

# C++大作业

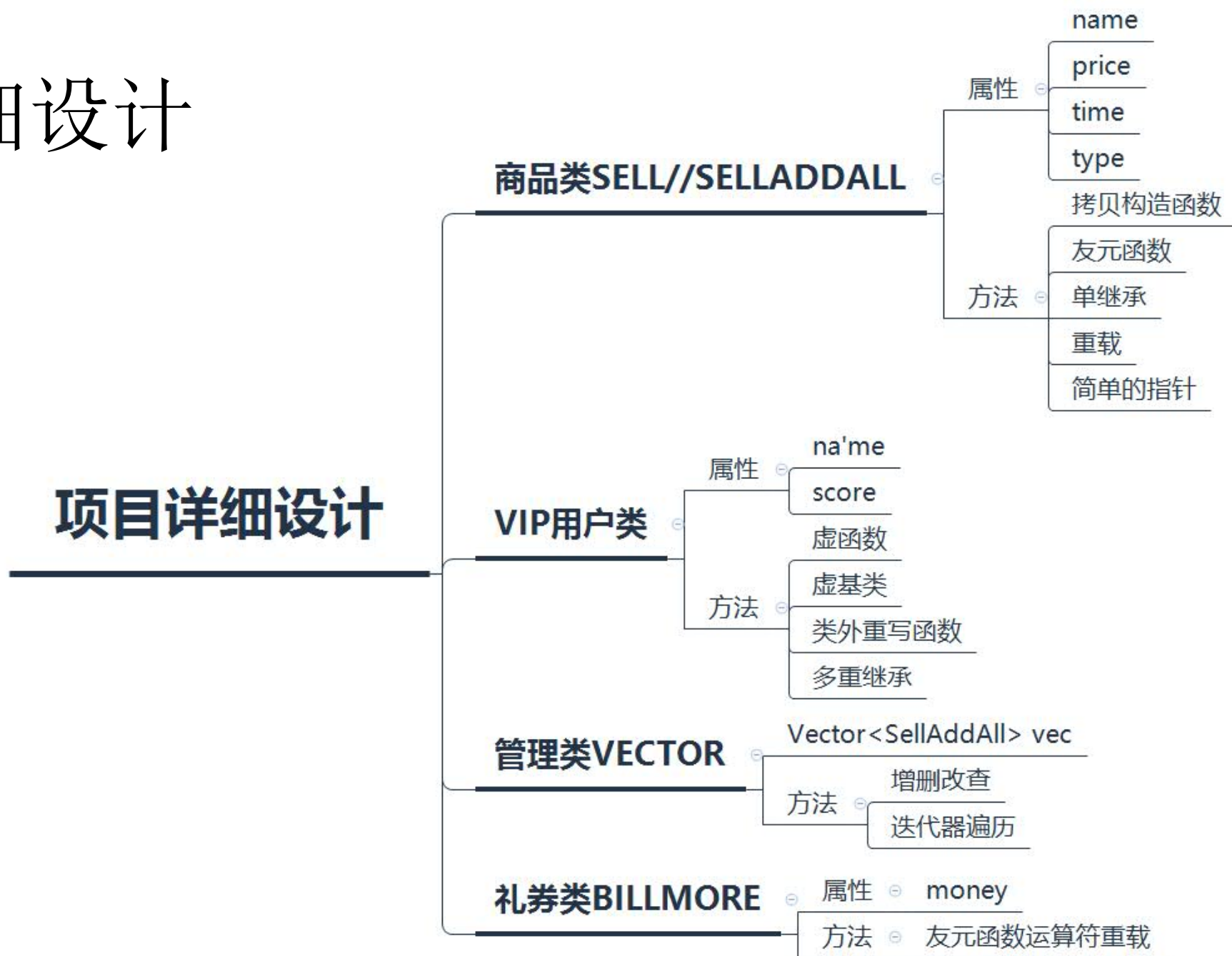
MacroF

xlccc

# 项目流程图



# 项目详细设计



# 构造函数与拷贝构造函数

```
class Sell{
public:
    char *name;
    int price;

    Sell();
    Sell(char*, int);
    Sell(const Sell &p);
    friend void printSell( Sell sell );//友元函数

    ~Sell()//析构函数 释放函数变量
    {
        cout<<"正在初始化"<<endl;
    }
    virtual void show();//虚函数
};
void Sell::show()
{
    cout<<name<<price<<endl;
};
Sell::Sell() {
    this->name = "xxx";
    this->price = 0;
};
Sell::Sell(char *name, int price): name(name), price(price) {};
Sell::Sell(const Sell &p)//拷贝构造函数
{
    this->name = p.name;
    this->price = p.price;
    cout<<"测试拷贝构造函数"<<endl;
};

Sell As ("k",1);//测试拷贝构造函数
A.show();
Sell Bs = As;
B.show();
```

```
"C:\C\1-2\Debug\kkk.exe"
欢迎来到自动贩卖机
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
1
请选择:
1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
1
测试拷贝构造函数
测试拷贝构造函数
测试拷贝构造函数
正在初始化
测试拷贝构造函数
测试拷贝构造函数
测试拷贝构造函数
正在初始化
正在初始化
测试拷贝构造函数
增加成功
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
```

```
"C:\C\1-2\Debug\kkk.exe"
欢迎来到自动贩卖机
可乐3
测试拷贝构造函数
炒饭5
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
1
```

# 友元函数

```
Sell();
Sell(char*, int);
Sell(const Sell &p);
friend void printSell( Sell sell );//友元函数

~Sell()//析构函数 释放函数变量
{
    cout<<"正在初始化"<<endl;
}
virtual void show();//虚函数
};
void Sell::show()
{
    cout<<name<<price<<endl;
};
Sell::Sell() {
    this->name = "xxx";
    this->price = 0;
};
Sell::Sell(char *name, int price): name(name), price(price) {};
