

**滨江学院**

**程序设计实训课程论文**

**题 目** 仓库管理系统（小超市）

**院 系 物联网工程学院**

**专 业计算机科学与技术**

**学生姓名 刘旅诚**

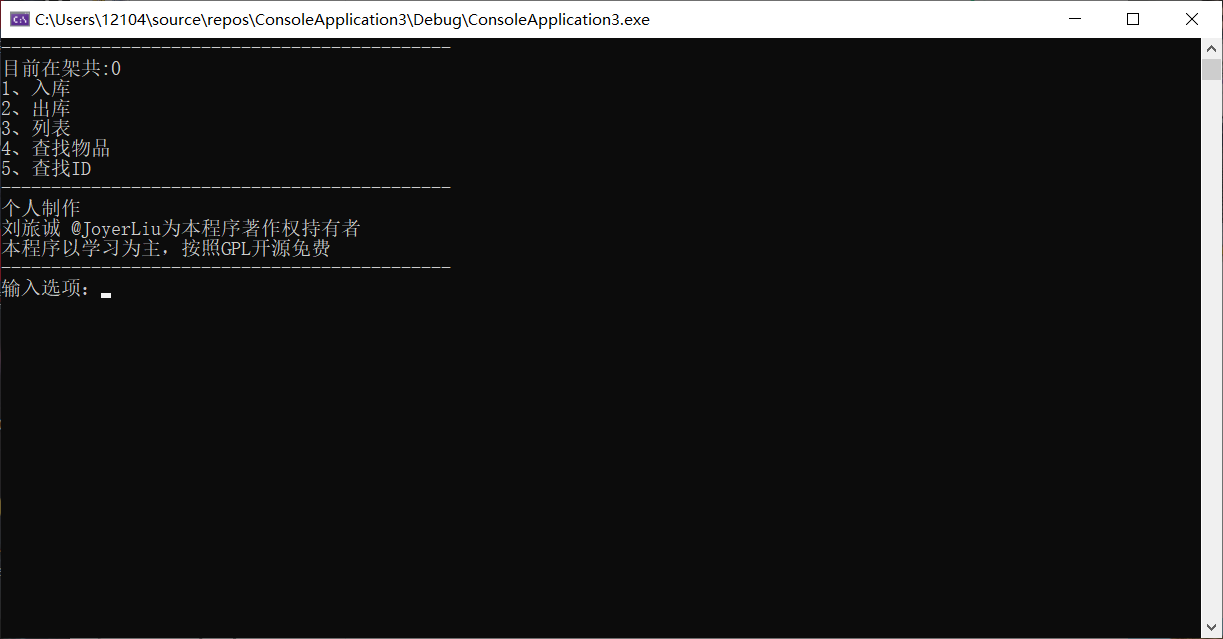
**学 号 20192341012**

**二Ｏ二一 年 六 月 九 日**

仓库管理系统（小超市）MIB

刘旅诚

主界面



目录

[1 引言 1](#_Toc75685903)

[2 系统设计 1](#_Toc75685904)

[系统结构图 1](#_Toc75685905)

[系统流程图 2](#_Toc75685906)

[API(MIB.h) 3](#_Toc75685907)

[3 系统实现 3](#_Toc75685908)

[源码 6](#_Toc75685909)

[MIB.h 6](#_Toc75685910)

[Main.cpp 8](#_Toc75685911)

[4 总结 10](#_Toc75685912)

# 1 引言

设计此程序（仓库管理系统，以下简称仓管）为了促进掌握C/C++的基础语法知识，提高阅读、编写C/C++程序的能力，掌握用C语言解决具体问题的一般方法和步骤，并在调试程序的过程中，提高解决问题的能力，为进一步学习其他语言打好基础。

