

Далее для сравнения производительности принято решение сравнения времени запросов к базе данных с разным количеством записей. Сравнение производилось на одном и том же компьютере с одинаковой конфигурацией. В качестве переменных, которые варьировались в запросе были выбраны: N подразделений и M — количество сотрудников. Распределение сотрудников по подразделениям было хаотичным. Сравнение времени выполнения вложенного запроса в зависимости от количества записей представлено на рисунке 4.

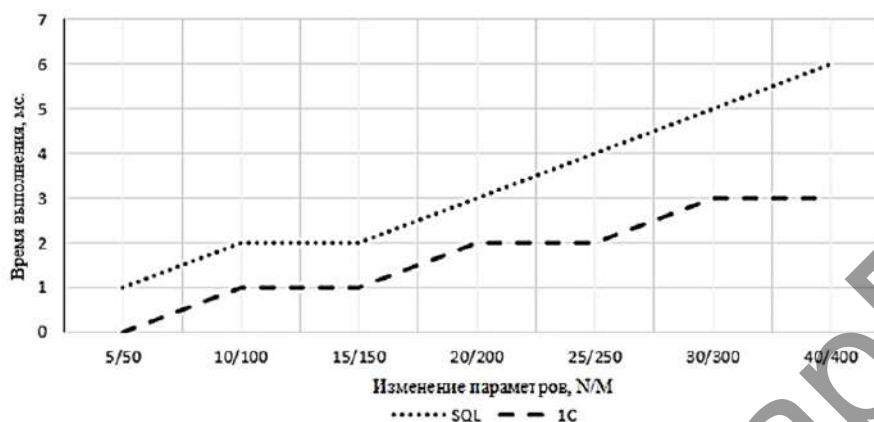


Рисунок 4 — Диаграмма сравнения времени выполнения вложенного запроса

При сравнении производительности запросов выявлено преимущество языка запросов 1С.

Заключение. Результатом исследования являются выявленные отличительные черты каждого из языков запросов 1С и SQL.

Выявлены сходства в следующих операторах: выборки, источников выборки, условий отбора, операторы агрегирования и группировки данных, объединения запросов, вложенных запросов. То есть операторы имеют одинаковое написание, за исключением возможности двуязычного написания в 1С.

Выявлены следующий различия:

- 1С предоставляет возможность обращение к свойствам объектов через точку или несколько точек;
- в 1С есть обращение к табличным частям объектов, что в SQL должно было быть представлено двумя и более сущностями;
- программный запрос 1С не позволяет изменять объекты, в отличие от языка SQL, в котором есть оператор ALTER и UPDATE, которые используются для изменения баз данных и изменения строк сущностей, соответственно. В свою очередь изменение данных в 1С происходит при помощи объектных запросов;
- время выполнения запросов на 1С меньше, чем на SQL, что может объясняться разной конструкцией полей (в SQL поля объединены связями, а в 1С — ссылками, что предоставляет более скорый доступ к полям).

Список цитируемых источников

1. Анализ данных и процессов : учеб. пособие / А. А. Барсеян [и др.] — 3-е изд., перераб. и доп. — СПб. : БХВ-Петербург, 2009. — 512 с.
2. Кузнецов, С. Д. Основы баз данных : учебное пособие / С. Д. Кузнецов — 2-е изд., испр. — М. : Интернет-Университет Информ. Технологий ; БИНОМ. Лаб. знаний, 2007. — 484 с.

УДК 004.75

Д. С. Кислый, Г. М. Раковцы

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПРОВЕДЕНИЯ ШКОЛЬНЫХ ОЛИМПИАД ПО ПРОГРАММИРОВАНИЮ

Введение. Ежегодно проводятся олимпиады по программированию среди школьников и студентов в различных учреждениях образования в несколько этапов. При этом организация такого мероприятия неизбежно связана с обработкой большого количества бумажной информации. Большая часть действий является общей для всех подобных олимпиад: регистрация участников, тестирование их программ и подведение итогов,

а в некоторых случаях и подачей студентом или учащимся апелляции по вопросу полученных баллов. Все это, как правило, осуществляется вручную, а позже и набором протокола в электронном виде, однако, это требует больших временных затрат, а также не исключает ошибки, вызванные человеческим фактором.

Основная часть. Целью данного исследования является разработка информационной системы для автоматизации проведения школьных олимпиад по программированию.

Для осуществления обозначенной цели необходимо осуществить следующие задачи:

1. Провести анализ предметной области.
2. Выбрать методы и инструменты разрабатываемого программного обеспечения.
3. Создать базу данных разрабатываемого программного обеспечения.
4. Выполнить реализацию программного продукта.
5. Выбор метода шифрования данных.
6. Провести отладку и тестирование программного продукта.

Для разработки проекта будет использоваться среда разработки Microsoft Visual Studio, которую можно использовать для различных аспектов разработки программного обеспечения. Для написания программного обеспечения будет использоваться язык программирования C#.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данный продукт позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight [1].

Для разработки базы данных будет использоваться инструмент проектирования баз данных — SQLite.