Sell::Sell(const Sell &p)//拷贝构造函数
{
    this->name = p.name;
    this->price = p.price;
    cout<<"测试拷贝构造函数"<<endl;
};
void printSell( Sell sell )//友元函数
{
    cout<<"友元函数测试中: "<<endl;
    cout<<sell.name<<endl;
}

//派生类
```

"C:\C1-2\Debug\kkk.exe"

欢迎来到自动贩卖机

测试拷贝构造函数

友元函数测试中:

xxx

正在初始化

请选择:

1. 管理员列表
2. 客户购买列表
3. 程序结束

1

请选择:

1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单

# 多重继承与虚基类

```
class Vip1
{
protected:
    char Name [10];
public:
    Vip1(char *n)
    {
        strcpy(Name,n);
    }
    virtual void show();
};
class Vip2:virtual public Vip1
{
private:
    char university_name[20];
public:
    Vip2(char *n,char *un_n):Vip1(n)
    {
        strcpy(university_name,un_n);
    }
    void show();
};
class Vip3:virtual public Vip1 //虚基类
{
private:
    int course1;
public:
    Vip3(char *n,char *a,int c1):Vip1(n)
    {
        course1=c1;
    }
    void show();
};
class All : public Vip2 , public Vip3
```

```
};
class All : public Vip2 , public Vip3
{
private:
    int u;
public:
    All (char *n,char *a,char *un_n,int c1,int uu):Vip1(n),Vip2(n,un_n),Vip3(n,a,c1)
    {
        u=uu;
    };
    void show();
};
void All::show()
{
    // Vip2::show();
    Vip3::show();
}

void Vip1::show()
{
    cout<<endl;
    cout<<"姓名: "<<Name<<endl;
}
void Vip2::show()
{
    Vip1::show();
    cout<<"大学名称: "<<university_name<<endl;
}
void Vip3::show()
{
    Vip1::show();
    cout<<"会员积分: "<<course1<<endl;
}
//////////////////////////////////主菜单
```

