



ОНЛАЙН-ОБРАЗОВАНИЕ

# Онлайн-образование






# Меня хорошо видно && слышно?

Ставьте  , если все хорошо  
Напишите в чат, если есть проблемы





# Защита проекта

## Тема: ORM



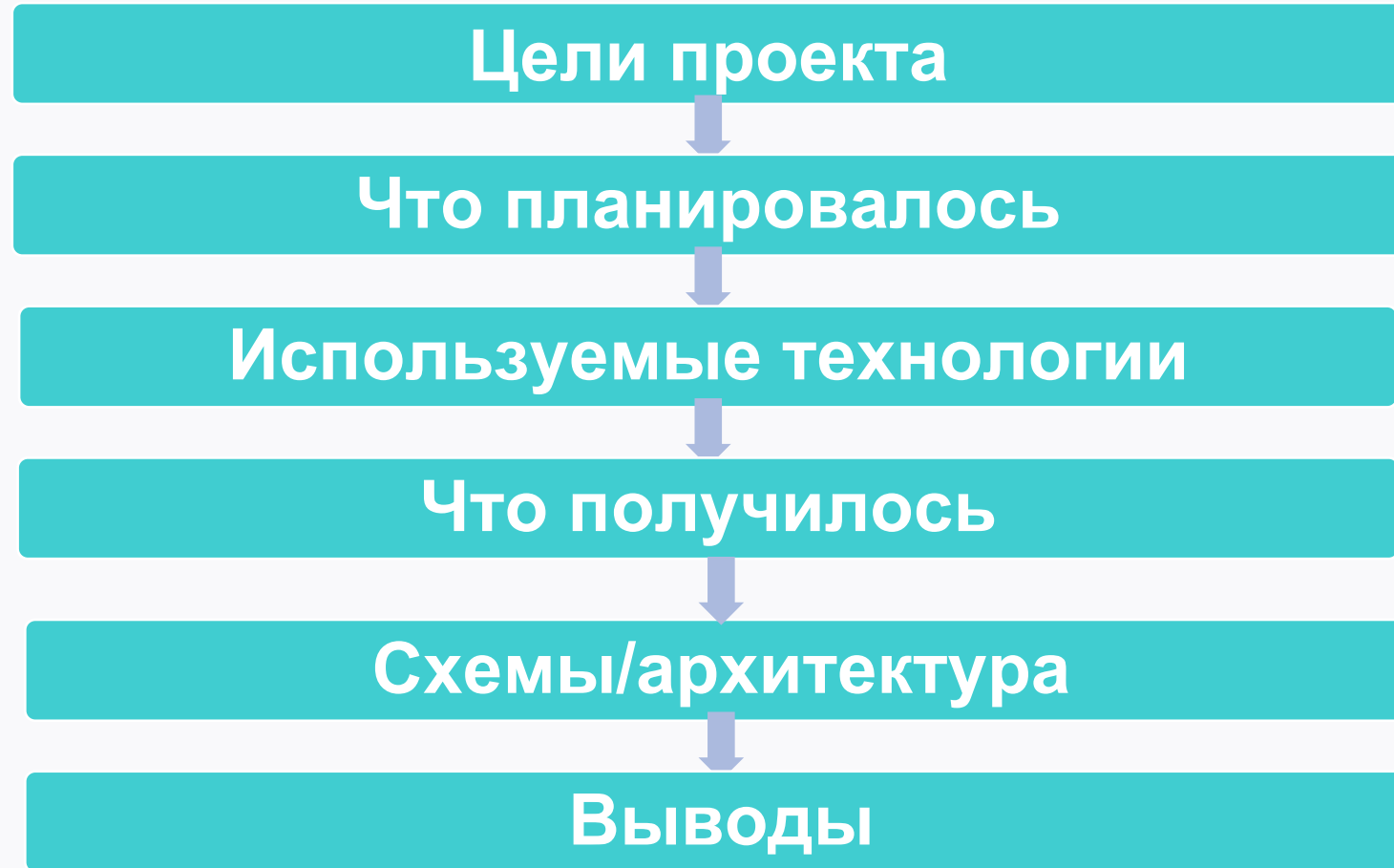
Кругликов Евгений Александрович

инженер-программист

ДМТ Трейдинг



# План защиты



# Цели проекта

1

Разработка инструмента для работы с базами данных, обладающего низким порогом вхождения с точки зрения построения клиентского кода:

