

使用 C++ 实现植物检索表辅助查询系统

徐 热

兰州职业技术学院 甘肃兰州

【摘要】判断植物种类是农学、生态学、植物学专业人员的必备能力之一。在实地考察的过程当中，涉及植物的识别问题时，常用植物检索表来进行识别，但多有不便。因此可尝试采用面向对象的程序设计思路，用 C++ 语言编写植物辅助识别系统，实现查询植物主要特征和植物检索表功能，同时便于更新和维护。

【关键词】C++ 语言；植物检索表

Programming Auxiliary Query System for Plant Retrieval List By C++ Language

Re Xu

Lanzhou Vocational Technical College

【Abstract】Judging plant species is one of the necessary abilities for agronomy, ecology and botany professionals. In the process of field investigation, when involving the identification of plants, plant retrieval tables are commonly used to identify, but there are many inconveniences. Therefore, we can try to use C++ language to write plant auxiliary recognition system on object-oriented programming ideas to query the main features of plants and plant search function, and easy to update and maintain.

【Keywords】C++ Language; Plant Retrieval List

目前市面上的很多软件有 AI 植物识别功能，该功能针对一般生活上的使用已经足够，但用作学习研究时，难免有不精确的情况出现。在专业学习研究中使用植物检索表居多，但植物检索表缺乏使用指导，又存在不易携带，不易查询的特点。生物、农学等相关专业从业者有一定的专业知识基础，故面向专业从业者制作的程序可以涉及专业知识，但使用必须有引导，于是我们想到可以将植物检索表编写成程序，以满足简单的植物学学习需要。同时，由于用户群体对相关知识信息有一定了解，我们甚至可以引导用户自行编写植物信息库。

1 程序设计思路与准备

在设计程序之前，我们先对实际情况进行分析：

1.1 植物检索表的程序实现可能

植物检索表是植物分离中识别和鉴定植物不可缺少的工具，是根据法国拉马克二歧分类原则，将一群植物相对的特征，特性分成对应的两个分支，直到查询到植物检索表的终点为止。由以上定义可知，植物检索表形式规整简洁，其将植物的主要特

征分为两类，可用计算机语言的 0 和 1 代替。但传统植物检索表对植物特征的描述多使用文字语言，在编写程序过程中多有不便。为解决这一问题，我们将同一植物类群中不同的植物特征编为不同的编号，将编号按顺序组成数组（下称植物编号组），每个数组代表一种植物，以此进行存储，便于查询。之后，将每一个编号代表的特征写入文本文档中（下称对应特征），需要时再进行读取。由此可将植物检索表编写为程序使用，相比人工手动查询可明显提高效率。

1.2 程序基本功能

由于编写完整的植物检索表工作量太过巨大，根据实际情况，我们决定选取科（Family）这一层次开始，以常见的桑科植物和松科植物为代表，确定了程序应当实现的基本功能：

（1）实现植物查找功能：在按照指引输入相关编号后，能输出相应植物特征信息。

（2）实现植物检索表功能：在实地使用时，按照指引对相关特征进行判断，并根据判断结果输出

植物属名。为实现相应基本功能，经过讨论，编写程序过程中，有如下注意事项：

(3) 将桑科和松科植物检索表按特征编号，将生成的植物编号组和对应特征存储在 txt 文件中便于读取。

(4) 为便于使用文件流读取，txt 文件中的文字应当有严格统一而又简洁的排版格式。

(5) 编写程序的过程中应当考虑到更新的便利性，保证只用通过新增或修改 txt 文件即可完成扩充或修改植物检索表的工作，而不用修改程序代码。

2 程序主要功能

程序主要实现功能如下：

(1) 植物辅助识别系统：向用户给出提示指导，

引导用户对植物进行识别，并给出识别结果。

(2) 植物信息查询系统：按照用户要求将现有植物信息直接输出。

程序主要层次结构如下图 2 所示：

由于编写完整的植物检索表太过复杂，信息存储文件需要长期更新，同时为满足用户自行对数据库进行编写和更新的要求，程序同时还应当有便于维护的特点，在对数据进行更新的过程中甚至不用修改代码。