# 虚函数

```
class Sell{
public:
    char *name;
    int price;

    Sell();
    Sell(char*, int);
    Sell(const Sell &p);
    friend void printSell( Sell sell );//友元函数

    ~Sell()//析构函数 释放函数变量
    {
        cout<<"正在初始化"<<endl;
    }
    virtual void show();//虚函数
};

void Sell::show()
{
    cout<<name<<price<<endl;
};

Sell::Sell() {
    this->name = "xxx";
    this->price = 0;
};

Sell::Sell(char *name, int price): name(name), price(price) {};

Sell::Sell(const Sell &p)//拷贝构造函数
```

```
////////////////////////////////多继承
class Vip1
{
protected:
    char Name [10];
public:
    Vip1(char *n)
    {
        strcpy(Name,n);
    }
    virtual void show();//虚函数
};

class Vip2:virtual public Vip1
{
private:
    char university_name[20];
public:
    Vip2(char *n,char *un_n):Vip1(n)
    {
        strcpy(university_name,un_n);
    }
    void show();
};

class Vip3:virtual public Vip1 //虚基类
{
private:
    int course1;
public:
    Vip3(char *n,char *a,int c1):Vip1(n)
    {
        course1=c1;
    }
    void show();
};
```

# 运算符重载

```
};  
class BillMore//随机促销礼券  
{  
public:  
    int money1 ;  
    int money2 ;  
    // money1 = money2 =0;  
    BillMore(void)  
    { money1 = money2 =0;  
    cout<<"总金额为:"<<money1+money2<<endl;  
    };  
    BillMore(int a)  
    {  
        money1 = vec[rand()%a].price;  
        money2 = 0;  
        cout<<"总金额为:"<<money1+money2<<endl;  
    }  
    BillMore(int a,int b )  
    {  
        money1 = vec[rand()%a].price;  
        money2 = vec[rand()%b].price;  
        cout<<"总金额为:"<<money1+money2<<endl;  
    };  
    ~BillMore() {};  
    friend BillMore operator+(const BillMore &px,const BillMore &py);//友元函数重载运算符  
    void print(void);  
    //cout<<"总金额为:"<<money1+money2<<endl;  
};  
void BillMore::print()  
{  
    cout<<"恭喜您获得总金额为:"<<money1+money2<<"的抵扣券"<<endl;  
};  
  
BillMore operator+(const BillMore &px,const BillMore &py)//友元函数重载运算符  
{  
    return BillMore(px.money1+py.money1,px.money2+py.money2);  
};
```



# Vector存储class类（类中包含多个变量）

```
static vector<SellAddAll> vec;
```

```
class QuerySell
{
public:
    void querysell()
    {
        cout << "当前商品信息为: " << endl;
        /* for (auto it = vec.begin(); it != vec.end(); ++it)
        {
            cout << (*it).name << (*it).price << " ";
        }*/
        for(int i = 0 ;i <vec.size() ;i++)
        {
            cout<<"商品名称: "<<vec[i].name<<"商品价格: "<<vec[i].price<<
            "供应时间: "<<vec[i].time<<"商品种类: "<<vec[i].type<<endl;
        }
        cout << endl;
    }
};
```

# 重载

```
class SellAddAll: public Sell{
public:
    char *type;
    char *time;

    SellAddAll();
    SellAddAll(char*, int, char*,char*); //重载
    void display();
};

SellAddAll::SellAddAll() {
    this->time = "全天供应";
    this->type = "yyy";
}

SellAddAll::SellAddAll(char *name, int age, char *time,char *type): Sell(name, age){
    this->time = time;
    this->type = type;
}

void SellAddAll::display() {
    cout<<"商品名称是:"<<name<<",价格是:"<<price<<", 供应时间是"<<time<<", 商品类型是:"<<type<<endl;
}
```

# 增（管理员）

```
"C:\C\1-2\Debug\kkk.exe"
欢迎来到自动贩卖机
请选择：
1. 管理员列表
2. 客户购买列表
3. 程序结束
1
请选择：
1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
1
正在初始化
正在初始化
正在初始化
增加成功
请选择：
1. 管理员列表
2. 客户购买列表
3. 程序结束
```