- простота подключения в проект
- простота построения клиентского кода
- проверка и контроль клиентского кода на этапе компиляции

2

Разработка расширяемого инструмента для работы с базами данных

# Что планировалось

- 1 Работа с таблицей базы данных как с обычной структурой данных
- 2 Формирование запросов стандартным C-подобным синтаксисом
- 3 Контроль данных компилятором в клиентской части кода
- 4 Гибкость и расширяемость, поддержка множества баз данных
- 5 Простота подключения (изначально планировалась как one-header библиотека) и минимум сторонних зависимостей



# Используемые технологии

1 C++17

2 STL

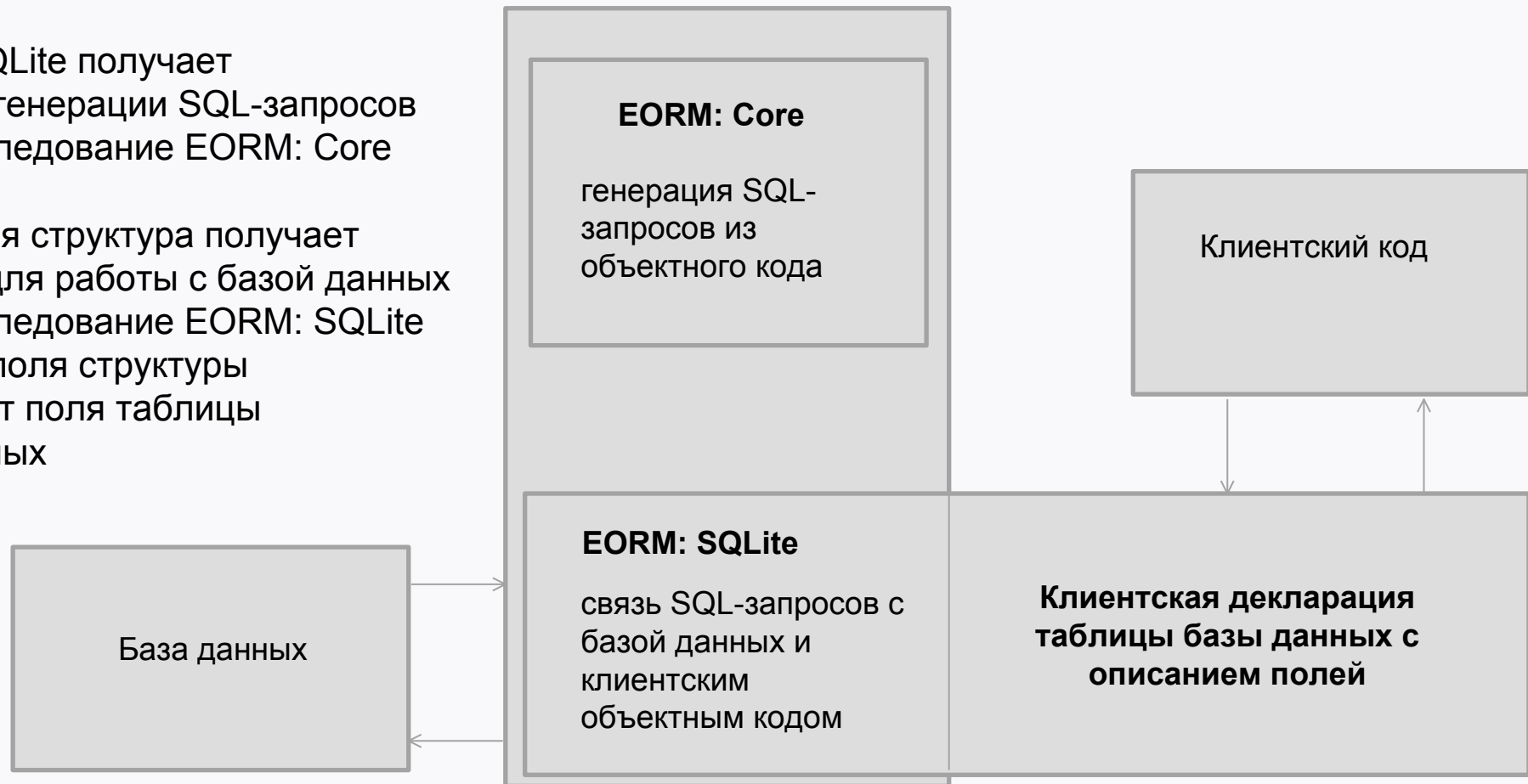
3 SQLite (для работы EORM-SQLite)



