

《程序设计综合实验》

实验指导书

武汉理工大学

2024 年 2 月

1 实验项目简介

1.1 实验项目背景

计费软件，主要面向网吧、机房、电子阅览室等，由于其在计费管理上所体现的突出的优越性，可满足既想实现轻松管理，又想实现效益最大，效益可持续化的多重需要，成为了网吧、机房经营管理人员的理想选择，是打造品牌网吧和机房的前提。

计费软件定义计费策略，具有多种计费方式的计费平台，按时间计费，用卡进行消费；特殊计费，如按流量计费、包天或者月计费等。本系统可以根据计费策略，进行各种形式计费，并能灵活进行费用结算，以及统计日常报表。计费软件提供计费管理、费用管理、查询统计、系统管理功能。

1.2 实验项目需求与设计

(1) 功能结构

系统功能结构图参考如下：

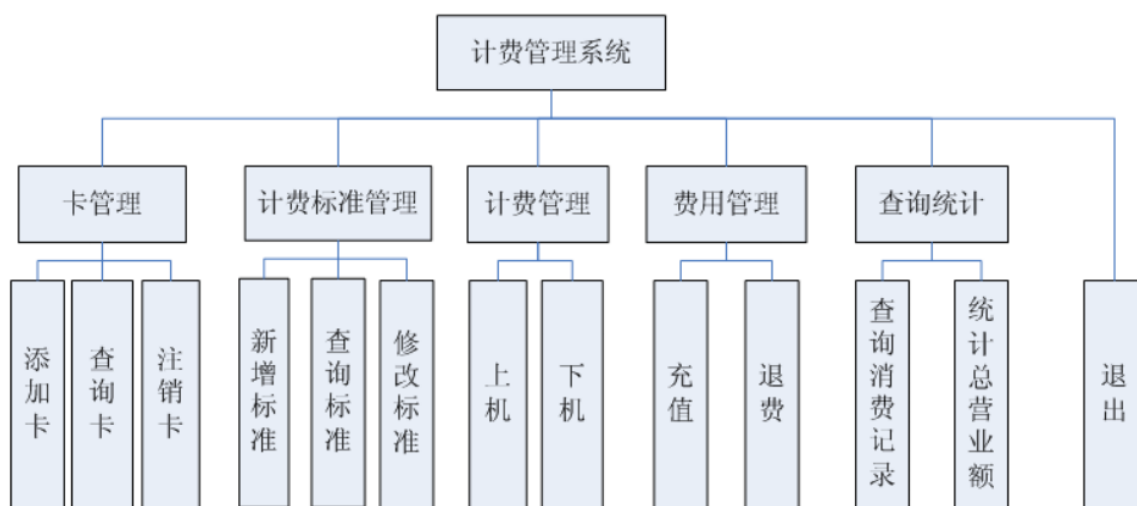


图 1 系统功能结构图

(2) 系统设计

系统层次设计可参考图 2。

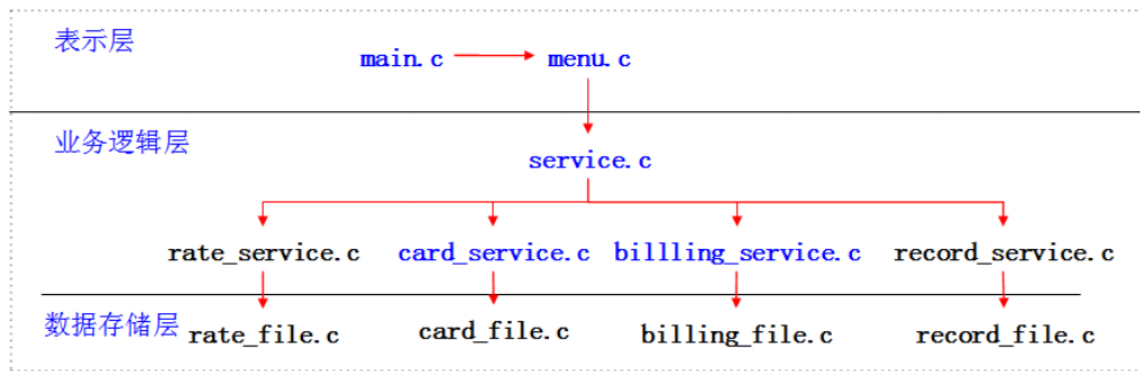


图2 系统层次设计

(3) 项目需求与设计总结

- ① 熟悉软件开发流程，了解各个阶段的产出物及内容形式。
- ② 对项目功能与实现进行了解，掌握开发方向。

2 实验内容

2.1 实验一 计费软件的需求分析与人机接口设计

2.1.1 实验目的

- (1) 熟练使用格式化输入输出函数。
- (2) 熟练使用数据类型。
- (3) 熟练使用并掌握分支语句。
- (4) 掌握如何逐行输入菜单。
- (5) 掌握如何根据菜单项编号，输入菜单。
- (6) 熟练使用循环输出语句。
- (7) 了解函数的定义、熟练调用函数，实现输出菜单功能。
- (8) 熟练搭建程序框架。
- (9) 综合应用以上知识，完成“计费软件的需求分析与人机接口设计”。

2.1.2 实验内容

- (1) 搭建开发环境。

开发工具：Microsoft Visual Studio

操作：下载 Microsoft Visual Studio 软件、安装、配置。

- (2) 创建项目，输出“计费软件”。

步骤一：创建解决方案

选择解决方案为“Blank Solution”，解决方案名为：AMS。

步骤二：创建项目

“计费软件”是一个控制台程序，选择项目类型为“Empty Project”。

项目名为 AccountManagement。

步骤三：添加 main.c 文件

在项目中添加一个“.cpp”文件，文件名为“main.c”，作为程序的入口文件。

