

Еремеев Евгений Михайлович

начальник лаборатории

Научный исследовательский испытательный

центр средств ПВО межвидового назначения

г. Знаменск, Астраханская область

СПОСОБЫ ПРИМЕНЕНИЯ ИНСТРУМЕНТАРИЯ КРОССПЛАТФОРМЕННОЙ БИБЛИОТЕКИ QT ДЛЯ РАБОТЫ С БАЗАМИ ДАННЫХ ПРИ СОЗДАНИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Аннотация: в статье рассмотрены основные пути совершенствования информационной составляющей системы объективного контроля полигонных испытаний на основе современных инструментальных систем разработки программного обеспечения.

Ключевые слова: кроссплатформенная библиотека, база данных, информационная система.

В процессе развития системы полигонных испытаний и в особенности ее информационной составляющей в значительной степени повысилась роль инструментальных средств разработки программного обеспечения сбора, обработки и отображения результатов экспериментов системы объективного контроля (СОК).

Программное обеспечение СОК достаточно сложное и должно соответствовать многим требованиям. Кроме пользовательских требований, налагаемых на удобство и возможности программного продукта, есть и другие требования, касающиеся разработки программного обеспечения. Большую роль здесь играют средства, которыми программист пользуется в процессе своей работы. Во многих случаях бывает удобно владеть инструментарием, который имеет достаточно широкую область применения и может служить для решения большого количества задач разного масштаба: от построения небольших программ для создания мощных программных комплексов. Также часто возникает вопрос о поддержке

нескольких программных платформ т.к., ориентируясь только на одну платформу Windows, можно потерять большое количество потенциальных пользователей в тех случаях, когда возникает необходимость перехода на другие операционные системы.

В настоящее время наиболее широко используется инструментарий разработки программных средств библиотеки Qt (последняя версия Qt5.4).

Qt расширяет возможности программиста с помощью набора макросов, метаинформации и сигнально-слотовых соединений, но использует при этом лишь средства языка C++ и является совместимым со всеми распространёнными современными его компиляторами.

Для разработки программ с использованием библиотеки Qt была создана интегрированная среда разработки Qt Creator. Её первая версия была представлена одновременно с официальным выпуском Qt 4.5.0. Это полноценная кроссплатформенная среда для создания новых проектов и работы с ними.

В настоящее время программное обеспечение (ПО) в интересах информационного обеспечения СОК совершенствуется по пути создания информационно-аналитической системы обеспечения полигонных испытаний на основе существующих кроссплатформенных инструментальных средств, таких как Qt.

Широкие возможности библиотеки Qt тем не менее ограничены возможностью приобретения и использования баз данных для создания информационно-аналитической системы объективного контроля.

В целом вся система объективного контроля предназначена для регистрации данных, обработки исходных данных (данных ВСИ, ТИ, РТИ), проведения совместного и событийного этапов обработки и предоставления результатов обработки в виде разнообразных электронных таблиц, диаграмм, графиков, удобных для проведения углубленного анализа результатов испытаний ВВТ и их документирования. Хранение и обработку такого объема информации может обеспечить СУБД среднего класса.

Исходя из характера задач и возможностями приобретения базы данных, наиболее приемлемыми в существующих условиях являются СУБД InterBase компании Borland и СУБД MySQL.

В библиотеке Qt имеются драйверы для работы со многими существующими системами управления базой данных (СУБД). Список используемых в библиотеке Qt драйверов приведен в таблице 1.

Таблица 1

Список драйверов библиотеки Qt

Драйвер	База данных
QDB2	IBM DB2 версии 7.1 и выше
QIBASE	InterBase компании Borland
QMYSQL	MySQL
QOCI	Oracle
QODBC	ODBC(SQL Server)
QPSQL	PostgreSQL версий 6.x и 7.x
QSQUTE	SQLite версии 3 и выше
QSQUTE2	SQLite версии 2
QTDS	Sybase Adaptive Server

Одной из особенностей использования драйверов Qt является то, что из-за лицензионных ограничений не все драйверы входят в состав издания Qt с открытым исходным кодом (QtOpenSourceEdition). При настройке конфигурации Qt драйверы InterBase и MySQL можно либо непосредственно включить в состав Qt, либо использовать как подключаемые модули (plugins). Qt поставляется вместе с SQLite -общедоступной, не нуждающейся в сервере базой данных.