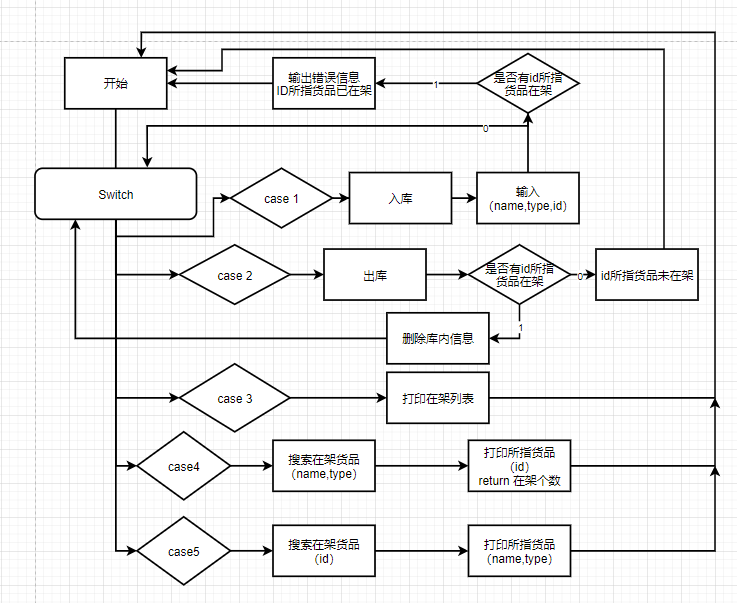
此程序的需求如下：

1. 建立严格的仓管系统，严格管理物资的出入库。
2. 以名字、类型/品类、唯一识别码（name,type,id）作为每样在架货品的识别方式，并予以输入。
3. 以唯一识别码（id）为凭据完成出库。
4. 输出在架货品列表。
5. 以 名字、类型/品类 （name,type）搜索在架货品，并返回在架数量与在架货品的唯一码列表。
6. 以唯一识别码（id）搜索在架货品，并返回此在架货品的相关信息。