步骤四：编辑程序

1) 程序入口：控制台程序的入口函数为 `main()` 函数，在 `main.c` 文件中添加 `main()` 函数，作为程序的入口函数。

2) 输出文本：在 `main()` 函数中，调用 `printf()` 输出文本。

步骤五：编译和连接程序

步骤六：运行程序

(3) 在控制台界面显示选择菜单，并输出用户选择的菜单名称。

在“创建项目”的基础上迭代开发，在 `main()` 函数中实现本次迭代的功能，实现步骤如下：

步骤一：导入解决方案

步骤二：输出系统名称和系统菜单

根据系统的功能，每个功能对应一个菜单项，每个菜单项都对应一个编号。系统的功能包括：添加卡、查询卡、上机、下机、充值、退费、查询统计、注销卡。

步骤三：提示用户选择菜单编号

使用 `switch~case` 语句，根据输入的菜单项编号，输出对应的菜单名。

步骤四：输出用户的选择

(4) 循环输出菜单。

在本次迭代中，`do...while` 循环的循环体语句就是上一个迭代功能，输出菜单，其中包括输出菜单、提示用户选择和输出用户的选择。

使用 `do...while` 循环，首先输出一次菜单，供用户选择。循环结束的条件为输入的菜单项编号为 0，则不再输出菜单，结束程序。

(5) 将输出菜单语句，封装成函数。

“主菜单函数”功能在“循环输出菜单”功能迭代上进行开发，步骤如下：

步骤一：定义 `outputMenu()` 函数。

步骤二：调用 `outputMenu()` 函数。

步骤三：声明 `outputMenu()` 函数。

(6) “搭建程序框架”在“主界面”功能的基础上进行迭代开发。

程序整体按三层结构进行搭建：表示层、业务逻辑层、数据存储层。

(7) 根据三层结构的思路，修改程序代码。

编写相应预处理指令：文件包含，条件编译，宏定义。

2.2 实验二 计费软件的数据结构设计

2.2.1 实验目的

- (1) 掌握数组的概念。
- (2) 熟练使用数组存储数据。
- (3) 掌握结构体的概念。
- (4) 熟练使用结构体存储数据。
- (5) 掌握 typedef 定义类型。
- (6) 掌握结构体变量的定义。
- (7) 熟练引用结构体成员。
- (8) 熟练使用 for 循环语句。
- (9) 熟练使用 while 循环语句。
- (10) 掌握指针的概念。
- (11) 熟练使用指针和数组名作为函数参数。
- (12) 综合应用以上知识，完成“计费软件的数据结构设计”。