# 删/查

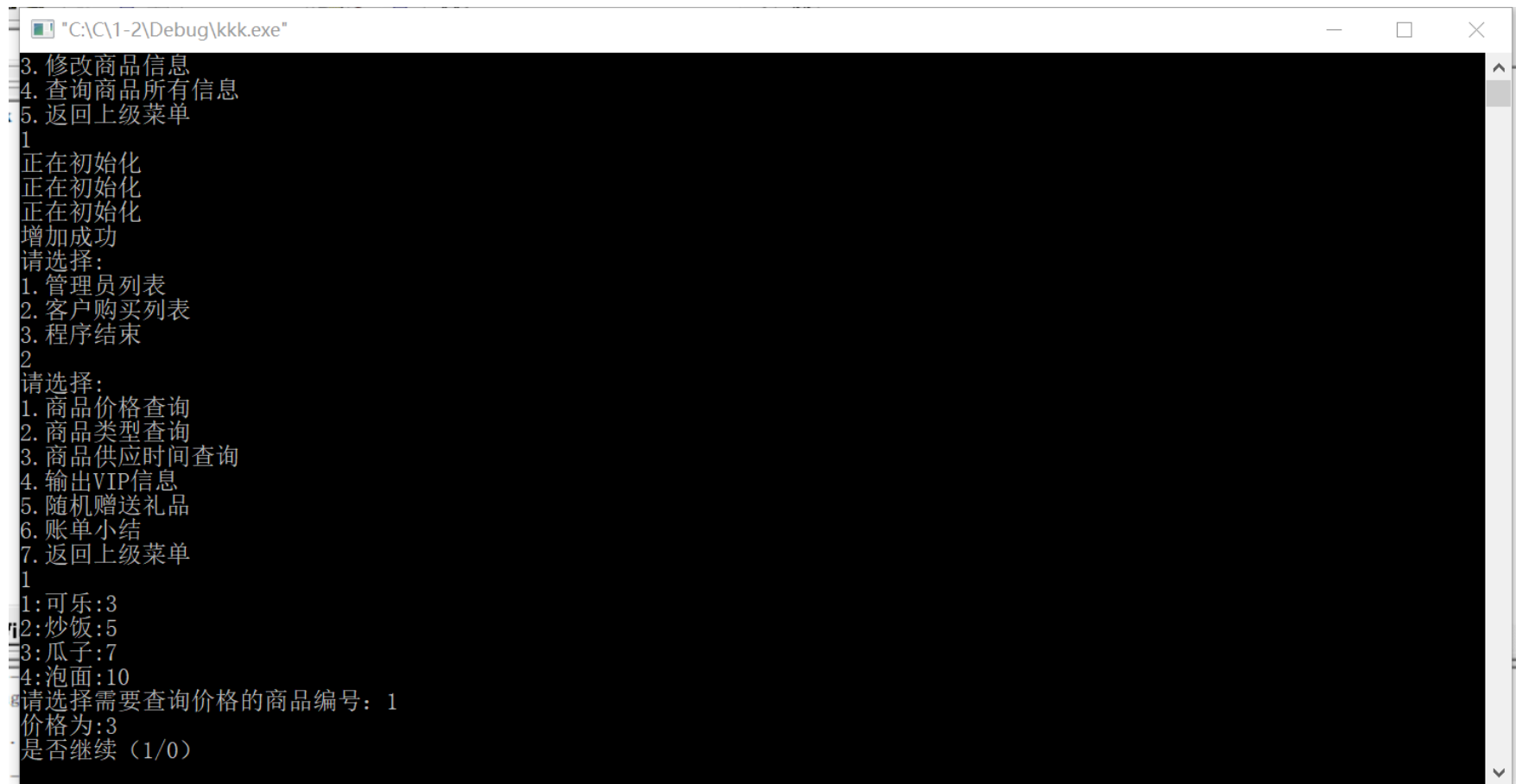
```
"C:\C\1-2\Debug\kkk.exe"
请选择:
1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
2
1: 可乐3
2: 炒饭5
3: 瓜子7
4: 泡面10
请选择需要删除的商品编号:
1
正在初始化
删除成功:
是否继续 (1/0)
0
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
1
请选择:
1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
4
当前商品信息为:
商品名称: 炒饭 商品价格: 5 供应时间: 早上八点 商品种类: 主食
商品名称: 瓜子 商品价格: 7 供应时间: 全天 供应商品种类: 零食
商品名称: 泡面 商品价格: 10 供应时间: 晚上八点 商品种类: 宵夜

请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
```

