">
        <property name="text">
          <string>输入</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>SearchInputDialog</receiver>
    <slot>accept()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>248</x>
        <y>254</y>
      </hint>
      <hint type="destinationlabel">
        <x>157</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
  <connection>
    <sender>buttonBox</sender>
    <signal>rejected()</signal>
    <receiver>SearchInputDialog</receiver>
    <slot>reject()</slot>
    <hints>
```



```
<hint type="sourcelabel">
  <x>316</x>
  <y>260</y>
</hint>
<hint type="destinationlabel">
  <x>286</x>
  <y>274</y>
</hint>
</hints>
</connection>
</connections>
</ui>
```

### **./UI/vegcode\_input\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>VegCodeInputDialog</class>
  <widget class="QDialog" name="VegCodeInputDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>400</width>
        <height>150</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>400</width>
        <height>150</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>输入蔬菜编号</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>30</x>
          <y>100</y>
          <width>341</width>
          <height>32</height>
        </rect>
      </property>
      <property name="orientation">
        <enum>Qt::Horizontal</enum>
      </property>
      <property name="standardButtons">
        <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
      </property>
      <property name="centerButtons">
        <bool>true</bool>
      </property>
```



```
</widget>
<widget class="QWidget" name="horizontalLayoutWidget">
  <property name="geometry">
    <rect>
      <x>30</x>
      <y>10</y>
      <width>341</width>
      <height>80</height>
    </rect>
  </property>
  <layout class="QHBoxLayout" name="horizontalLayout">
    <item>
      <widget class="QLabel" name="label">
        <property name="text">
          <string>蔬菜编号</string>
        </property>
      </widget>
    </item>
    <item>
      <widget class="QLineEdit" name="lineEdit"/>
    </item>
  </layout>
</widget>
</widget>
<resources/>
<connections>
  <connection>
    <sender>buttonBox</sender>
    <signal>accepted()</signal>
    <receiver>VegCodeInputDialog</receiver>
    <slot>accept()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>248</x>
        <y>254</y>
      </hint>
      <hint type="destinationlabel">
        <x>157</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
  <connection>
    <sender>buttonBox</sender>
    <signal>rejected()</signal>
    <receiver>VegCodeInputDialog</receiver>
    <slot>reject()</slot>
    <hints>
      <hint type="sourcelabel">
        <x>316</x>
        <y>260</y>
      </hint>
      <hint type="destinationlabel">
        <x>286</x>
        <y>274</y>
      </hint>
    </hints>
  </connection>
</connections>
</ui>
```

**./UI/yearinput\_dialog.ui**

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>YearInputDialog</class>
  <widget class="QDialog" name="YearInputDialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>300</width>
        <height>100</height>
      </rect>
    </property>
    <property name="minimumSize">
      <size>
        <width>300</width>
        <height>100</height>
      </size>
    </property>
    <property name="maximumSize">
      <size>
        <width>300</width>
        <height>100</height>
      </size>
    </property>
    <property name="windowTitle">
      <string>输入年份</string>
    </property>
    <widget class="QDialogButtonBox" name="buttonBox">
      <property name="geometry">
        <rect>
          <x>40</x>
          <y>60</y>
          <width>201</width>
          <height>32</height>
        </rect>
      </property>
      <property name="orientation">
        <enum>Qt::Horizontal</enum>
      </property>
      <property name="standardButtons">
        <set>QDialogButtonBox::Cancel|QDialogButtonBox::Ok</set>
      </property>
      <property name="centerButtons">
        <bool>true</bool>
      </property>
    </widget>
    <widget class="QWidget" name="horizontalLayoutWidget">
      <property name="geometry">
        <rect>
          <x>20</x>
          <y>0</y>
          <width>241</width>
          <height>61</height>
        </rect>
      </property>
      <layout class="QHBoxLayout" name="horizontalLayout">
        <item>
          <widget class="QLabel" name="label">
```