2.2.2 实验内容：

- (1) 在“搭建程序框架”功能迭代上进行开发，数据设计。

根据“计费软件”需求，分析出系统数据信息包括：卡信息、计费信息、充值退费信息、计费标准信息、管理员信息。对以上各类信息做出相应的表设计。

- (2) 定义卡信息变量。

步骤一：直接定义卡信息变量。

根据数据设计内容定义卡信息变量。

步骤二：用结构体定义卡信息。

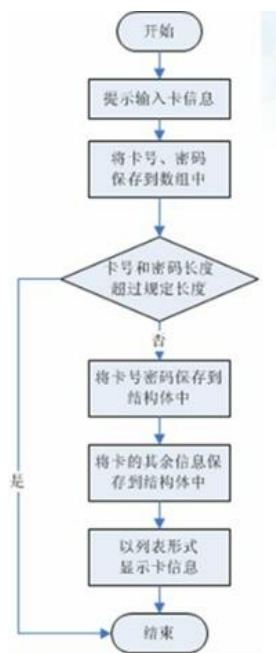
卡信息的结构体类型定义如下：

```
typedef struct Card
{
    char aName[18];    // 卡号
    char aPwd[8];      // 密码
    int nStatus;       // 卡状态(0- 未上机; 1-正在上机; 2-已注销; 3-失效)
    time_t tStart;     // 开卡时间
    time_t tEnd;       // 卡的截止时间
    float fTotalUse;    // 累计金额
    time_t tLastTime;  // 最后使用时间
    int nUseCount;     // 使用次数
    float fBalance;     // 余额
    int nDel;          // 删除标识(0-未删除,1-删除)
}Card;
```

(3) 添加卡信息。

在“数据结构”的基础上进行迭代开发，添加一张新的卡，将卡信息存到结构体变量中。

卡信息包括：卡号、密码、开卡金额、卡状态、开卡时间、截止时间、最后使用时间、使用次数、累积金额和删除标识。



(4) 显示卡信息。

从卡信息结构体中，获取卡号，密码，开卡金额，卡状态，并以列表方式显示。

使用 printf() 函数和制表符“t”，按照表格的形式显示卡号，密码，开卡金额，卡状态。

卡号	密码	状态	开卡金额
11	11	0	10.0

(5) 保存添加的卡信息。

将添加卡的信息保存到结构体数组中，上一次迭代，是将添加的卡信息保存在卡的结构体中，这样只能保存一条卡信息。



(6) 查询对应的卡信息。

根据输入的卡号，查询卡号相同的卡信息。

(7) 显示查询到的卡信息。

```

-----菜单-----
1. 添加卡
2. 查询卡
3. 上机
4. 下机
5. 充值
6. 退费
7. 查询统计
8. 注销卡
9. 退出
请选择菜单项编号 (0~8) : 2
-----查询卡-----
请输入查询的卡号<长度为1~18>: test
卡号 状态 余额 累计使用 使用次数 上次使用时间
test 0 50.0 50.0 0 2013-09-24 13:35
-----菜单-----
1. 添加卡
2. 查询卡
3. 上机
4. 下机
5. 充值
6. 退费
7. 查询统计
8. 注销卡
9. 退出
请选择菜单项编号 (0~8) : 2
-----查询卡-----
请输入查询的卡号<长度为1~18>: ttt
没有该卡的信息!
  
```

2.3 实验三 计费软件的文件存储管理

2.3.1 实验目的

- (1) 了解文件操作的过程
- (2) 学习如何将卡信息保存到文本文件
- (3) 学习对文件、文件名的读取
- (4) 了解文件类型指针
- (5) 了解 FILE 类型结构体
- (6) 了解如何打开文件、如何读取数据
- (7) 了解如何关闭文件
- (8) 学习如何操作字符串
- (9) 学习字符串分隔、类型转换

2.3.2 实验内容

(1) 搭建三层架构

- ① 在数据访问层实现添加卡，在 card_file.c 文件中，定义 saveCard () 函数
- ② 在业务逻辑层实现添加卡，修改 addCard () 函数，在此函数中调用数据访问层里的 saveCard 函数，保存卡信息。
- ③ 在表示层实现添加卡
- ④ 编译并运行程序

(2) 在数据访问层，实现用文件保存添加的卡

- ① 实现 saveCard () 函数，将卡信息保存到文件。
- ② 调用 saveCard 函数，将添加的卡信息保存到项目目录下的 card.txt 文件

(3) 读取文件中的卡信息

- ① 在 card_file.c 文件中，定义 readCard () 函数
- ② 编写 readCard () 函数，实现从文件中读取卡信息的功能
- ③ 编译并运行程序