# 改

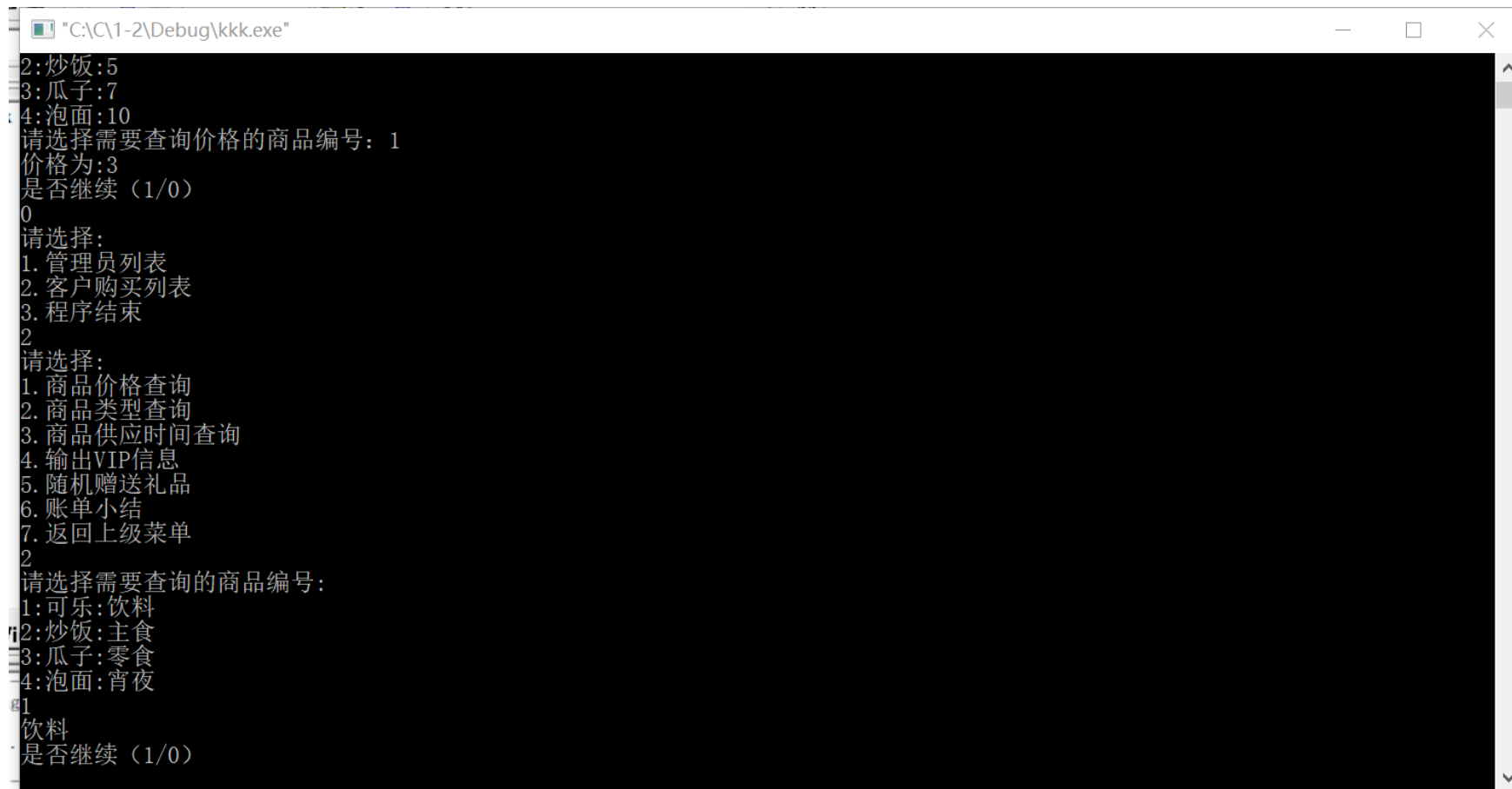
```
请选择:
1. 初始化商品信息
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
3
1: 可乐3
2: 炒饭5
3: 瓜子7
4: 泡面10
请选择需要修改的商品编号:
2
修改成功
1: 可乐3
2: 自热小火锅30
3: 瓜子7
4: 泡面10
```

