设计文档

规范

＊代码规范

1.花括号行尾式

2.局部变量和代码块之间用空行分开，方法与方法之间要有空行隔开

3.运算符前后有空格

4.要有必要注释，函数功能必须有

5.要有正确缩进

6.尽量少用全局变量

7.写完一个功能模块就要测试

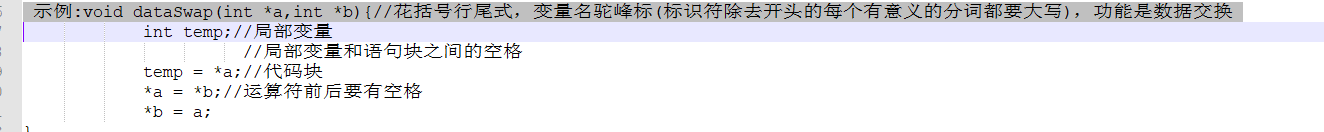
8.谨慎使用++,--运算符；

＊命名规范

1.函数名首字母小写，驼峰标识，如dataSwap；

2.函数名和变量名遵循见名知意的原则，如:dataSwap让人一看就知道数据交换函数

3.存储文件字母一律大写



文件约定：

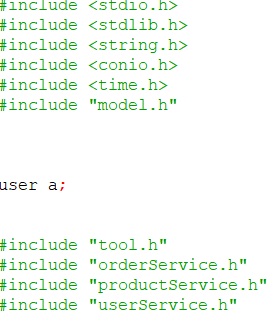
1.所有结构体放在一个头文件model.h，所有的头文件放在项目根目录；

2．与结构体事务管理相关功能模块写在结构体名+Service.h中:，如，userService.h;

3.工具类函数写在tool.h,比如密码加密解密等；

4.存储数据文件名为对应结构体名+.dat,比如user.dat,所有存储文件统一放在项目根目录；‘

4.主项目中个头文件调用顺序应改为：



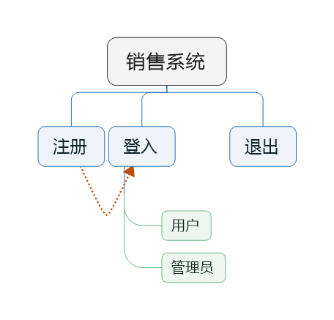
问题分析：

尝试模拟一个小型商城。。。。。。。。。。。

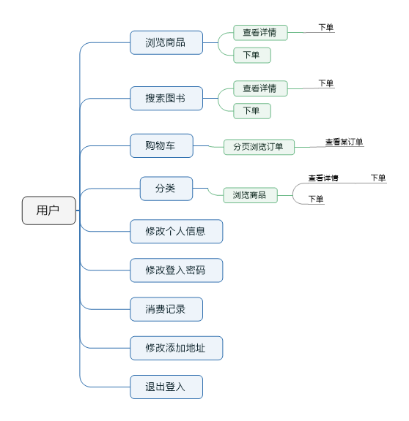
需求分析

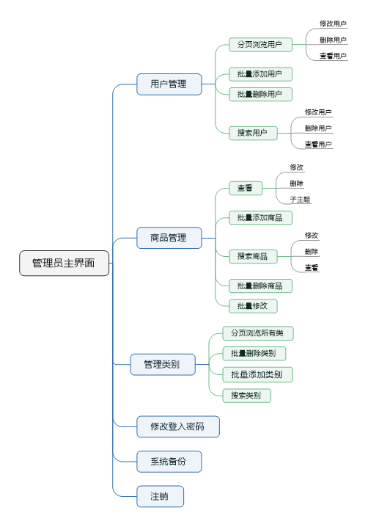
应用需要能够根据用户的最近浏览情况，以至于可以推荐他们需要的东西，并且东西种类齐全，以供选择。管理员能够看到客户的购买记录，浏览记录，修改客户信息，上台新的物品，管理物品。用户能够注册，登录自己帐号，修改密码等等，把自己想要的东西加购物车或者收藏，看到自己的购买以及浏览记录

总体设计



详细设计





事务结构体表

商品表（product）

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 允许为空 |
| 商品ID（id） | Int | 自动增长 | Not null |
| 商品名称（name） | 字符串 | 30 | Not null |
| 商品描述（descr） | 字符串 | 100 | Null |
| 商品原价（normalPrice） | double | 默认 | Not null |
| 打折后价（memberPrice） | double | 默认 | Not null |
| 是否上架（status） | Int(1或0) | 默认 | null |
| 库存（stock） | Int | 默认 | Not null |
| 销量（sales） | Int | 默认 | Not null |
| 所属类别（ofCategory） | Char[] | 30 | null |

类别表（category）

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 允许为空 |
| 类别ID（id） | long | 默认 | Not null |
| 类别名称（name） | 字符串 | 30 | Not null |
| 归属商品（goods） | Long 【】 | 默认 | Not null |

用户表（user）

|  |  |  |  |
| --- | --- | --- | --- |
| 用户帐号（userId） | 字符串 | 10 | Not null |
| 姓名（name） | 字符中 | 9 | Not null |
| 密码（passWord） | Int | 6 | Not null |
| Email(email) | 字符串 | 15 | Not null |
| 电话（phone） | 字符串 | 默认 | Null |
| 角色（role） | Int | 默认 | Not null |
| 常用地址（addr） | 字符串数组 | 80 | null |
| 备用地址1（addr1） | 字符串数组 | 80 | Null |
| 备用地址2(addr2) | 字符串数组 | 80 | Null |