3 程序算法设计与结构

在以上程序设计思路下，该程序算法流程图如下图 3 所示：

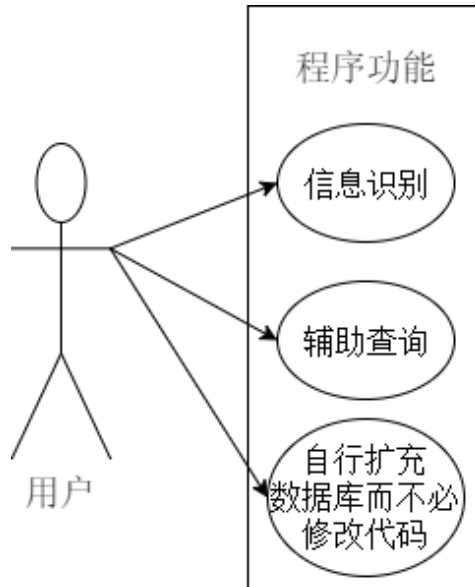


图 1 用户功能需求分析

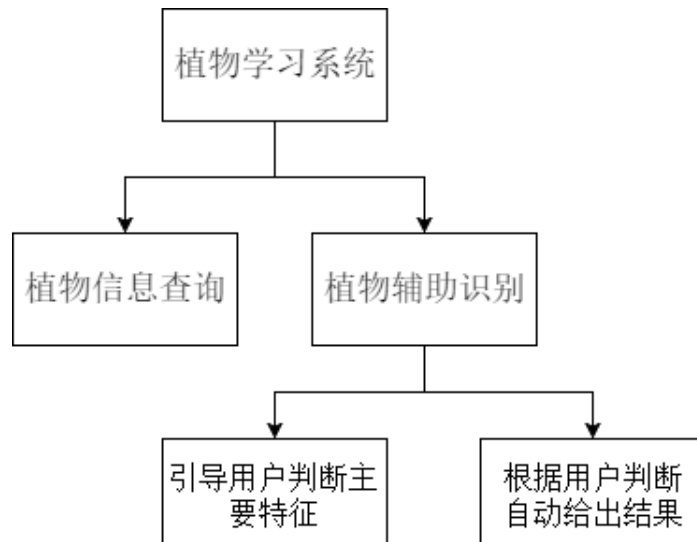


图 2 程序主要结构

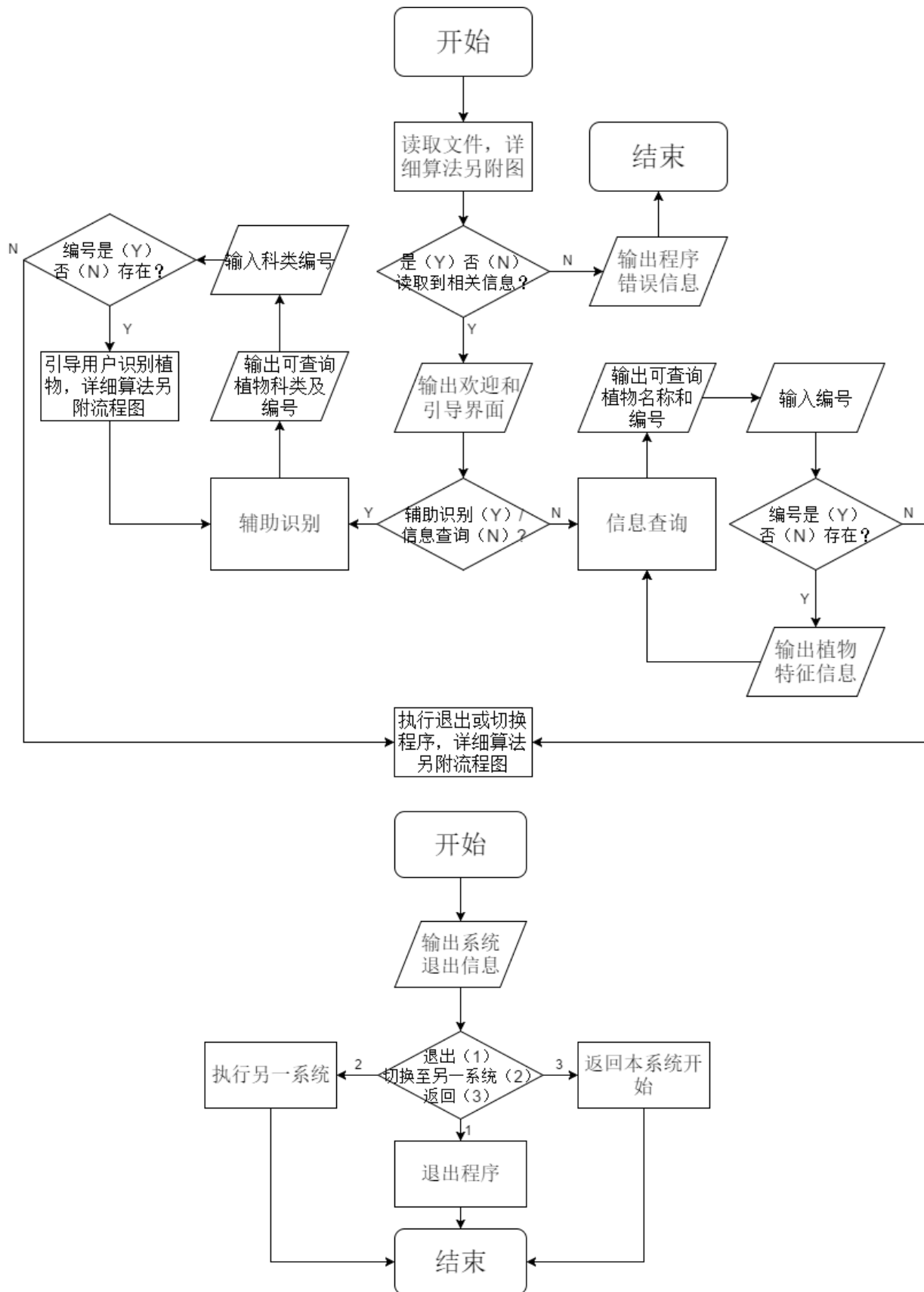


图3 主程序算法 (上) 与退出算法 (下)

