

Реализации модельного SQL-интерпретатора

Сборка - `make` (создание исполняемых файлов client и server)

Удаление исполняемых и временных файлов - `make clean_temp`

Запуск на исполнение -

`<path>server` - для сервера

`<path>client < testi` (на вход передается набор команд) - для клиента

1. Реализация архитектуры «Клиент — Сервер» : один клиент, клиент и сервер на одной ЭВМ.

```
class BaseSocket {
public:
    explicit BaseSocket(int desc = -1, SocketAddress * addr = NULL);
    /* функции чтения и записи */
    int GetSockDescriptor();
    ~BaseSocket();
protected:
    SocketAddress * sock_addr;
};

class ClientSocket : public BaseSocket {
public:
    void Connect();
};

class UnClientSocket : public ClientSocket {
public:
    UnClientSocket(const char *);
};

class ServerSocket : public BaseSocket {
protected:
    void Bind();
    void Listen();
    virtual void OnAccept(BaseSocket *) {}
public:
    BaseSocket * Accept();
};

class UnServerSocket : public ServerSocket {
public:
    UnServerSocket(const char *);
};
```

2. Реализация архитектуры базы данных.

Информация хранится в виде таблиц. Каждая таблица описывается одним файлом данных, состоящим из заголовка и записей.

Объектно-ориентированный интерфейс управления таблицами:

```
class IField {
public:
    virtual Type GetType() = 0;
    virtual long & Long() = 0;
    virtual std::string & Text() = 0;
};

class ITableStruct {
public:
    virtual ITableStruct * AddText(const std::string &, size_t) = 0;
    virtual ITableStruct * AddLong(const std::string &) = 0;
    virtual ITableStruct * SetName(const std::string &) = 0;
};

class ITable {
public:
    static ITable * Create(ITableStruct *);
    static ITable * Open(const std::string &);
    static ITableStruct * CreateTableStruct();
    static void Drop(const std::string &);
    virtual IField * GetField(const std::string &) = 0;
    virtual void Add() = 0;
    virtual void Delete() = 0;
    virtual void Drop() = 0;
    virtual bool ReadFirst() = 0;
    virtual bool ReadNext() = 0;
    virtual void Update() = 0;
    virtual ~ITable() {}
};
```