(4) 解析读取的卡信息

- ① 在 card_file.c 文件中，定义 praseCard () 函数
- ② 编写 praseCard () 函数，实现解析卡信息字符串的功能
- ③ 在 tool.c 文件中，定义 stringToTime () 函数

④编写 stringToTime () 函数，将字符串转换为 time_t 类型的数据

⑤在 praseCard () 函数中，调用 stringToTime () 函数

⑥在 readCard () 函数中，调用 praseCard () 函数

⑦编译并运行程序

(5) 将解析后的卡信息保存到链表中

①在 card_service.c 文件中，定义 getCard () 函数，将读取出来的卡信息保存到链表

②编写 getCard () 函数，将卡信息保存到链表中。

③在 card_file.c 文件中，定义函数 getCardCount () 函数

④编写 getCardCount () 函数，获取卡信息文件中的卡信息数量

⑤在 getCard () 函数中，调用 getCardCount () 函数

⑥编译运行

(6) 根据条件查询并显示卡信息

①修改 queryCard () 函数，先调用 getCard () 函数，从文件中获取到卡信息保存到链表中，再根据卡号查询卡。

② 编译运行程序

2.4 实验四-五 计费软件的链表基本操作与数据动态存储管理

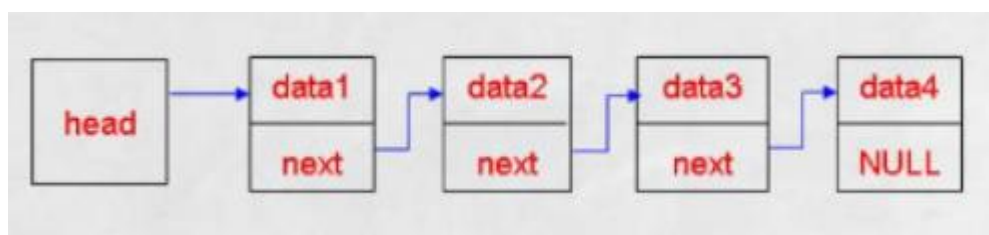
2.4.1 实验目的

- (1) 了解如何使用链表实现添加卡信息
- (2) 学习链表的组成
- (3) 理解头指针、节点、表尾的概念和使用方法
- (4) 学习链表节点的组成
- (5) 理解数据域、指针域的概念和使用方法
- (6) 学习如何添加节点
- (7) 理解如何为节点分配内存、如何将节点添加到链表
- (8) 学习内存的分配
- (9) 理解内存结构、动态分配内存、动态释放内存的概念和使用方法
- (10) 实现查询卡的功能
- (11) 将链表写入文件
- (12) 从文件中读入信息至链表

2.4.2 实验内容

(1) 定义卡信息链表

- ① 使用链表来保存添加的卡信息，首先要分析和设计链表中结点的结构
一个链表有下列部分组成：



- ② 声明卡信息链表结点类型

③ 优化卡信息链表的定义

④ 编译并运行

(2) 初始化卡信息链表

①在 card_service.c 文件中，定义函数 initCardList ()

②编写 ititCardList ()，实现初始化卡信息链表的功能

(3) 将卡信息添加到链表

①删除之前用结构体数组添加卡的代码。

②重新编写 addCard () 函数，将卡信息添加到链表中，实现思路如下：



③编译并运行。

(4) 释放卡信息链表

①了解计算机内存结构，内存开辟了是否需要释放

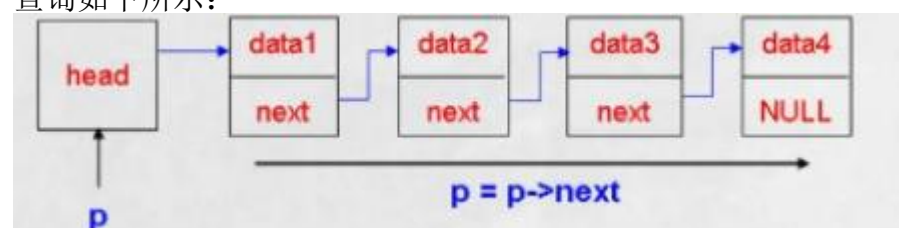
②释放卡信息链表的过程

③编译并运行

(5) 根据卡号，精确查询

①删除之前迭代中，用结构体数组实现 queryCard () 函数的代码

②重新编写 queryCard () 函数，在链表中查询卡信息。用 while 循环便利链表查询如下所示：



③编译并运行

(6) 根据关键字，模糊查询

①在 card_service.c 文件中，定义函数 queryCards ()