# 查询价格//类型//供应时间



```
"C:\C\1-2\Debug\kkk.exe"
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
1
正在初始化
正在初始化
正在初始化
增加成功
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
2
请选择:
1. 商品价格查询
2. 商品类型查询
3. 商品供应时间查询
4. 输出VIP信息
5. 随机赠送礼品
6. 账单小结
7. 返回上级菜单
1
1: 可乐:3
2: 炒饭:5
3: 瓜子:7
4: 泡面:10
请选择需要查询价格的商品编号: 1
价格为:3
是否继续 (1/0)
```

# 查询商品类型



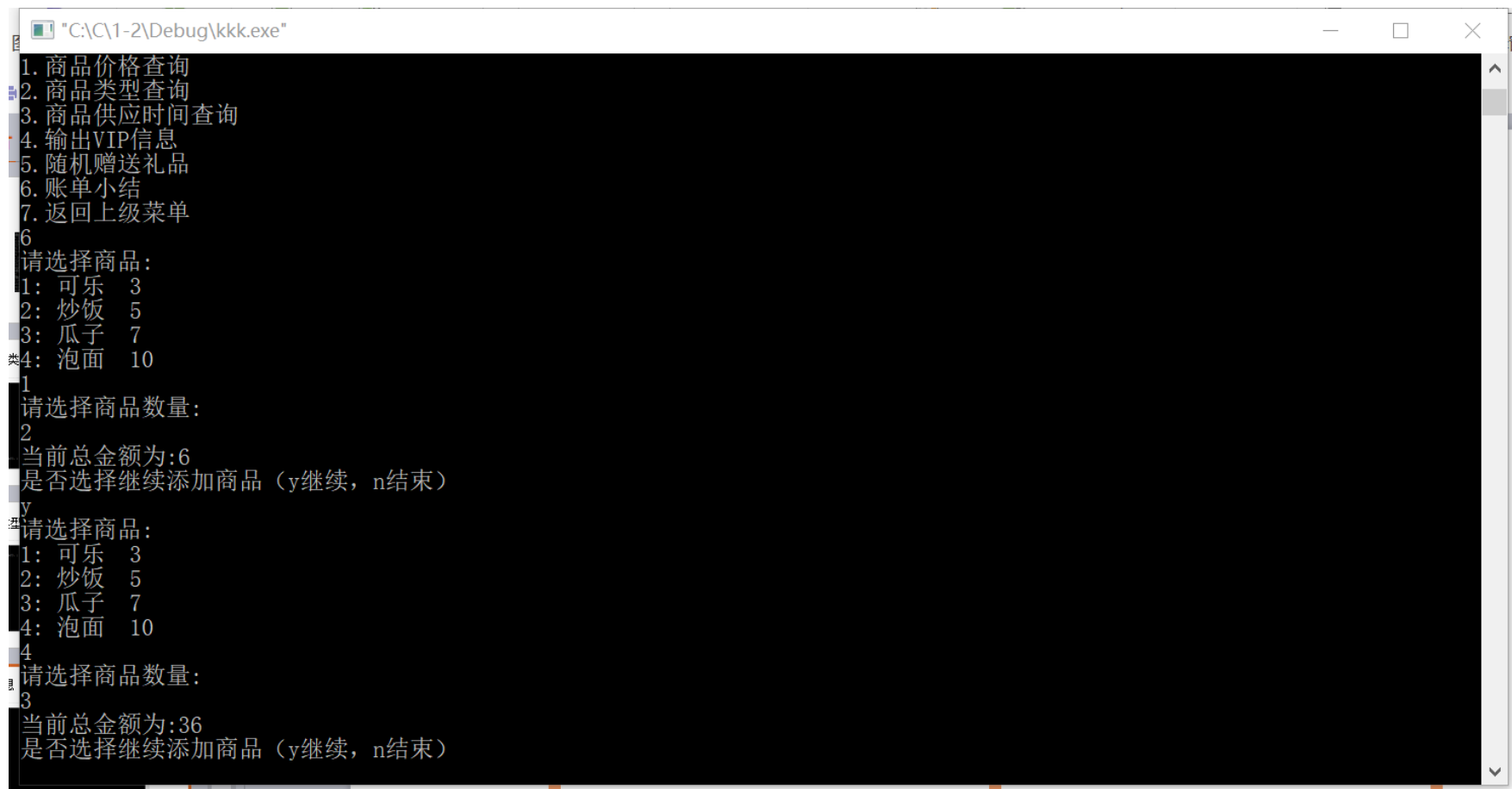
```
"C:\C\1-2\Debug\kkk.exe"
2:炒饭:5
3:瓜子:7
4:泡面:10
请选择需要查询价格的商品编号: 1
价格为:3
是否继续 (1/0)
0
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
2
请选择:
1. 商品价格查询
2. 商品类型查询
3. 商品供应时间查询
4. 输出VIP信息
5. 随机赠送礼品
6. 账单小结
7. 返回上级菜单
2
请选择需要查询的商品编号:
1:可乐:饮料
2:炒饭:主食
3:瓜子:零食
4:泡面:宵夜
1
饮料
是否继续 (1/0)
```