# Что получилось

## Основная идея:

- EORM: SQLite получает функции генерации SQL-запросов через наследование EORM: Core
- Клиентская структура получает функции для работы с базой данных через наследование EORM: SQLite при этом поля структуры описывают поля таблицы базы данных





# Что получилось

Декларация таблицы базы данных SQLite

```
9 struct MyTableInfo: SQLiteTable
10 {
11     TableColumn<int> ID;
12     TableColumn<double> C1;
13     TableColumn<std::time_t> C2;
14     TableColumn<std::string> C3;
15     TableColumn<float> C4;
16     TableColumn<float> C5;
17
18     MyTableInfo(const SQLiteDatabase& db): SQLiteTable({"MyTable"}, db)
19     {
20         registerColumns(ID, C1, C2, C3, C4, C5);
21     }
22 };|
23
```

# Что получилось

Декларация таблицы базы данных SQLite с уточнением свойств колонок

```
9 struct MyTableInfo: SQLiteTable
10 {
11     TableColumn<int> ID = { {"ID"}, TCS(TCS::PRIMARY_KEY | TCS::AUTOINCREMENT) };
12     TableColumn<double> C1 = { {"C1"}, TCS::DEFAULT, 3.14 };
13     TableColumn<std::time_t> C2 = { {"C2"}, TCS(TCS::DEFAULT | TCS::NOT_NULL) };
14     TableColumn<std::string> C3 = { TCS::DEFAULT };
15     TableColumn<float> C4 = { 5.65f };
16     TableColumn<float> C5;
17
18     MyTableInfo(const SQLiteDatabase& db): SQLiteTable({"MyTable"}, db)
19     {
20         registerColumns(ID, C1, C2, C3, C4, C5);
21     }
22 };
23
```