```

        <property name="text">
            <string>输入年份</string>
        </property>
    </widget>
</item>
<item>
    <widget class="QLineEdit" name="lineEdit"/>
</item>
</layout>
</widget>
</widget>
<resources/>
<connections>
    <connection>
        <sender>buttonBox</sender>
        <signal>accepted()</signal>
        <receiver>YearInputDialog</receiver>
        <slot>accept()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>248</x>
                <y>254</y>
            </hint>
            <hint type="destinationlabel">
                <x>157</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
    <connection>
        <sender>buttonBox</sender>
        <signal>rejected()</signal>
        <receiver>YearInputDialog</receiver>
        <slot>reject()</slot>
        <hints>
            <hint type="sourcelabel">
                <x>316</x>
                <y>260</y>
            </hint>
            <hint type="destinationlabel">
                <x>286</x>
                <y>274</y>
            </hint>
        </hints>
    </connection>
</connections>
</ui>

```

### ./database/typeinfo.info

```

{'1','根茎类'}
{'2','果菜类'}
{'3','瓜类'}
{'4','叶菜类'}
{'5','菌类'}

```

### ./database/veginfo.info

```

{'1','白萝卜','1','植物蛋白、维生素 C 和叶酸等'}
{'2','胡萝卜','1','植物蛋白、维生素 C 和叶酸等'}

```



```
{'3','葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'4','大葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'5','小葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'6','蒜','1','蛋白质、脂肪、硫胺素、尼克酸、维生素 E 等'}
{'7','洋葱','1','胡萝卜素、硫胺素、核黄素、维生素 C 等'}
{'8','莴笋','1','胡萝卜素、维生素 B、钙、铁等'}
{'9','山药','1','薯蓣皂苷、维生素 B、维生素 C、维生素 E 等'}
{'10','马铃薯','1','钙、磷、铁、钾、碘、胡萝卜素、硫胺素、核黄素、尼克酸、维生素 C 等'}
{'11','红薯','1','胡萝卜素、钾、去氧表雄酮等'}
{'12','菜椒','2','胡萝卜素、钙、铁等'}
{'13','青椒','2','胡萝卜素、钙、铁等'}
{'14','尖椒','2','胡萝卜素、钙、铁等'}
{'15','甜椒','2','胡萝卜素、钙、铁等'}
{'16','朝天椒','2','胡萝卜素、钙、铁等'}
{'17','线椒','2','胡萝卜素、钙、铁等'}
{'18','南瓜','2','类胡萝卜素、各类氨基酸、维生素 C 等'}
{'19','丝瓜','2','钙、磷、铁及维生素 B1、维生素 C、皂甙、木糖胶、瓜氨酸等'}
{'20','扁豆','2','胡萝卜素、维生素 C、钾、磷、铁、锌等'}
{'21','西瓜','3','苹果酸、果糖、蛋白氨基酸、番茄素、维生素 C 等'}
{'22','甜瓜','3','胡萝卜素、维生素 B1、维生素 B2、烟酸、钙、磷、铁等'}
{'23','白瓜','3','核黄素、维生素 A、维生素 C、胆固醇、视黄醇当量、胡萝卜素等'}
{'24','黄瓜','3','维生素 B2、维生素 C、维生素 E、胡萝卜素、尼克酸、钙、磷、铁等'}
{'25','苦瓜','3','钾、钠、磷、胡萝卜素、尼克酸、硫胺素、抗坏血酸等'}
{'26','大白菜','4','多种维生素及钙、磷、铁等'}
{'27','小白菜','4','多种维生素及钙、磷、铁等'}
{'28','生菜','4','多种维生素及钙、磷、铁等'}
{'29','菠菜','4','类胡萝卜素、维生素 C、维生素 K、矿物质（钙质、铁质等）、辅酶 Q10 等'}
{'30','韭菜','4','维生素 C、维生素 B1、维生素 B2、尼克酸、胡萝卜素、碳水化合物及矿物质'}
{'31','芹菜','4','维生素 A、维生素 B1、维生素 B2、维生素 C 和维生素 P，钙，铁，磷等矿物质'}
{'32','空心菜','4','维生素 A 原、B 族维生素、维生素 C 及烟酸和蛋白质、脂肪、钙、磷、铁等'}
{'33','木耳','5','钙、磷、铁、胡萝卜素、维生素 B1、维生素 B2、烟酸、磷脂、固醇等'}
{'34','银耳','5','蛋白质、碳水化合物、脂肪、粗纤维、无机盐、少量维生素 B 类等'}
{'35','平菇','5','各种氨基酸、矿物质、B 族维生素等'}
{'36','草菇','5','各种氨基酸、维生素 C、灰分等'}
{'37','金针菇','5','赖氨酸和精氨酸等多种氨基酸、锌、朴菇素等'}
{'38','香菇','5','各类氨基酸、维生素、矿物质'}
```

#### ./database/plantinfo.info

