

# Messenger

## Step 1

### Console User Interface

Придумать название для мессенджера, который будет создаваться ближайшие месяцы.  
Разработать консольный интерфейс пользователя без реализации функционала.

Интерфейс должен реализовывать следующие possibilities (возможности):

1. Registration - Регистрация пользователя
2. Authorization - Авторизация и аутентификация
3. Add contact - Добавление нового пользователя в список контактов
4. View contacts - Просмотр списка контактов пользователя
5. Send message - Отправка сообщения
6. View chats - Просмотр списка сообщений (всех чатов)
7. View tet-a-tet - Просмотр переписки (сообщений чата с выбранным пользователем)
8. Придумайте еще possibilities, которые стоило бы реализовать в мессенджере.

К коду предъявляются следующие требования:

1. В функции main располагается бесконечный цикл и вызывается единственная функция (CUI), отвечающая за отображение консольного интерфейса. Функция возвращает некоторый код, который зависит от логики работы вашего интерфейса. Выход из цикла осуществляется командой break, когда функция вернет соответствующий код.
2. В функции CUI отображается главное меню мессенджера, в котором можно выбрать одну из possibilities. Для обработки выбранной possibility используйте конструкцию switch case.

3. При выборе possibility вместо главного меню появляются дополнительные поля ввода, необходимые для выполнения данной possibility.

Например, для Registration, требуется ввести login и password.

4. После ввода информации вызывается функция (пока без реализации), которая реализует данную possibility. Каждая такая функция должна возвращать некоторое значение, в зависимости от логики ее работы. В любом случае, следует предусмотреть возврат кода ошибки.

Например, для Registration, функция возвращает id зарегистрированного пользователя, либо код ошибки (число  $\leq 0$ ).

5. После вызова функции, реализующей некоторую possibility, проверяется возвращенное ею значение. Выводится информация об успешном выполнении possibility, либо информация об ошибке.

Например, для Registration, «The user was successfully registered with id 1534», либо «When registering the user, an error occurred»

6. После вывода информации о выполнении possibility программа ждет нажатия любой клавиши на клавиатуре, после чего консоль очищается и появляется главное меню.

Пример возможной реализации CUI представлен в листинге ниже. Это всего лишь пример. Ваша реализация может отличаться.

```

#include <iostream>
#include <string>

int MessSiCUI();
int Registration(std::string login, std::string pass, int day, int month, int year);
int AddContact(int user_id, int friend_id);

//...

int main()
{
    while(true)
    {
        system("cls");
        /*...*/ MessSiCUI();
        //...
        break;
        //...
        system("pause");
    }
}

int MessSiCUI()
{
    std::cout << "MessSiCUI © Semen Igonin===" << std::endl;
    std::cout << "Current User: no user" << std::endl;
    std::cout << "Main menu:" << std::endl;
    std::cout << "\t1. Registration" << std::endl;
    std::cout << "\t2. Authentication" << std::endl;
    std::cout << "\t3. Add Contact" << std::endl;
    //...

    std::cout << "Your choice: ";
    int choice = 0;
    std::cin >> choice;

    system("cls");

    std::cout << "MessSiCUI © Semen Igonin===" << std::endl;
    switch(choice)
    {
    case 1:
        {
            std::string login, password;
            std::cout << "Registration==" << std::endl;
            std::cout << "Login: ";
            std::cin >> login;
            std::cout << "Password: ";
            std::cin >> password;

            int result = Registration(login, password, 1, 1, 2001);
            if(result > 0)
            {
                std::cout << "User with id " << result
                    << " was registered" << std::endl;
            }
            else
            {
                std::cout << "Registration error, code "
                    << result << std::endl;
            }
        }
        break;
    case 2:
        {
            //...
        }
        break;

```

```
        case 3:
        {
            int user_id, friend_id;

            std::cout << "Add contact==" << std::endl;
            std::cout << "user_id: ";
            std::cin >> user_id;
            std::cout << "friend_id: ";
            std::cin >> friend_id;

            int result = AddContact(user_id, friend_id);
            if(result > 0)
            {
                std::cout << "User " << friend_id
                    << " was added to" << user_id << std::endl;
            }
            else
            {
                std::cout << "Add contact error, code "
                    << result << std::endl;
            }
        }
        break;
    //...
    }
    return 0;
}

int Registration(std::string login, std::string pass, int day, int month, int year)
{
    return 1; //registered user id or err_code;
}

int AddContact(int user_id, int friend_id)
{
    return 1; //1 or err_code;
}

//...
}
```