# Что получилось

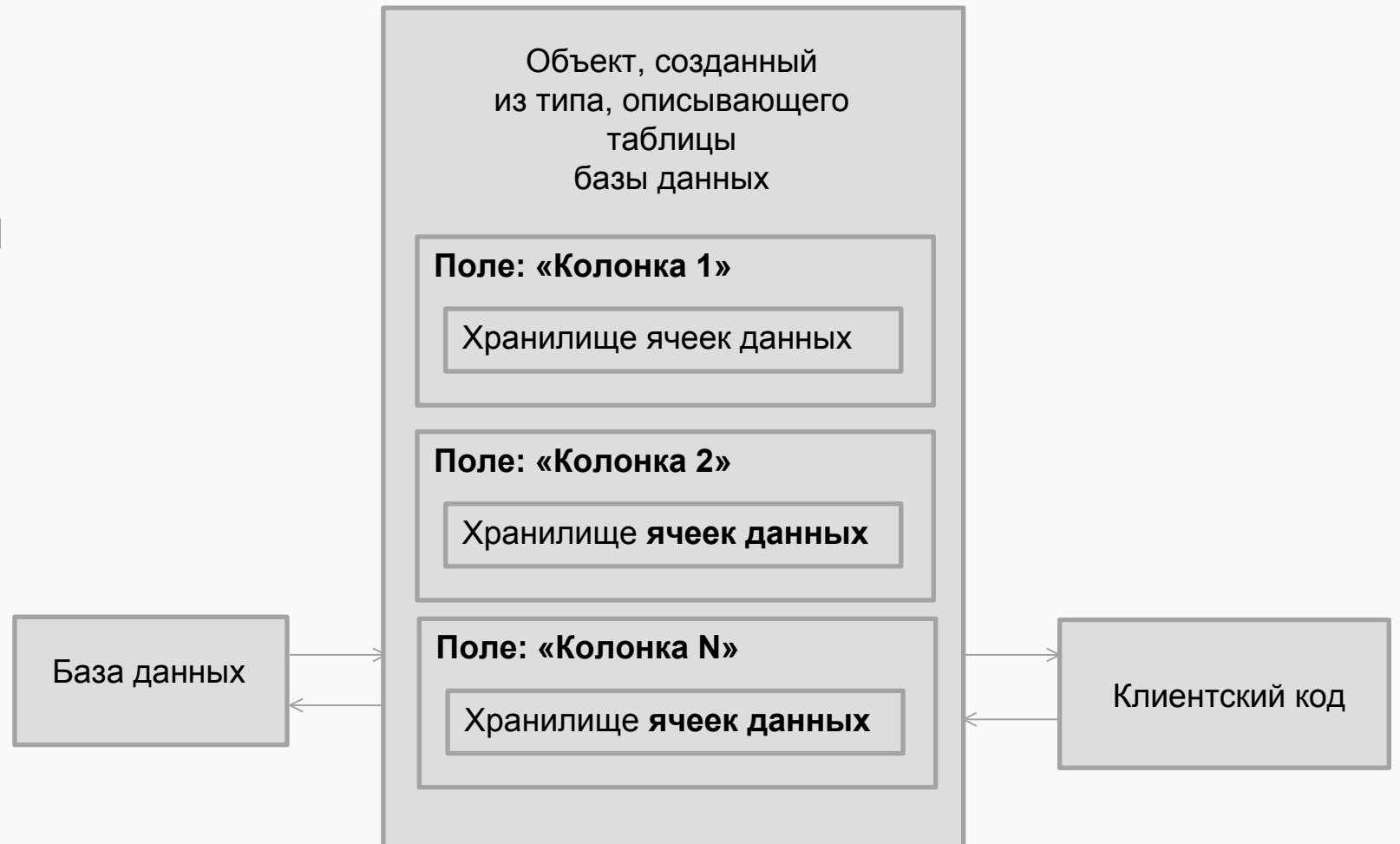
Создание и удаление таблицы в базе данных:

```
30     .... SQLiteDatabase db{"my.db"};  
31     .... MyTableInfo tbl{db};  
32  
33     .... tbl.createTable();  
34  
35     .... // ...  
36  
37     .... tbl.dropTable();
```

# Что получилось

Обмен данными между базой данных и клиентским кодом:

- Свойства объекта таблицы являются полями таблицы базы данных и хранят буфер данных (ячеек)
- Доступ к ячейкам имеет общий вид:  
«Объект таблицы».«Поле»[индекс\_строки]
- Все операции чтения и изменения базы данных производятся через буферы данных свойств объекта таблицы





# Что получилось

Добавление строк в таблицу (операция insert):

```
49     tbl.clearRows();
50
51     tbl.addRow(0, 0.314, time_t(0), "Row-01", 44.1f, 55.23f);
52     tbl.addRow(1, 3.140, time_t(0), "Row-02", 44.1f, 55.23f);
53     tbl.addRow(2, 31.40, time_t(0), "Row-03", 44.1f, 55.23f);
54     tbl.addRow(3, 31.40, time_t(0), "Row-04", 44.1f, 55.23f);
55     tbl.addRow(4, 31.40, time_t(0), "Row-05", 44.1f, 55.23f);
56     tbl.addRow(5, 31.40, time_t(0), "Row-06", 44.1f, 55.23f);
57
58     tbl.insertRows();
```