```
{'1','1','7','22.540','2015'}
{'2','21','1','2.600','2010'}
{'3','31','21','2.800','2017'}
{'4','4','21','23.150','2013'}
{'5','15','21','23.660','2018'}
{'6','26','1','5.763','2019'}
{'7','37','20','12.560','2017'}
{'8','18','1','2.500','2010'}
{'9','10','21','2.800','2017'}
```



```
{'10','13','12','2.800','1029'}
{'11','32','21','23.660','2018'}
{'12','13','1','5.763','2019'}
{'13','34','20','12.560','2017'}
{'14','24','1','2.500','2010'}
{'15','16','1','2.600','2010'}
{'16','23','21','2.800','2017'}
{'17','18','12','2.800','1029'}
{'18','19','21','23.660','2018'}
{'19','35','1','5.763','2019'}
{'20','21','20','12.560','2017'}
{'21','3','3','100.200','2018'}
{'22','30','15','63.273','2018'}
{'23','37','32','233.230','2017'}
{'24','29','20','321.100','2014'}
{'25','20','15','200.300','2015'}
{'26','22','10','30.000','2017'}
{'27','2','10','30.200','2017'}
{'28','3','3','2.310','2017'}
{'29','5','10','2.173','2018'}
{'30','18','6','30.320','2016'}
{'31','20','5','15.231','2015'}
{'32','14','21','70.000','2014'}
{'33','15','9','24.231','2008'}
{'34','21','12','2.350','2017'}
{'35','1','23','30.250','2015'}
{'36','20','23','100.100','2007'}
{'37','35','23','15.230','2011'}
{'38','14','35','100.540','2013'}
{'39','27','20','13.560','2017'}
{'40','13','1','2.500','2010'}
{'41','12','21','12.800','2017'}
{'42','33','22','16.800','2009'}
{'43','32','21','23.660','2018'}
{'44','13','12','15.763','2019'}
{'45','34','20','12.560','2017'}
{'46','14','12','32.500','2010'}
{'47','36','35','22.600','2010'}
{'48','33','21','2.800','2017'}
{'49','35','23','56.763','2014'}
{'50','21','30','52.560','2012'}
{'51','23','2','2.333','2017'}
{'52','90','12','2.413','2008'}
```

### ./database/backupFile.bak

```
#
{'1','根茎类'}
{'2','果菜类'}
{'3','瓜类'}
{'4','叶菜类'}
{'5','菌类'}
#
{'1','白萝卜','1','植物蛋白、维生素 c 和叶酸等'}
{'2','胡萝卜','1','植物蛋白、维生素 c 和叶酸等'}
{'3','葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'4','大葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'5','小葱','1','蛋白质、碳水化合物等多种维生素及矿物质等'}
{'6','蒜','1','蛋白质、脂肪、硫胺素、尼克酸、维生素 e 等'}
```