Qt также позволяет легко программировать такие распространенные идиомы баз данных, как отображение зависимых представлений для записей, связанных отношением «главная-подчиненные» (master-detail), и возможность многократной детализации выводимых на экран данных (drill-down).

Одной из особенностей работы с Qt является соединение с базой данных и выполнение запросов. Для выполнения запросов в InterBase и MySQL необхо-

димо сначала установить соединение с базой данных. Обычно настройка соединений с базой данных выполняется отдельной функцией, которую вызывают при запуске приложения.

Например:

```
– QSqlDatabase *db = QSqlDatabase::addDatabase(«QOCI8»);
– db-> setHostName(«mozart.коп ко rdia.edu»);
– db-> setDatabaseName(«musicdb»);
– db-> setUsername(«gbatstone»);
– db-> setPassword(«T17aV44 »);
– db-> lastError().showMessage();
– return false.
```

После установки соединения можно применять `QSqlQuery` для выполнения любой инструкции SQL, поддерживаемой используемой базой данных. Ниже приводится пример выполнения команды `SELECT`:

```
– QSqlQuery query;
– query. execr(«SELECT title, year FROM cd WHERE year >= 1998»);
```

После вызова функции `exec()` можно посмотреть результат запроса:

```
– QString title = query.value(0).toString(); int year = query.value(1).toInt();
– cerr «qPrintable(title)» " «year»«end»;
```

Другой особенностью работы с Qt является табличное представление данных. Во многих случаях табличное представление является самым простым представлением набора данных для пользователей. При работе с СУБД `InterBase` и `MySQL` необходимо воспользоваться классом `QSqlTableModel` и его подклассом `QSqlRelationalTableModel`. Класс `QSqlTableModel` и его подкласс `QSqlRelationalTableModel` используются для просмотра и взаимодействия пользователей с данными, хранимыми в базе данных.

Некоторые базы данных не поддерживают внешние ключи. В этом случае необходимо убрать фразы `Foreign Key`. Информационная система будет работать, но база данных не будет поддерживать целостность данных на уровне ссылок.

Еще одна особенность работы с Qt заключается в создании форм по технологии «master-detail». Конструктор создания форм начинается с настройки модели `SqlRelationalTableModel`, которая управляет таблицей.

Вызов `set Relatione` указывает модели на то, что ее поле содержит идентификатор внешнего ключа из таблицы и что вместо идентификаторов необходимо выводить на экран содержимое соответствующего поля.

Если пользователь переходит в режим редактирования этого поля (например, нажимая клавишу F2), модель автоматически выведет на экран поле с выпадающим списком имен, и если пользователь выбирает другое поле, то таблица будет обновлена.

Для пользователей, предпочитающих иметь дело с высокоуровневым интерфейсом базы данных, который не требует знания синтаксиса SQL, классы `QSqlTableModel` и `QSqlRelationalTableModel` являются наиболее приемлемыми. Эти классы представляют таблицы SQL в том же виде, как и классы других моделей Qt. Они могут использоваться самостоятельно для кодирования в программе просмотра и редактирования данных или могут подключаться к представлениям, с помощью которых конечные пользователи будут сами просматривать и редактировать данные.

Рассмотренные особенности работы с кроссплатформенной библиотекой Qt позволят разработчикам информационно-аналитической системы решить наиболее важные проблемы при работе с базами данных и, в частности, проблемы разработки пользовательского интерфейса.

Список литературы

1. Земсков Ю.В. Программирование на C++ с использованием Qt4 / Ю.В. Земсков. – Волгоград: ZI-soft, 2007. – 346 с.
2. Алексеев Е.Р. Программирование на языке C++ в среде QtCreator / Е.Р. Алексеев, Г.Г. Злобин. – М.: ALTLinux, 2015. – 247 с.
3. Камаев В.А. Современные технологии программирования / В.А. Камаев, В.В. Костерин. – 2-е изд. – М.: Высшая школа, 2006. – 453 с.