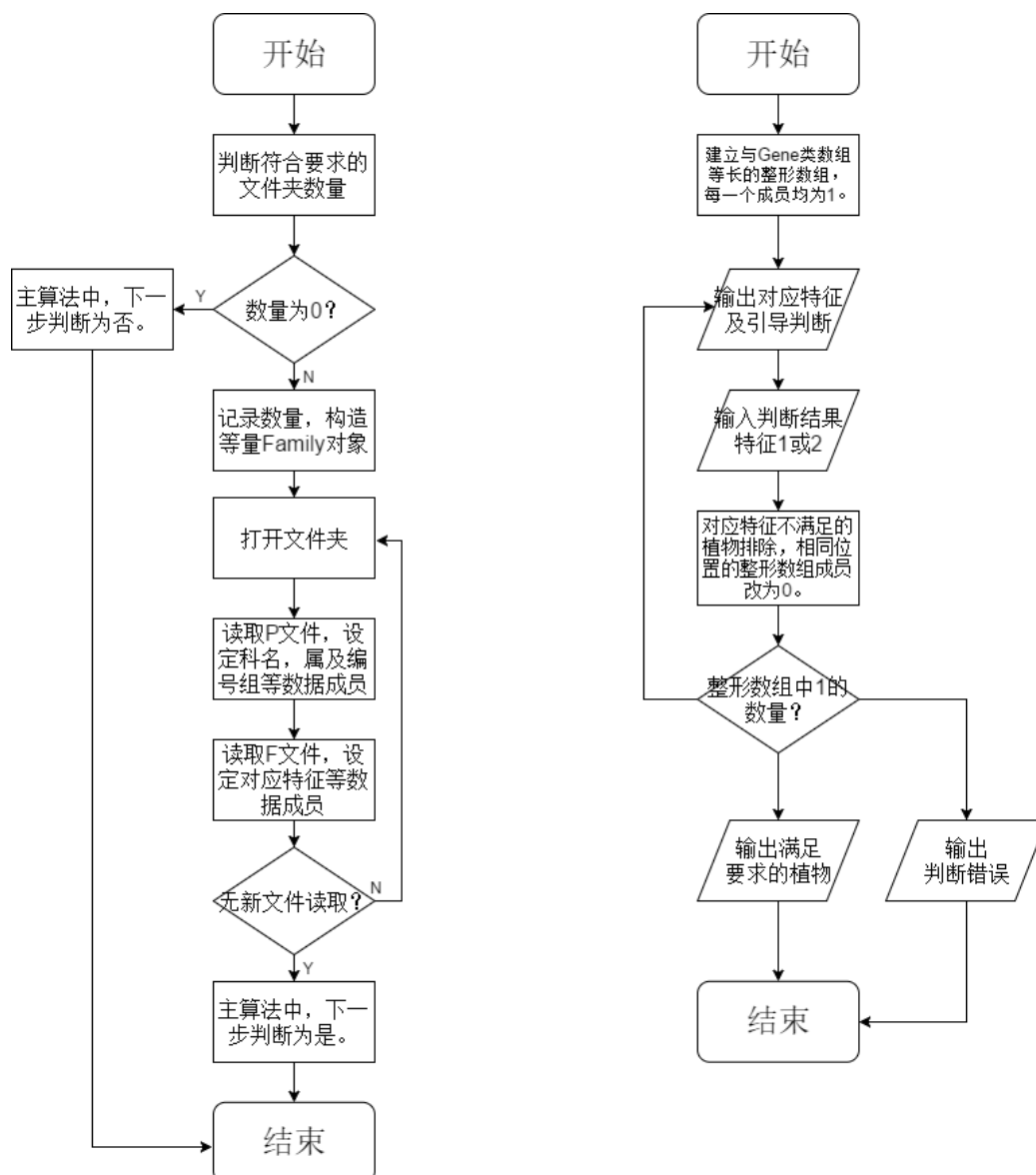


图4 文件读取算法（左） 植物辅助识别系统算法（右）

表1 文件格式

文件类型	功能	存储及命名格式	txt 排版格式
文件夹	储存 txt 文件。其文件名起编号作用。	按顺序命名为 G0, G1, ……，编号超过 10 后编号位数会适当增加。	无
对应特征文件	储存编号对应特征，内容读取后将构造 Plantfeature 类数组。	每个文件夹里只有一个，全部命名为 F.txt。	每一行全部为：特征名:特征 1/特征 2。 最后一行为空。
编号组文件	储存植物特征编号，内容读取后将构造 Gene 类数组。	每个文件夹中只有一个，全部命名为 P.txt	第一行为科名，第二行及以后全部为：属名:000000（编号）。 最后一行为空。

程序主要结构如下：

(1) “Family”类，代表科。在该类中实现程序的主要功能：读取植物编号组文件（下称 P 文件）和对应特征文件（下称 F 文件），并将它们保存在

自己的对象成员中，实现处理并输出相关信息的功能。

(2) “Gene”类，代表属。实现储存属名和植物编号组的功能。同属于一个科的属存入一个 Gene

类数组中，该数组作为“Family”类的对象成员。

(3) “Plantfeature”类，代表对应特征。每个对应特征由特征名，特征 1 和特征 2 三个成员组成。属于一个科的所有对应特征存入一个 Plantfeature 类数组中，该数组作为“Family”类的对象成员。