# 2 系统设计

## 系统结构图

## 系统流程图

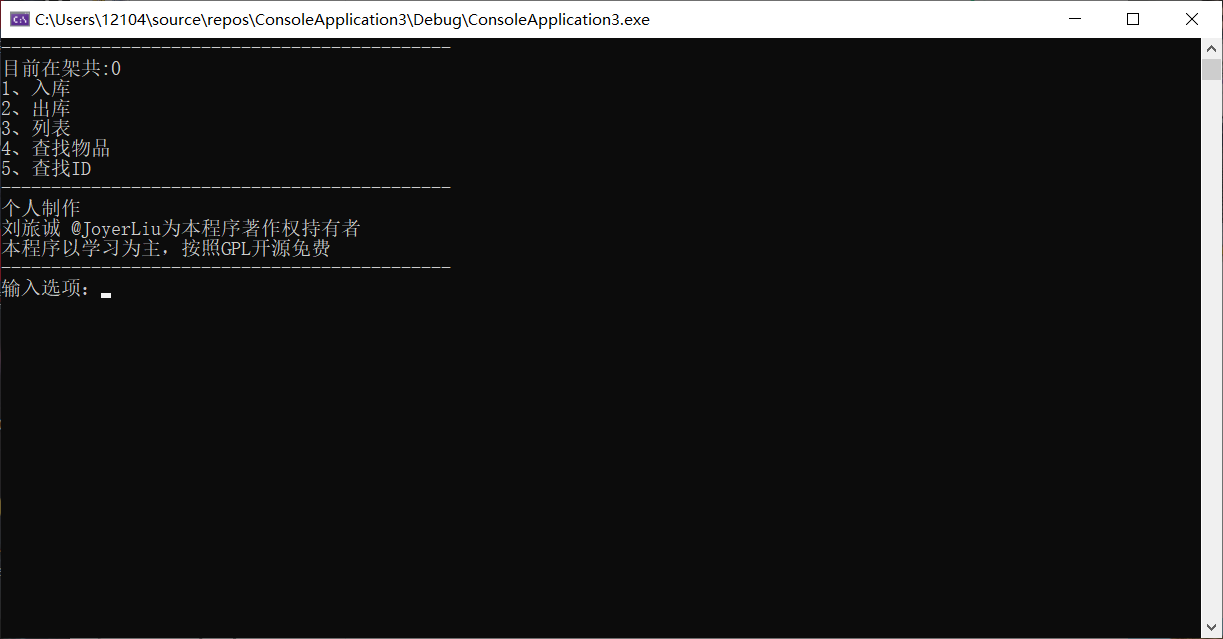


## API(MIB.h)

# 3 系统实现

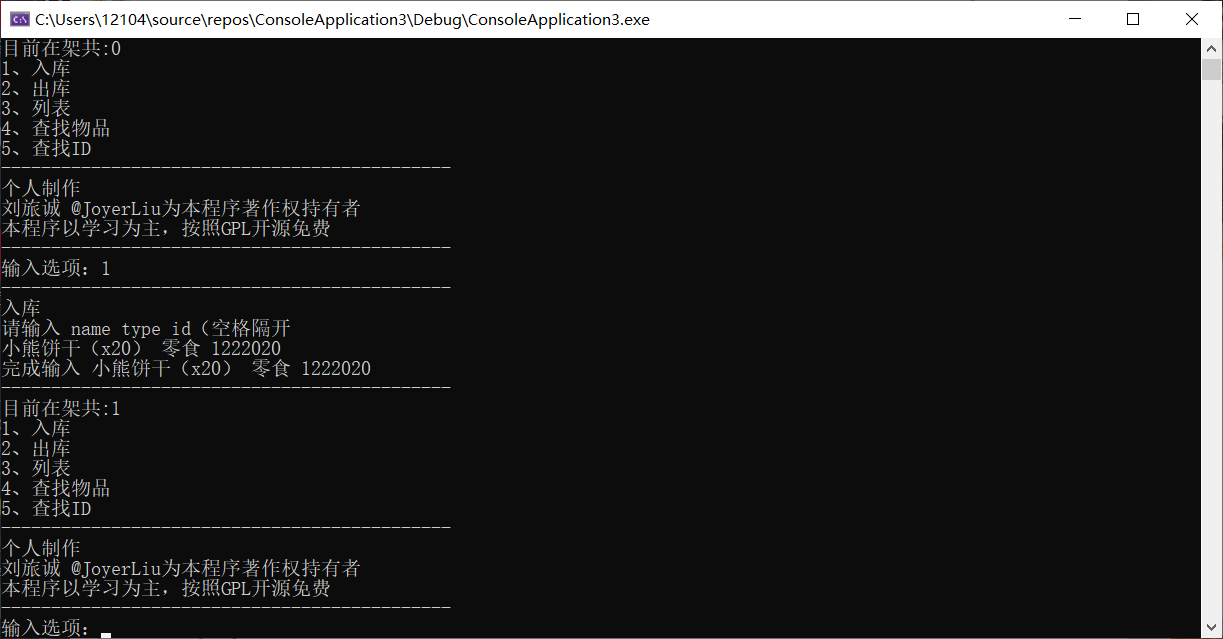
MIB.h Damo

1. 主界面

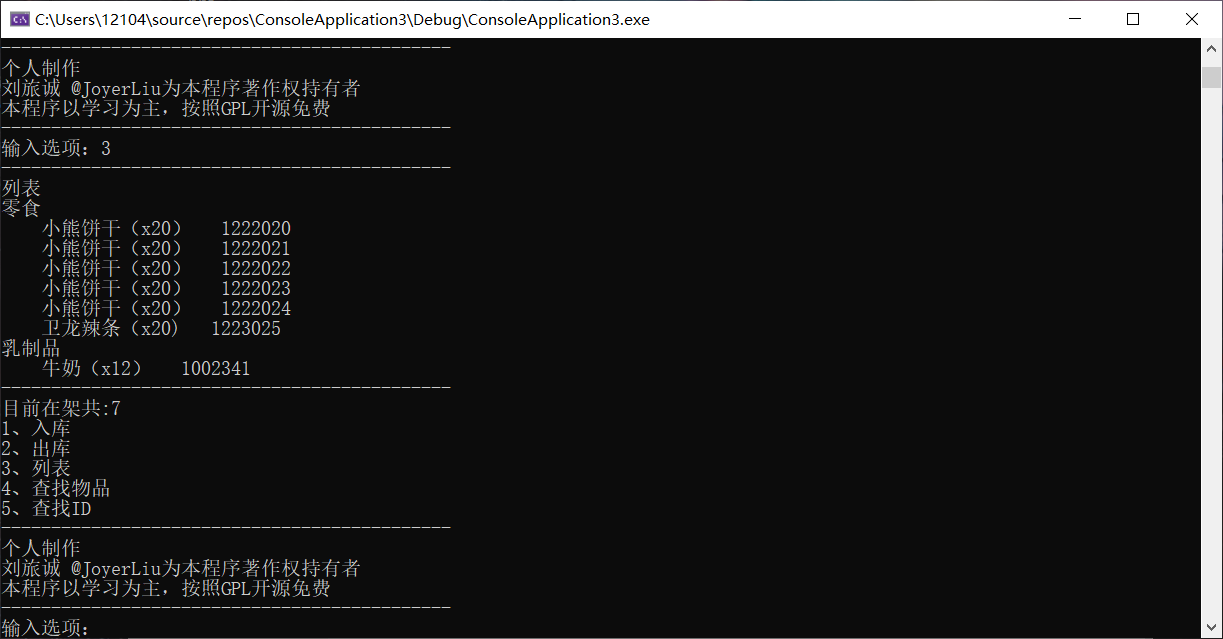


1. 入库界面，输入货品名字与货品类型与货品ID，API可以按需接入外部输入