②编写函数 queryCards ()，实现模糊查询的功能，实现过程如下：

```

Card* queryCards(const char* pName, int* pIndex)
{
    lpCardNode node = NULL;
    Card* pCard = NULL; // 保存查询到的符合条件的卡信息
    ...

    // 遍历链表
    while(p != NULL)
    {
        // 判断当前结点中的卡号是否包含输入的关键字
        if(strstr(node->data.aName, pName) != NULL)
        {
            // 如果包含，则保存结点中的数据
            ...
            // 重新为指针分配内存
            ...
        }

        // 如果不相同，则将p指针指向下一个结点
    }
    return pCard;
}

```

③在 menu.c 文件中，修改 query（）函数

④编译并运行

（7）将链表写入文件

①将链表写入卡信息文件。

（8）从文件中读入信息至链表

①将卡信息文件中的卡信息读入到链表中。

2.5 实验六 计费软件的基本功能实现

2.5.1 实验目的

- (1) 学习文件操作相关函数
- (2) 学习文件的定位查找
- (3) 学习文件的修改
- (4) 学习的保存
- (5) 学习二进制文件的操作
- (6) 学习文件类型指针
- (7) 学习二进制文件的操作步骤

2.5.2 实验内容

(1) 更新上机卡

① 介绍

对已添加的卡，用户通过输入相匹配的卡号和密码，即可以开始上机，并在界面显示上机信息

② 输入

- a、输入菜单项编号 3，实现“上机功能”
- b、输入卡号和密码
- c、链表和文件中添加过的卡信息

③ 处理

- a、执行上机操作时，从卡信息文件中获取卡信息，添加到链表中。
- b、以卡号和密码为条件，遍历链表中的卡信息，找到与输入条件相符的卡信息。
- c、如果找到，以列表方式显示该卡的上机信息，并更新卡信息。如果未找到，就提示用户。

④ 输出

- a、如果找到需要上机的卡的信息，则以列表的方式显示。分为三列显示



卡信息，依次为：卡号，余额，上机时间。

b、如果未找到，则向界面输出“上机失败”，提示用户

(2) 添加消费记录

① 介绍

用户进行上机操作时，更新卡信息的同时，在 data 文件夹下面新增一个 billing.ams 文件来保存用户的消费信息。

消费信息包含：卡号，上机时间，下机时间，消费金额，消费状态，删除标识。

② 输入

a、输入菜单项编号“3”，实现“上机”功能

b、输入卡的消费信息

卡号：由数字和字母组成，最大长度是 18。

密码：由数字和字母组成，最大长度是 8。

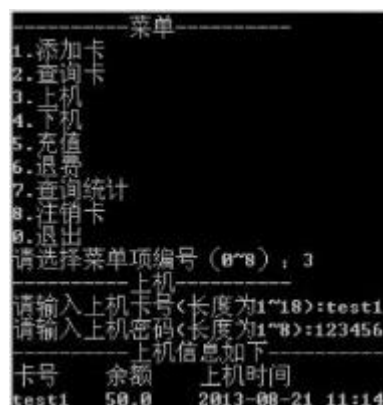
上机时间：用户上机时间，从系统获取当前时间。

下机时间：默认为 0。

消费金额：将用户每一次消费及消费的金额，进行累加。

消费状态：0 表示“未结算”，1 表示“已经结算”。

删除标识：0 表示“未删除”，1 表示“删除”。



③ 处理

a、获取项目目录下的 billing.ams 的文件路径

b、获取上机卡的消费信息

c、将上机卡的消费信息，写到项目目录下 billing.ams 二进制文件的末尾

④ 输出

a、上机失败，输出“上机失败！”提示用户。

b、上机成功，输出卡号，余额，上机时间。

c、卡状态不是未上机，提示“该卡正在使用或已注销！”。

d、卡余额小于 0，提示“余额不足！”

2.6 实验七 计费软件的扩展功能实现

2.6.1 实验目的

- (1) 练习文件操作相关函数
- (2) 练习文件的定位查找
- (3) 练习文件的修改
- (4) 练习文件的保存
- (5) 练习二进制文件的操作
- (6) 练习文件类型指针
- (7) 练习二进制文件的操作步骤
- (8) 分析项目需求，并完成相应的功能代码

