Нижегородский государственный технический университет им. Р.Е. Алексеева.

Институт радиоэлектроники и информационных технологий.

Кафедра «Прикладная информатика и математика».

Документация к курсовой работе по Технологиям программирования.

Выполнили: студенты группы 23-ПМ-1

Антипин Р.

Бровкин Г.

Бушковский Д.

Горохов Д.

Куликов М.

Любимцев П.

Проверил: Крючков А.В.

г. Нижний Новгород

2024 г.

Цели и задачи: Разработать программное обеспечение для психиатрической больницы.

Требования:

1. Приложение должно работать через сервер и клиент;
2. Приложение должно осуществлять управление списком пациентов и докторов (добавление, удаление, редактирование, вывод);
3. Приложение должно иметь следующую информацию:
   1. О пациентах: имя, диагноз, метод лечения;
   2. О докторах: имя, специальность.;
4. Приложение должно выводить об успешной коммуникации сервера.
5. Приложение может использовать стандартные средства для UNIX-систем, например - <socket.h>.

Описание классов, структур и функций:

1. struct Patient - структура, с помощью которой «пациент» записывается в базу данных с своими характеристиками:
   1. int id - идентификация пациента в базе данных;
   2. string name - имя пациента;
   3. string diagnosis - его диагноз;
   4. string treatment - метод лечения диагноза.
2. struct Doctor - структура, с помощью которой «доктор» записывается в базу данных с своими характеристиками:
   1. int id - идентификация доктора в базе данных;
   2. string name - имя доктора;
   3. string specialization - его специализация.
3. class PsychiatryServer - класс, в котором заложена база данных с пациентами и докторами, в ней есть такие функции как:
   1. void loadPatients() - загружает пациентов в базу данных;
   2. void loadDoctors() - загружает докторов в базу данных;
   3. void savePatients() - утверждает пациентов в базе данных;
   4. void saveDoctors() - утверждает докторов в базе данных;
   5. PsychiatryServer() - вызывает конструктор класса;
   6. ~PsychiatryServer() - вызывает его деструктор;
   7. int addPatient(const std::string& name, const std::string& diagnosis, const std::string& treatment) - добавляет пациента;
   8. int removePatient(int id) - удаляет пациента из базы данных;
   9. int editPatient(int id, const std::string& newDiagnosis, const std::string& newTreatment) - редактирует характеристики пациента;
   10. int addDoctor(const std::string& name, const std::string& specialization) - добавляет доктора;
   11. int removeDoctor(int id) - удаляет доктора из базы данных;
4. void \*get\_in\_addr(struct sockaddr \*sa) - функция, позволяющая найти адрес хранения нужной информации;
5. void sigchld\_handler(int s) - функция, с помощью которой находится хэндлер;
6. void connect\_with\_client(int &sockfd) - функция, с помощью которой сервер подключается к клиенту и взаимодействует с ним.