



C++ 面向对象程序设计

Assignment 4

Listing 1: login.h

```
1 #include <string>
2 #include <vector>
3 #include <algorithm>
4 #include <fstream>
5 #include <iostream>
6 #include <utility>
7
8 struct User {
9     User(std::string name, std::string pass, std::string mail);
10    ~User();
11
12    std::string username;
13    std::string email;
14    std::string password;
15 };
16
17 struct Login {
18     Login();
19     ~Login();
20
21     void readFile(std::string path);
22
23     bool checkUsername(std::string &new_username);
24
25     bool checkEmail(std::string &new_email);
26
27     void changeUsername(std::string username, std::string newUsername);
28
29     void changePassword(std::string username, std::string newPassword);
30
31     void changeMail(std::string username, std::string newMail);
32
33     void registerUser(std::string username, std::string password, std::string email); // function to
        create a user
34
35     void loginUser(std::string username, std::string password);
36
```

```

37 void removeUser(std::string username);
38
39 User *getUser(std::string username);
40
41 std::vector<User *> users;
42 std::vector<User *> LoggedUsers;
43 };

```

Listing 2: login.cpp

```

1  #include "../include/login.h"
2
3  User::User(std::string name, std::string pass, std::string mail) :
4  username(std::move(name)), password(std::move(pass)), email(std::move(mail)) {}
5
6  User::~User() {}
7
8  Login::Login() {}
9
10 Login::~Login() {
11     for (auto user : users) {
12         delete user;
13     }
14 }
15
16 void Login::readFile(std::string path) {
17     std::ifstream ifile(path);
18     std::string line;
19     std::getline(ifile, line); // skip the header line
20     std::string username, password, email, loggedIn;
21     while (std::getline(ifile, username, ',')) {
22         std::getline(ifile, password, ',');
23         std::getline(ifile, email, ',');
24         std::getline(ifile, loggedIn, ',');
25         registerUser(username, password, email);
26         if (loggedIn == "yes") {
27             loginUser(username, password);
28         }
29     }
30 }
31
32 bool Login::checkUsername(std::string &new_username) {
33     return std::any_of(users.begin(), users.end(), [&new_username](User *user) {
34         return user->username == new_username;
35     });
36 }
37

```

```

38 bool Login::checkEmail(std::string &new_email) {
39     return std::any_of(users.begin(), users.end(), [&new_email](User *user) {
40         return user->email == new_email;
41     });
42 }
43
44 void Login::changeUsername(std::string username, std::string newUsername) {
45     for (auto &user: users) {
46         if (user->username == username) {
47             std::swap(user->username, newUsername);
48             break;
49         }
50     }
51 }
52
53 void Login::changePassword(std::string username, std::string newPassword) {
54     for (auto &user: users) {
55         if (user->username == username) {
56             std::swap(user->password, newPassword);
57             break;
58         }
59     }
60 }
61
62 void Login::changeMail(std::string username, std::string newMail) {
63     for (auto &user: users) {
64         if (user->username == username) {
65             std::swap(user->email, newMail);
66             break;
67         }
68     }
69 }
70
71 void Login::registerUser(std::string username, std::string password, std::string email) {
72     if (checkUsername(username) || checkEmail(email)) {
73         std::cerr << "username already taken" << std::endl;
74         return;
75     }
76     users.emplace_back(new User(username, password, email));
77 }
78
79 void Login::loginUser(std::string username, std::string password) {
80     for (auto &user: users) {
81         if (user->username == username && user->password == password) {
82             if (std::find(LoginedUsers.begin(), LoginedUsers.end(), user) != LoginedUsers.end())
83                 return;

```

```

84     LoggedUsers.emplace_back(user);
85     break;
86 }
87 }
88 }
89
90 void Login::removeUser(std::string username) {
91     for (auto it = users.begin(); it != users.end(); ++it) {
92         if ((*it)->username == username) {
93             auto login = std::find(LoggedUsers.begin(), LoggedUsers.end(), *it);
94             if (login != LoggedUsers.end()) {
95                 LoggedUsers.erase(login);
96             }
97             users.erase(it);
98             break;
99         }
100     }
101 }
102
103 User *Login::getUser(std::string username) {
104     for (auto &user: users)
105         if (user->username == username)
106             return user;
107     return nullptr;
108 }

```

运行 所有 CTest x

✓ 测试 已通过: 5 共 5 个测试 - 130 毫秒

测试结果	时间
✓ LOGINTEST.checkRegistry	30 毫秒
✓ LOGINTEST.checkSameMail	20 毫秒
✓ LOGINTEST.checkremove	30 毫秒
✓ LOGINTEST.checkLogin	20 毫秒
✓ LOGINTEST.checkPointer	30 毫秒

```

5: [ OK ] LOGINTEST.checkPointer (0 ms)
5: [-----] 1 test from LOGINTEST (0 ms total)
5:
5: [-----] Global test environment tear-down
5: [=====] 1 test from 1 test suite ran. (0 ms total)
5: [ PASSED ] 1 test.
5: 0

100% tests passed, 0 tests failed out of 5

Total Test time (real) = 0.24 sec
进程已结束,退出代码0

```