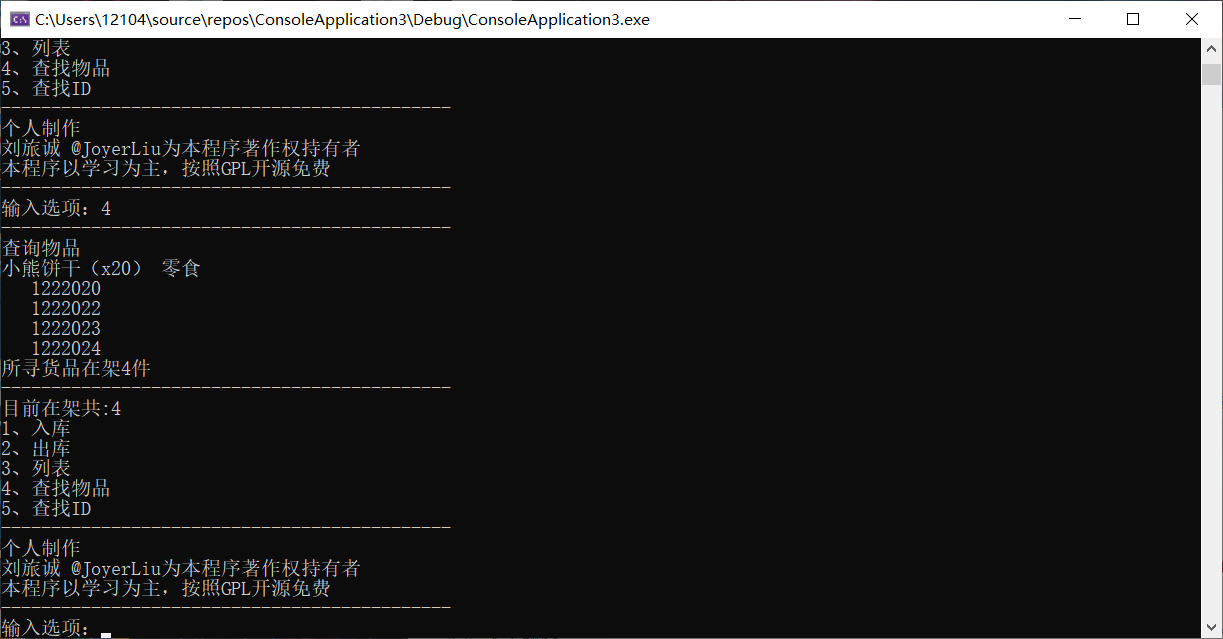




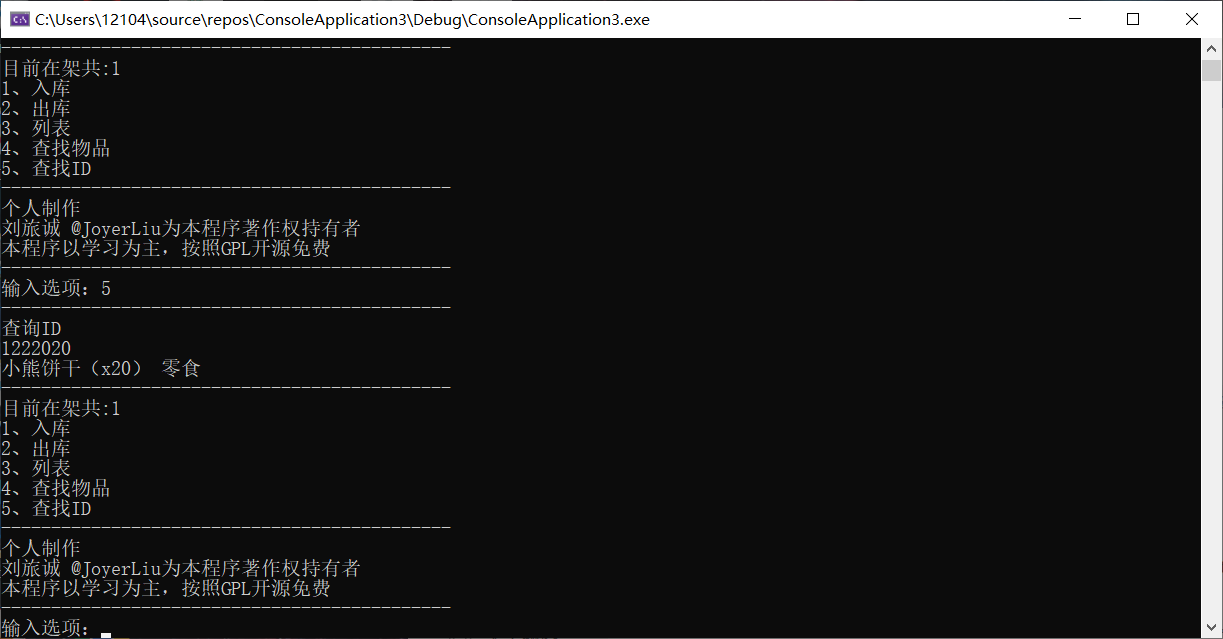
1. 列表，自动列表



1. 查找物品，以货品名与货品类型查找相对ID



1. 查找ID，以货品ID查找相对货品名与货品类型



1. 出库



## 源码

/\*本源码按照GPL开源协议开源，最终著作权属于本人\*/

### MIB.h

#pragma once

#include <iostream>

#include <vector>

#include <map>

using namespace std;

class goods

{

public:

long G\_id;

string G\_name;

string G\_type;

goods(string name, string type, long id)

{

G\_id = id;

G\_type = type;

G\_name = name;