(4) 主程序：实现读取文件，引导辅助识别和信息查询，退出程序，检测程序运行状态等其他全部主要功能。为便于使用文件流读取，txt 文件中的文字及 txt 文件的保存路径应当有严格统一而又简洁的格式，在以后的更新和维护当中也应当严格按照格式执行。格式如下表 1 所示：

4 程序编写过程中出现的问题

虽然有充分的前期准备，但在程序的实际编写过程中，仍然出现了不少问题。以下选取部分有代表性的问题进行分析。

4.1 数组的浅拷贝问题

程序的编写过程当中，需要将 txt 文件中读取的属名和植物编号组，通过“Gene”类的成员函数“SetNumber(char* number)”“SetName(char* name)”为相应数据成员赋值。同时由于读取的属名和植物编号组数目不定，程序中采用了动态数组作为缓冲区，反复定义和释放进行读取。简单使用赋值号将导致浅拷贝问题，在缓冲区释放时将会同时释放。为解决浅拷贝问题，我们必须实现深拷贝。

数组的深拷贝与类的深拷贝有一定区别。查阅资料发现，数组的深拷贝有四种方式：for 数组循环拷贝，clone()函数拷贝，System.arraycopy()函数拷贝和 Arrays.copyOf()拷贝。最终本程序采用了 for 循环拷贝。

for 循环优点是编写程序时相对简便，原理清晰。由于其它方式都需要定义相应的头文件或实现与 Java 语言的兼容，难度较大，故在程序编写时不采用。但 for 循环并非是最优解，最主要的缺点在于，在时间复杂度上该方法在四个方法中属于最慢者。且由于涉及汉字的录入，在代码完成时会提出警告。但程序测试过程中未发现明显的问题。

代码如下：

```
void Gene::Setnumber(char* number)
{
    int i;
    Gene_number = new char[strlen(number)];
```

```
for (i = 0; i <= strlen(number); i++)
    Gene_number[i] = number[i];
}
```

4.2 文件的遍历问题

为实现最初设想时提出的“便于维护”特点，在程序编写过程中，我们为储存 txt 文件的格式提出了规定，将文件夹按顺序编号命名，txt 文件命名固定。在程序运行过程中会自动增加编号，确保能按顺序读取相应的文件。txt 文件内容格式固定，确保能使用 getline 函数进行标准读取并存储。但运行过程中还是发现了一定问题。由于文件命名格式固定，字符长度固定，故文件夹编号只能存储“0~9”共 10 个，程序能读取的最大文件数量有限。目前只完成了有代表性的两个科，故该问题不明显。但在实际使用中，随着科的数量增加，编号可能会出现溢出的情况。虽然 C++语言提供了返回路径下文件夹数量的功能，但该功能的实现相对较复杂，在此情况下不适用。该问题无法使用简单的代码得以解决，只能在科的数量超过 10 时，对所有文件的编号进行扩充，由 1 位编号扩展为 2 位，3 位，并通过循环嵌套结构才能得以解决。

4.3 动态数组空间释放问题

由于该程序结构特殊，在程序设计的过程中，我们使用了大量的动态数组。由于数组长度的不确定性，分布的内存空间往往较大，在程序使用过程中如不能及时释放，将会造成极大的内存泄露。为防止内存泄露，我们在程序设计的过程中，必须设计合理的程序退出机制，并在退出过程中释放所有动态内存空间。由于多数数组是类的数据成员，我们选择在析构函数中释放所有的内存空间，并在退出时执行所有的析构函数。但如果在程序运行过程中强制中断程序，仍然会造成内存泄露问题，该问题仅凭简单代码无法得到解决。我们只能在程序设计过程中增加提示，希望用户能通过退出通道退出程序。

4.4 植物识别系统编写问题

本程序基本功能之一是植物辅助识别系统。即根据程序引导判断特征并输入，程序自动判断植物类型或要求进一步判断。该系统代码较为复杂。初期设想过程中我组打算借助现有的二叉树或线性表结构编写，但由于结构复杂，调试失败。经过讨论，