```

{'7','洋葱','1','胡萝卜素、硫胺素、核黄素、维生素 C 等'}
{'8','莴笋','1','胡萝卜素、维生素 B、钙、铁等'}
{'9','山药','1','薯蓣皂苷、维生素 B、维生素 C、维生素 E 等'}
{'10','马铃薯','1','钙、磷、铁、钾、碘、胡萝卜素、硫胺素、核黄素、尼克酸、维生素 C 等'}
{'11','红薯','1','胡萝卜素、钾、去氧表雄酮等'}
{'12','菜椒','2','胡萝卜素、钙、铁等'}
{'13','青椒','2','胡萝卜素、钙、铁等'}
{'14','尖椒','2','胡萝卜素、钙、铁等'}
{'15','甜椒','2','胡萝卜素、钙、铁等'}
{'16','朝天椒','2','胡萝卜素、钙、铁等'}
{'17','线椒','2','胡萝卜素、钙、铁等'}
{'18','南瓜','2','类胡萝卜素、各类氨基酸、维生素 C 等'}
{'19','丝瓜','2','钙、磷、铁及维生素 B1、维生素 C、皂甙、木糖胶、瓜氨酸等'}
{'20','扁豆','2','胡萝卜素、维生素 C、钾、磷、铁、锌等'}
{'21','西瓜','3','苹果酸、果糖、蛋白氨基酸、番茄素、维生素 C 等'}
{'22','甜瓜','3','胡萝卜素、维生素 B1、维生素 B2、烟酸、钙、磷、铁等'}
{'23','白瓜','3','核黄素、维生素 A、维生素 C、胆固醇、视黄醇当量、胡萝卜素等'}
{'24','黄瓜','3','维生素 B2、维生素 C、维生素 E、胡萝卜素、尼克酸、钙、磷、铁等'}
{'25','苦瓜','3','钾、钠、磷、胡萝卜素、尼克酸、硫胺素、抗坏血酸等'}
{'26','大白菜','4','多种维生素及钙、磷、铁等'}
{'27','小白菜','4','多种维生素及钙、磷、铁等'}
{'28','生菜','4','多种维生素及钙、磷、铁等'}
{'29','菠菜','4','类胡萝卜素、维生素 C、维生素 K、矿物质（钙质、铁质等）、辅酶 Q10 等'}
{'30','韭菜','4','维生素 C、维生素 B1、维生素 B2、尼克酸、胡萝卜素、碳水化合物及矿物质'}
{'31','芹菜','4','维生素 A、维生素 B1、维生素 B2、维生素 C 和维生素 P，钙，铁，磷等矿物质'}
{'32','空心菜','4','维生素 A 原、B 族维生素、维生素 C 及烟酸和蛋白质、脂肪、钙、磷、铁等'}
{'33','木耳','5','钙、磷、铁、胡萝卜素、维生素 B1、维生素 B2、烟酸、磷脂、固醇等'}
{'34','银耳','5','蛋白质、碳水化合物、脂肪、粗纤维、无机盐、少量维生素 B 类等'}
{'35','平菇','5','各种氨基酸、矿物质、B 族维生素等'}
{'36','草菇','5','各种氨基酸、维生素 C、灰分等'}
{'37','金针菇','5','赖氨酸和精氨酸等多种氨基酸、锌、朴菇素等'}
{'38','香菇','5','各类氨基酸、维生素、矿物质'}

#
{'1','1','7','22.540','2015'}
{'2','21','1','2.600','2010'}
{'3','31','21','2.800','2017'}
{'4','4','21','23.150','2013'}
{'5','15','21','23.660','2018'}
{'6','26','1','5.763','2019'}
{'7','37','20','12.560','2017'}
{'8','18','1','2.500','2010'}
{'9','10','21','2.800','2017'}
{'10','13','12','2.800','1029'}
{'11','32','21','23.660','2018'}
{'12','13','1','5.763','2019'}
{'13','34','20','12.560','2017'}
{'14','24','1','2.500','2010'}

```



```
{'15','16','1','2.600','2010'}
{'16','23','21','2.800','2017'}
{'17','18','12','2.800','1029'}
{'18','19','21','23.660','2018'}
{'19','35','1','5.763','2019'}
{'20','21','20','12.560','2017'}
{'21','3','3','100.200','2018'}
{'22','30','15','63.273','2018'}
{'23','37','32','201.500','2016'}
{'24','29','20','321.100','2014'}
{'25','20','15','200.300','2015'}
{'26','22','10','30.000','2017'}
{'27','2','10','30.200','2017'}
{'28','3','3','2.310','2017'}
{'29','5','10','2.173','2018'}
{'30','18','6','30.320','2016'}
{'31','20','5','15.231','2015'}
{'32','14','21','70.000','2014'}
{'33','15','9','24.231','2008'}
{'34','21','12','2.350','2017'}
{'35','1','23','30.250','2015'}
{'36','20','23','100.100','2007'}
{'37','35','23','15.230','2011'}
{'38','14','35','100.540','2013'}
{'39','27','20','13.560','2017'}
{'40','13','1','2.500','2010'}
{'41','12','21','12.800','2017'}
{'42','33','22','16.800','2009'}
{'43','32','21','23.660','2018'}
{'44','13','12','15.763','2019'}
{'45','34','20','12.560','2017'}
{'46','14','12','32.500','2010'}
{'47','36','35','22.600','2010'}
{'48','33','21','2.800','2017'}
{'49','38','12','32.800','2009'}
{'50','19','21','23.660','2011'}
{'51','35','23','56.763','2014'}
{'52','21','30','52.560','2012'}
```

## II. 项目源码上传地址

本项目已 push 到 github，所有源码可在以下链接看到：

[https://github.com/hunto/farm\\_manager](https://github.com/hunto/farm_manager)