}

};

class wareHouseMIB

{

private:

map<string, string> Findgoods\_type()//搜索货品类型

{

map<string, string> typ;

int a = 0;

for (vector<goods>::iterator i = goods\_.begin(); i != goods\_.end(); i++)

{

typ[i->G\_type] = i->G\_type;

a++;

}

return typ;

}

public:

vector<goods> goods\_;

int Input(string name, string type, long id)

{

if (Findgoods(id) == -1)

goods\_.push\_back(goods(name, type, id));

else

cout << "输入失败" << endl << "ID所指向的货物已在架，请检查ID输入" << endl;

return 0;

}

int Findgoods(string name, string type)

{

int count\_ = 0;

for (int i = 0; i < goods\_.size(); i++)

{

if (goods\_[i].G\_name == name && goods\_[i].G\_type == type)

{

cout <<" "<< goods\_[i].G\_id<<endl;

count\_++;

}

}

cout << "所寻货品在架" << count\_ << "件" << endl;

return count\_;

}

int Findgoods()

{

return goods\_.size();

}

int Findgoods(long id)

{

for (int i = 0; i < goods\_.size(); i++)

{

if (goods\_[i].G\_id == id)

return i;

}

return -1;

}

int outWare(long id)

{

int i = Findgoods(id);

if (i >= 0)

{

goods\_.erase(goods\_.begin() + i);

return 1;

}

return 0;

}

void OutgoodsList()

{

map<string, string> typ = Findgoods\_type();

for (map<string, string>::iterator i = typ.begin()++; i != typ.end(); i++)

{

cout << i->first << endl;

for (vector<goods>::iterator j = goods\_.begin(); j != goods\_.end(); j++)

{

if (i->first == j->G\_type)

{

cout << " " << j->G\_name << " " << j->G\_id << endl;

}

}

}

}

};

### Main.cpp

#include <iostream>

#include <vector>

#include <map>

#include "MIB.h"

using namespace std;

int main()

{

wareHouseMIB MIB;

string name, type;

long id;

int a = 0;

while (true)

{

cout << "---------------------------------------------" << endl;

cout << "目前在架共:"<<MIB.Findgoods() << endl;

cout << "1、入库" << endl << "2、出库" << endl << "3、列表" <<endl<<"4、查找物品"<<endl<<"5、查找ID"<< endl;

cout << "---------------------------------------------" << endl;

cout << "个人制作\n" << "刘旅诚 @JoyerLiu为本程序著作权持有者\n" << "本程序以学习为主，按照GPL开源免费\n";

cout << "---------------------------------------------" << endl;

cout << "输入选项：";

cin >> a;

cout << "---------------------------------------------" << endl;

switch (a)

{

case 1:

cout << "入库" << endl;

cout << "请输入 name type id（空格隔开" << endl;

cin >> name >> type >> id;

MIB.Input(name, type, id);

cout << "完成输入 " << name << " " << type << " " << id << endl;

break;

case 2:

cout << "出库/删除" << endl;

cout << "请输入 id:" ;

cin >> id;

MIB.outWare(id);

break;

case 3:

cout << "列表" << endl;

MIB.OutgoodsList();

break;

case 4:

cout << "查询物品" << endl;

cin >> type >> name;

MIB.Findgoods(type,name);

break;

case 5:

cout << "查询ID" << endl;

cin >> id;

if (MIB.Findgoods(id) != -1)

cout << MIB.goods\_[MIB.Findgoods(id)].G\_name<<" "<< MIB.goods\_[MIB.Findgoods(id)].G\_type << endl;

else

cout << "未找到该物品或该物品未上架" << endl;

break;

}

}

}

# 4 总结

使用C/C++完成了基本仓库管理系统的编写，调试过程中未出现死循环或内存泄漏的Bug。本次的设计只能算是完成了基础功能，包括出库入库与数据查询功能，但并未有附加功能以及数据库接口。

通过这次课题使我巩固了我的编写程序的能力，使我对数据结构的构建有了更深的认知。