我组重新设计了算法。决定将植物识别系统的功能函数放入 Family 类中。在该类中动态定义与 Gene 类数组等长的 int 型数组,使之在逻辑上存在一一对应的关系。在该数组的每一个存储空间中存入数字“1”。在每一次判断植物特征时,将不符合要求的

植物对应的数字由“1”改为“0”。每次判断结束后进行检测,当检测到只剩下一个“1”时,将对应的植物属名输出。当检测到没有“1”时,提示用户存在判断错误。该算法相比传统结构简单,良好地实现预期功能,解决了相关问题。

代码如下:

```
void Family::Judgegene()
{
    int* select;
    select = new int[gene_number];
    int m = 0;
    int n = 0;
    int x = 0;
    int p = 0;
    int q = gene_number;
    int k = 0;
    for (x = 0; x < gene_number; x++)
        select[x] = 1;
    cout << "请根据提示进行识别, 您所见的植物: " << endl;
    for (n = 0; n < feature_number; n++)
    {
        this->FeatureOutput(cout, n);
        cin >> p;
        if (p == 1)
        {
            for (x = 0; x < gene_number; x++)
            {
                if (select[x] != 0 && gene[x].Getnumber(n) != '0')
                    select[x] = 0;
            }
        }
        if (p == 2)
        {
            for (x = 0; x < gene_number; x++)
            {
                if (select[x] != 0 && gene[x].Getnumber(n) != '1')
                    select[x] = 0;
            }
        }
    }
    m = 0;
```

```

    for (x = 0; x < gene_number; x++)
    {
        if (select[x] != 0)
            m++;
    }
    if (m == 1)
        break;
}
for (x = 0; x < gene_number; x++)
{
    if (select[x] != 0)
        cout << "根据识别, 您所见的植物应该是: " << gene[x].Getname() << "植物。" << endl;
    system("pause");
    delete[] select;
    return;
}
}
}

```

5 程序最终实现情况

本程序在编写过程中, 使用了类的对象成员, 浅拷贝与深拷贝, 文件流读取, 动态分配数组等简单的结构和功能, 基本实现了设想的要求。但在易用性和可维护性上仍然有改进的空间。

参考文献

- [1] 赵宏.程序设计基础[M].清华大学出版社,北京.2019
- [2] 中国科学院中国植物志编辑委员会.中国植物志[M].科学出版社,北京.2004
- [3] 高丽彬,金云,周国飞.利用植物检索表的方式对教学中几种常见易混淆植物的区分[J].畜牧与饲料科学,2015,36(02):82-85.
- [4] 陈艳,许丽娟,文晓棠.算法设计与分析课程教学改革探索[J].计算机时代,2022(07):136-140
- [5] 杨宏伟,张云.计算机视觉技术在植物学中的应用[J].生物信息学,2007(01):41-43+48.
- [6] 胡刚,周平,周刚,张忠华.运用植物识别类微信小程序辅助

植物学教学的探索与思考[J].教育现代化,2020,7(06):112-114.

- [7] 彭召意,赵菁菁,刘建国.C++中类的构造函数探究和使用[J].企业科技与发展,2016(10):50-53.
- [8] 郭艾寅.浅析 C++中类的拷贝[J].科技信息(学术研究),2007(12):156+158.

收稿日期: 2022 年 8 月 19 日

出刊日期: 2022 年 9 月 7 日

引用本文: 徐热, 使用 C++实现植物检索表辅助查询系统[J]. 国际计算机科学进展, 2022, 2(2): 34-40.
DOI: 10.12208/j. aics.20220020

检索信息: RCCSE 权威核心学术期刊数据库、中国知网 (CNKI Scholar)、万方数据 (WANFANG DATA)、Google Scholar 等数据库收录期刊

版权声明: ©2022 作者与开放获取期刊研究中心 (OAJRC) 所有。本文章按照知识共享署名许可条款发表。<http://creativecommons.org/licenses/by/4.0/>



OPEN ACCESS