2.6.2 实验内容

(1) 充值退费

① 介绍

当用户选择“5 充值”时，提示用户输入需要充值的卡的卡号、密码和充值金额。充值完成后，以列表的方式显示该卡的充值信息，包括卡号、充值金额和余额；如果失败，则提示用户。

② 输入

- a、输入菜单项编号“5”或者“6”
- b、输入需要充值或退费的卡的卡号和密码
- c、保存卡信息的文件路径 data/card.ams，和保存在文件中的卡信息
- d、保存卡的充值和退费信息的文件路径 data\money.ams

③ 处理

- a、读取卡信息文件中的卡信息，保存到链表中。
- b、以卡号和密码为条件，遍历链表，查询与该卡卡号和密码相同的卡信息
- c、如果找到，判断该卡是符合充值或退费的条件。
- d、若充值或退费，以列表的方式显示该卡的充值或退费信息
- e、若充值或退费失败，则提示用户。

④ 输出

- a、若果充值或退费失败，则输出相应的提示信息

b、若果充值或退费成功，则以列表的方式显示该卡的充值或退费信息



(2) 注销卡

① 介绍

用户选择菜单项“8 注销卡”后，提示用户输入需要注销的卡的卡号和密码。以卡号为条件，遍历卡信息链表，查找符合条件的卡。如果找到，以列表的方式显示注销的信息；如果未找到，提示用户。

注销的信息包括：卡号，退款金额

② 输入

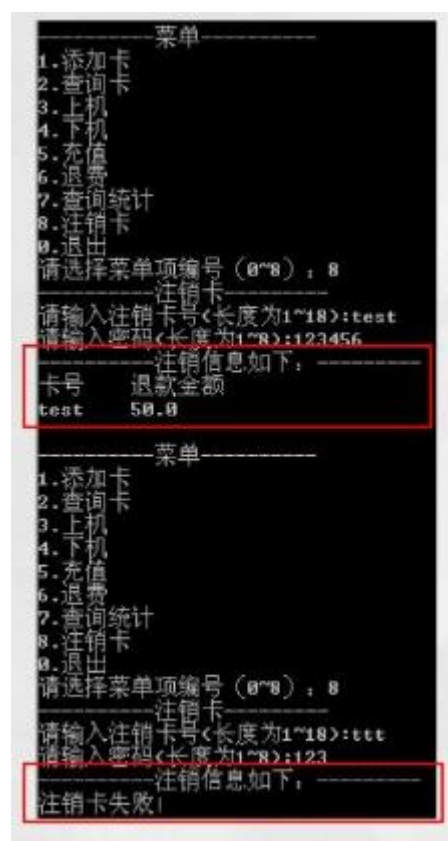
- 输入菜单项编号“8”
- 输入需要注销的卡的卡号和密码
- 保存卡信息的文件路径 data/card.txt
- 保存在文件中的卡信息

③ 处理

- 读取卡信息文件中的卡信息，保存到链表中。
- 以卡号为条件，遍历链表，找到第一个与该卡卡号和密码相同的卡信息
- 如果找到，判断该卡是否符合注销条件
- 注销失败，则提示用户
- 注销成功，以列表的方式显示该卡的注销信息

④ 输出

- 如果注销失败，则输出“注销卡失败！”提示用户
- 如果注销成功，则分两列来显示注销信息：卡号和退款金额



(3) 项目扩展（选做）

- ① 实现“查询统计”功能：查询消费记录、统计总营业额。
- ② 实现“费用管理”功能：充值、退费
- ③ 实现“计费标准管理”功能

2.7 实验八 系统验收

2.7.1 实验目的

- (1) 完成项目编码
- (2) 完成项目验收

2.7.2 实验内容

- (1) 项目过程：

项目按瀑布式模式里程碑阶段进行组织，经需求分析、系统设计、编码实现、系统测试、项目验收和关闭。编码采用迭代的方式进行增量开发。每个阶段的工作都被是为一个独立的项目，在完成编码和测试后，在进入下一个迭代。

- (2) 项目验收

项目完成后，对项目的功能及性能进行验收，并进行代码走查，对学生的技能水平进行相应的考察，若发现项目存在较大的问题，则进行重新开发，直至下次验收通过。