订单表（order）

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 数据类型 | 数据长度 | 允许为空 |
| 序号（number） | Int | 500 | null |
| 订单号（int） | Char[] | 14 | Not null |
| 订单状况（status） | Int(1或0) | 1 | Not null |
| 订单内商品（items） | Item结构体数组 | 3 | Not null |
| 订单总价（totalPrice） | Double | 默认 | Not null |
| 订单数量(itemNum) | int | 最大为3 | Not null |
| 下单地址（addr） | 字符串 | 默认 | Notnull |

订单内商品项目（item）

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 数据类型 | 数据长度 | 允许为空 |
| 序号（number） | Int | 默认 | Not null |
| 商品id(iid) | int | 默认 | Not null |
| 数量（count） | Int | 默认 | Not null |
| 单价（price） | Double | 默认 | Not null |

工具类功能模块函数

1 /\*\*\*\*\*\*\*\*\*\*\*\*获取时间字符串\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void getTimeStr(char timeStr[])

2. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*密码加密\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void encryption(char password[])

3. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*密码解密\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void decryption(char password[])

4. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*kmp算next\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//模糊搜索

void makeNext(char p[],int next[])

5. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*kmp算法\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//模糊搜索

int kmpSearch(char s[],char p[],int next[])

6. //函数名称 mainMenu

//主菜单界面

//出口参数 int c

//函数功能： 选则登陆或退出

int mainMenu()

7. //显示时间与欢迎界面

void showTime(int k)

8. /\*\*\*\*\*\*\*\*\*\*\*主题栏\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void displayTopic(char \*c)

9. /\*\*\*\*\*\*\*\*\*\*\*\*\*产生验证码\*\*\*\*\*\*\*\*\*\*\*/

void getVerificationCode(char verificationCode[],int n)

10/\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件尾部写入用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeUserAfterFile(user s[],char \*f,int n)

11. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*从user.dat读取用户信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int readUserFromFile(user userArray[], char \*f)

12. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*重新写入用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeUserToFile(user userArray[],char \*f,int n)

13. /\*\*\*\*\*\*\*\*\*查询用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int userSearch(user userArray[],int userTotal,char id[])

14. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*从文件中删除用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int deleteUserFromFile(user userDeleteArray[],int n)

15. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*重新写入商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeProductToFile(product productArray[],char \*f,int n)

16. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件尾部写入商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeProductAfterFile(product productArray[],char \*f,int n)

17. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*从product.bat读取商品信息\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int readProductFromFile(product productArray[], char \*f)

18. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据id加载product\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

product loadProductById(int id)

19. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*将订单写入文件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeOrderToFile(order orderArray[],char \*f,int n)

20.\*\*\*\*\*\*\*\*\*\*\*\*\*将order从文件读取出来\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int readOrderFromFile(order orderArray[],char \*f)

21. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*文件尾部写入商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int writeOrderAfterFile(order orderArray[],char \*f,int n)

订单类功能模块函数

22. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*获取订单编号\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

char\*\* getOrderID(int itemNum,char \*\*oIdArray,char userId[])

23./\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 获取订单号中的用户账号\*\*\*\*\*\*\*\*\*/

Void getUserIDFromOrderID(char orderId[] ,char userId[])

24. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分页打印订单\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printOrder(order orderArray[],int orderTotal,char currentUserId[])

25. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*管理员分页打印订单\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printAllOrder(order orderArray[],int orderTotal,char currentUserId[])

用户注册类模块

26．/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*用户注册\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void registe()

27. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分页打印用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printUser(user userArray[],int userTotal)

28. /\*\*\*\*\*\*\*\*\*\*管理员添加用户时用户信息输入模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void inputUser()

29. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*根据id加载user\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

user loadUserById(char userId[])

30. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*删除用户\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int deleteUser()

31. /\*\*\*\*\*\*\*\*\*\*\*\*\*显示用户细节\*\*\*\*\*/

int userDetail(user searchedUser)

32. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*搜索用户\*\*\*\*\*\*\*\*\*\*\*\*\*/

void search\_user()

33. //\*\*\*\*\*\*\*\*\*\*\*\*\*显示普通用户角色功能菜单\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int menuUser()

34. //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示管理员角色功能菜单\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int menuAdmin()

35. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改登录密码\*\*\*\*\*\*\*\*\*\*/

void passwordChange(char currentUserId[])

36. //管理员模块

void adminSystem(char currentUserId[])

37. //用户模块

void userSystem(char currentUserId[])

38. //根据用户角色不同进入不同的管理子系统

void enterSystem(int c,char currentUserId[])

产品类模块

39. /\*\*\*\*\*\*\*\*\*查询product\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Int productSearch(product productArray[],int productTatal,int id)

40. /\*\*\*\*\*\*\*\*\*\*管理员添加商品时商品信息输入模块\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void inputProduct()

41. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*删除商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int deleteProductFromFile(product productDeleteArray[],int n

42. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*删除商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int deleteProduct()

43. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*修改商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int updateProduct(int id[],int n)

44. /\*\*\*\*\*\*\*\*\*立即购该商品\*\*\*\*\*\*\*\*\*\*\*/

void buyPoduct(product searchedProduct,user currentUser)

45. /\*\*\*\*\*\*\*\*\*\*\*\*\*显示商品细节\*\*\*\*\*/

void productDetail(product searchedProduct,char currentUserId[])

46. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*分页打印商品\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printProduct(product productArray[],int productTotal,char currentUserId[])

47. /\*\*\*\*\*\*\*\*\*\*\*产品搜索函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int productSearchKmp(product productSearchedArray[],char p[])

48. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*产品搜索模块\*\*\*\*\*\*\*\*\*\*\*\*\*/

void search\_product(char currentUserId[])