# 输出VIP信息

```
"C:\C\1-2\Debug\kkk.exe"
3:瓜子:全天供应
4:泡面:晚上八点
1
全天供应
输出成功
是否继续 (1/0)
0
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
2
请选择:
1. 商品价格查询
2. 商品类型查询
3. 商品供应时间查询
4. 输出VIP信息
5. 随机赠送礼品
6. 账单小结
7. 返回上级菜单
4
输出唯一VIP会员信息:
姓名: 林轩
会员积分: 95
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
```

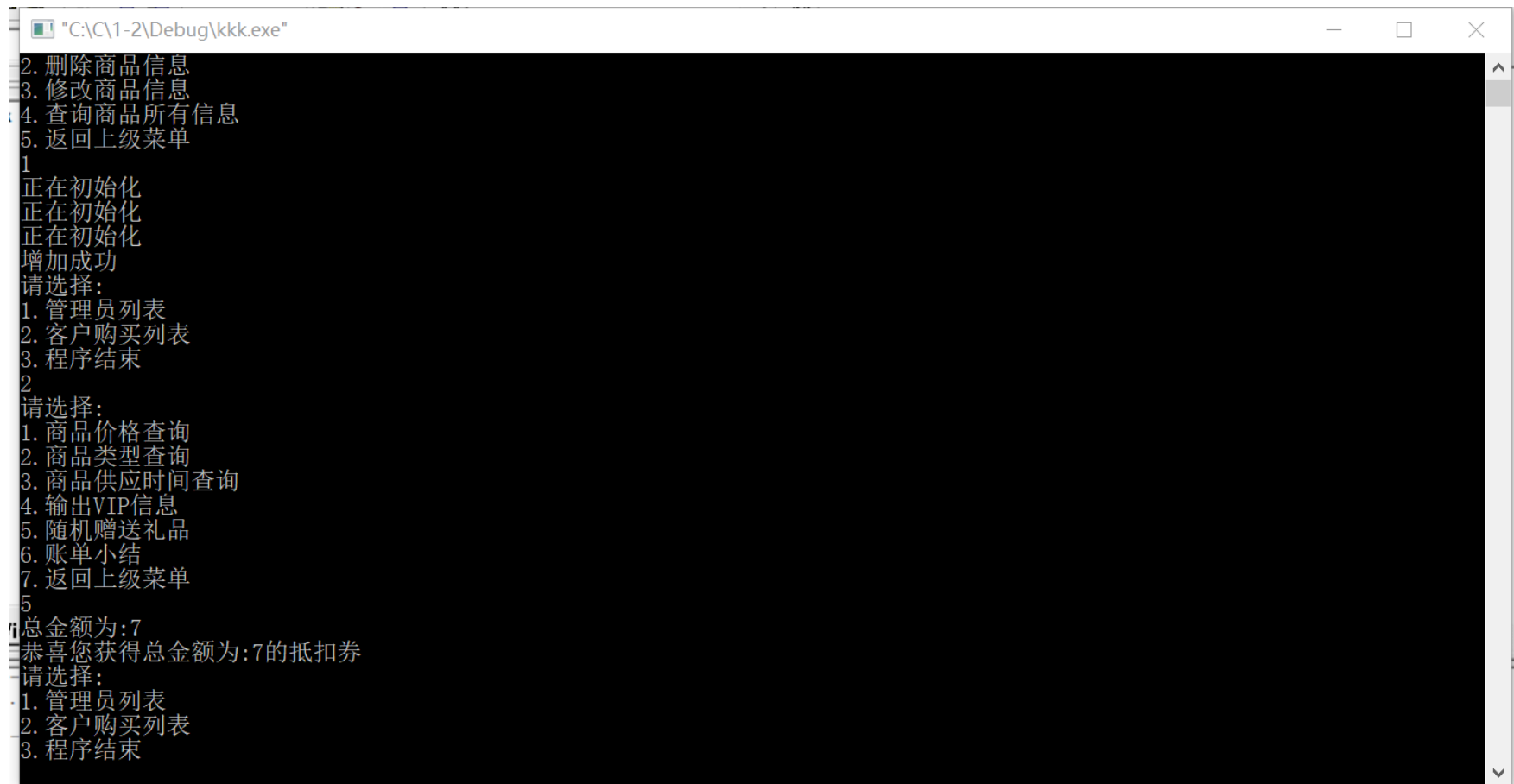


# 账单小结



```
"C:\C\1-2\Debug\kkk.exe"
1. 商品价格查询
2. 商品类型查询
3. 商品供应时间查询
4. 输出VIP信息
5. 随机赠送礼品
6. 账单小结
7. 返回上级菜单
6
请选择商品:
1: 可乐 3
2: 炒饭 5
3: 瓜子 7
4: 泡面 10
1
请选择商品数量:
2
当前总金额为:6
是否选择继续添加商品 (y继续, n结束)
y
请选择商品:
1: 可乐 3
2: 炒饭 5
3: 瓜子 7
4: 泡面 10
4
请选择商品数量:
3
当前总金额为:36
是否选择继续添加商品 (y继续, n结束)
```

# 随机小礼券



```
"C:\C\1-2\Debug\kkk.exe"
2. 删除商品信息
3. 修改商品信息
4. 查询商品所有信息
5. 返回上级菜单
1
正在初始化
正在初始化
正在初始化
增加成功
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
2
请选择:
1. 商品价格查询
2. 商品类型查询
3. 商品供应时间查询
4. 输出VIP信息
5. 随机赠送礼品
6. 账单小结
7. 返回上级菜单
5
总金额为:7
恭喜您获得总金额为:7的抵扣券
请选择:
1. 管理员列表
2. 客户购买列表
3. 程序结束
```