SQLite — это встраиваемая кроссплатформенная база данных, которая поддерживает достаточно полный набор команд SQL и доступна в исходных кодах (на языке C). Её притязанием на известность является как собственно сам движок базы, так и её интерфейс (точнее его движок) в пределах одной библиотеки, а также возможность хранить все данные в одном файле. Позиция функциональности SQLite где-то между MySQL и PostgreSQL. Однако, на практике, SQLite нередко оказывается в 2-3 раза (и даже больше) быстрее. Такое возможно благодаря высокоупорядоченной внутренней архитектуре и устранению необходимости в соединениях типа «сервер-клиент» и «клиент-сервер» [2].

Преимущества данной СУБД:

1. Является одной из популярных СУБД.
2. Безопасность к данным.
3. Консольная утилита для работы с базами.
4. Открытые исходные коды.

Недостатки данной СУБД:

1. Нет хранимых процедур.
2. Нет встроенной поддержки UNICODE
3. Не подходит для приложений, которые часто обращаются к базе данных.

SQLite поддерживает динамическое типизирование данных.

В приложении используется база данных, содержащая следующие таблицы:

1. «Роли» — содержит информацию об ролях в разрабатываемом программном продукте: учащиеся, студент, учитель, преподаватель, администратор (код_роли, роль).
2. «Авторизация» — содержит информацию об авторизованных субъектах системы (логин, пароль, фио, номер_телефона, время_авторизации, код_роли).
3. «Области» — содержит всю информацию об областях (код_области, область).
4. «Город» — содержит всю информацию об городах и в каких областях они находятся (код_города, город, код_области).
5. «Школа» — содержит информацию о школах и в каких городах они находятся (код_школы, вид, номер_школы, код_города).
6. «Класс» — содержит информацию о классах и может иметь примечание: профиль класса если имеется (класс, примечание).
7. «Тип задания» — содержит информацию о типе задания: текстовая задача или тестовое задание (код_типа, тип_задания).
8. «Задание» — содержит информацию о номере, описания задания, ответа и к какому типу относится то или иное задание (номер_задания, описание, ответы, код_типа).
9. «Участники» — содержит всю информацию об участниках олимпиады: из какого они города, школы и класса (код_участника, фио, код_рождения, код_школы, класс, фио_учителя).
10. «Результаты» — содержит информацию о прохождении олимпиады участником: номер задания, который участник отправил и выставление баллов за выполненную задачу (код_участника, номер_задания, отве, время_прохождения, балл).

На рисунке 1 представлена физическая модель разрабатываемой базы данных, которая отражает взаимосвязь таблиц, ключевые поля и типы данных в таблицах.

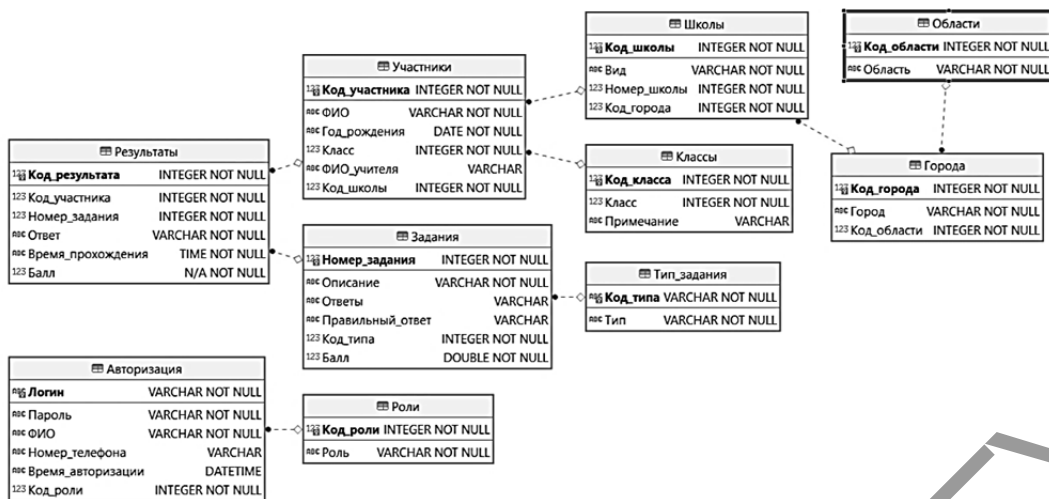


Рисунок 1 — Физическая модель базы данных

Данное разрабатываемое приложение является клиент-серверным и представляет собой следующий функционал, описанный ниже.

Функции клиента:

- 1) осуществляет авторизацию;
- 2) просматривает задания;
- 3) отправляет решения задачи;
- 4) просмотр результатов.

Функции сервера:

- 1) запуск олимпиады;
- 2) добавление задания;
- 3) осуществляет удаленное подключение к любому пользователю;
- 4) остановка олимпиады;
- 5) проверка правильности решения задания и составление отчета.

Взаимодействие между клиентом и сервером осуществляется по средству созданного программного продукта. На рисунке 2 изображено соединение клиента и сервера. Для входа необходимо ввести данные. Эти данные проверяются в базе данных, после чего пользователь войдет в систему.