# Что получилось

Обновление строк в таблице (операция update) и удаление строк (операция delete):

```
81     ...tbl.clearRows();  
82     ...  
83     ...tbl.addRow(0, 11.1, time_t(0), "UPDATED ROW", 22.22f, 33.333f);  
84     ...  
85     ...tbl.updateRows({tbl.ID > 1 && tbl.ID < 5});  
86
```

```
106  
107     ...tbl.deleteRows({tbl.ID > 3});  
108
```



# Что получилось

## Выборка строк из таблицы (операция select):

Пример вызова метода выборки строк из базы данных

```
56 tbl.clearRows();
57
58 tbl.selectRows(
59     [&tbl](const size_t rowIndex)
60     {
61         std::cout << "ROW_INDEX [" << rowIndex << "]: ";
62         if (rowIndex < tbl.ID.getRowsCount()) std::cout << tbl.ID[rowIndex] << "...";
63         if (rowIndex < tbl.C1.getRowsCount()) std::cout << tbl.C1[rowIndex] << "...";
64         if (rowIndex < tbl.C2.getRowsCount()) std::cout << tbl.C2[rowIndex] << "...";
65         if (rowIndex < tbl.C3.getRowsCount()) std::cout << tbl.C3[rowIndex] << "...";
66         if (rowIndex < tbl.C4.getRowsCount()) std::cout << tbl.C4[rowIndex] << "...";
67         if (rowIndex < tbl.C5.getRowsCount()) std::cout << tbl.C5[rowIndex] << "...";
68         std::cout << std::endl;
69     },
70     {}, {}, {}, {}, {}, 0, 0,
71     [] (const size_t rowsCount)
72     {
73         std::cout << "ROWS_COUNT:" << rowsCount << std::endl;
74     }
75 );
```

Результат выполнения

```
ROWS_COUNT:6
ROW_INDEX [0]: 1    0.314    0    Row 01    44.1    55.23
ROW_INDEX [1]: 2    3.14    0    Row 02    44.1    55.23
ROW_INDEX [2]: 3    31.4    0    Row 03    44.1    55.23
ROW_INDEX [3]: 4    31.4    0    Row 04    44.1    55.23
ROW_INDEX [4]: 5    31.4    0    Row 05    44.1    55.23
ROW_INDEX [5]: 6    31.4    0    Row 06    44.1    55.23
```

# Что получилось

Общий синтаксис на примере параметров функции выборки:

- список выбираемых колонок
- присоединения (операция join)
- условие выборки
- условие сортировки
- лимит выборки
- отступ выборки

```
339 (
340     {
341         Table1.ID,
342         Table1.C1,
343         Table1.C2
344     },
345     {
346         { TJT::INNER, Table2, { Table1.ID == Table2.ID } },
347         { TJT::OUTER, Table3, { Table1.C2 == Table2.C2 } },
348         { TJT::LEFT, Table4 },
349         { TJT::RIGHT, Table5 },
350     },
351     {
352         Table1.C1 == 3.14
353         && Table1.C2 > Table1.C3
354         && Table1.C3 == SqlName{"MyCustomColumn"}
355         && Table1.C4.IN( Table6.getSqlRowsSelect() )
356     }
357     ,
358     {
359         { Table1.C1, TRST::DEFAULT },
360         { Table1.C2, TRST::ASC },
361         { Table1.C3, TRST::DESC }
362     },
363     300,
364     25
365 );
```



# Вывод и планы по развитию

Запланируйте пару минут на рефлекссию в конце защиты проекта и расскажите о планах по развитию







Спасибо за внимание!

фото

Напишите ваше ФИО

Должность

Компанию

Контакты