The screenshot shows two windows:

- Авторизация**: Contains fields for 'Login:' and 'Password:', and a 'Войти' (Login) button.
- Главное меню**: Contains sections for 'Данные сервера' (Server Data) and 'Данные участника' (Participant Data).
 - Данные сервера**: Includes 'IPAddress:' (192.168.100.69) and 'Port:' (22).
 - Данные участника**: Includes 'ФИО:' and 'Класс:' fields.
 A 'Начать выполнение' (Start Execution) button is located at the bottom.

Рисунок 2 — Форма входа клиента и сервера

Данный программный продукт использует протокол TCP (протокол управления передачей) — один из основных протоколов передачи данных интернета, предназначенный для управления передачей данных. Для защиты данных используется хэширование данных [3].

Заключение. В ходе исследования была разработана система автоматизации проведения школьных олимпиад по программированию с использованием возможностей языка C# в среде Visual Studio. Данная система значительно упростит проверку решенных заданий участников, что в значительной степени сэкономит время ожидания результатов и подведения итогов.

Список цитируемых источников

1. Microsoft Visual Studio [Электронный ресурс].— Режим доступа : https://www.tadviser.ru/index.php/Microsoft_Visual_Studio/. — Дата доступа : 11.04.2021.
2. SQLite [Электронный ресурс]. — Режим доступа : <https://ru.bmstu.wiki/SQLite>. — Дата доступа : 11.04.2021.
3. Шапович, Е. Г. Системы сокрытия и шифрования информации с использованием стеганографии / Е. Г. Шапович, А. В. Шах // Сборник трудов IV Международной научно-практической конференции, 2018. — С. 101—104.

УДК 004.94

М. А. Кононович

Учреждение образования «Барановичский государственный университет», Барановичи, Республика Беларусь

ОТНОШЕНИЯ МЕЖДУ КЛАССАМИ И ОБЪЕКТАМИ НА ПРИМЕРЕ ИГРОВОГО ПРИЛОЖЕНИЯ

Введение. Компьютерные технологии развиваются с огромной скоростью. В современном мире компьютеры помогают в работе, используются в сфере образования, экономики, игр, досуга, позволяют людям легко связываться друг с другом.

Объектно-ориентированное программирование — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования [1].

Существует много разных типов отношений, которые два объекта могут иметь в реальной жизни. Для описания этих отношений используются определенные слова для их описания, например, является, имеет, использует, зависит, является частью и т.д. Все эти типы отношений имеют свои аналоги в виде отношений в объектно-ориентированном программировании.

Основная часть. Между классами возможны два типа отношений:

1. Отношение типа is-a (есть, является), при котором один класс есть подвидом другого класса. При таком отношении один класс расширяет (детализирует) возможности другого класса. Расширение возможностей класса осуществляется благодаря использованию наследования.

2. Отношение, при котором существует взаимосвязь между двумя классами. Здесь выделяют два под-вида взаимосвязи между классами:

– отношение типа has-a (класс содержит другой класс). В этом случае в классе объявляется один или несколько экземпляров другого класса. При данном отношении возможны два случая взаимодействия. Первый случай, это когда объект (экземпляр), который объявлен в классе, не является составной частью класса (агрегация) и его использование не влияет на функциональную работу класса. Второй случай, когда объект, который объявлен в классе, есть составной частью этого класса (композиция);

– отношение типа uses (класс «использует» другой класс). В этом случае класс содержит программный код другого вложенного класса, к которому он имеет доступ [2].

В качестве объекта исследования выбраны отношения между классами и объектами в объектно-ориентированном программировании. В качестве предмета исследования выбран игровой движок Unreal Engine 4. Целью исследования выступает анализ отношений между классами в реализованной игре.

Unreal Engine 4 — это игровой движок и редактор, разработанный компанией Epic Games для создания игр и приложений от мультиплатформенных игр с миллионными бюджетами до мобильных инди-разработок. Unreal Engine работает на операционных системах Windows и macOS и разработанные в нем проекты могут быть развернуты на платформах Windows, Mac, PlayStation 4, Xbox One, iOS, Android, HTML5 и Linux. В самом простом формате Unreal Engine 4 является коллекцией редакторов, используемых в различных стадиях производства игр или приложений [3].

Для анализа отношений между классами была создана игра графического изображения аквариума на языке программирования C++. Приложение позволяет управлять персонажем в виртуальный 3D мире аквариум. Пользователь может перемещаться в пространстве и рассматривать виртуальный 3D мир.

При создании проекта были созданы и использованы следующие основные классы:

1. UObject — базовый класс, от которого наследуется подавляющее большинство объектов.
2. Landscape — класс, позволяющий создать ландшафт. При необходимости изменять его размеры, добавить углубления и возвышенности и т. д. (рисунок 1).
3. Mar — класс для генерации объектов пространства. Генерация объектов пространства представлена на примере создания и расположения объекта «сундук» (рисунок 2).
4. Light — класс, предназначенный для подсчёта освещения на карте.
5. Character — класс, позволяющий создать персонажа, установить определенные его настройки, а также прописать логику его действий (рисунок 3).