

Литература

- 1 Вагнер, Б. Эффективное программирование на C#. 50 способов улучшения кода / Б. Вагнер. – М. : Вильямс И. Д., 2017. – 224 с.
- 2 Бычков, А. Дизайн и фриланс. Новый уровень / А. Бычков; АСТ, 2019. – 210 с.
- 2 Гриффитс, И. Программирование на C# 5.0 / И. Гриффитс. – М. : ЭКСМО, 2016. – 208 с.
- 3 Крейтон, Р. X. Unity Game Development Essentials / Р. X. Крейтон Packt Publishing, 2010. – 83 с.
- 4 Паласиос, Хорхе Unity 5.x. Программирование искусственного интеллекта в играх / Хорхе Паласиос. – М. : ДМК Пресс, 2016. – 849 с.

А. С. Городков, Е. А. Ружицкая
(ГГУ им. Ф. Скорины, Гомель)

РЕАЛИЗАЦИЯ ТРАНЗАКЦИЙ В ПРИЛОЖЕНИИ «АВТОМАТИЗИРОВАННАЯ СИСТЕМА УПРАВЛЕНИЯ БАНКОМ ПО РАБОТЕ С КЛИЕНТАМИ»

Базы данных обрабатывают миллионы одновременных запросов в секунду. Во многих случаях эти запросы могут затрагивать одну и ту же запись в базе данных. При неправильной обработке таких случаев целостность данных системы может быть нарушена. В качестве примера из банковской сферы можно привести ситуацию, когда необходимо обработать несколько параллельных операций на изменение баланса счета клиента или обработать операцию по переводу средств с одного счета на другой. Для сохранения согласованности данных при выполнении таких операций используются транзакции базы данных.

Транзакция базы данных – это последовательность нескольких операций, выполняемых в базе данных, которые служат единой логической единицей работы, а именно выполняются полностью или не выполняются вовсе.

В разработанном приложении для управления банком используется система управления базами данных MS SQL Server. MS SQL Server поддерживает обработку транзакций, а также все уровни изоляции транзакций, определенные в спецификации SQL. Для реализации банковской системы был выбран самый высокий уровень изоляции транзакций – serializable (последовательно чтение).

Уровень `serializable` обеспечивает самую строгую изоляцию транзакций. Он эмулирует последовательное выполнение транзакций для всех зафиксированных транзакций. Минусом использования уровня изоляции `serializable` являются частые блокировки транзакций. Например, если две параллельные транзакции прочитают баланс клиента, а потом попытаются его обновить, то эти транзакции будут заблокированы и их выполнение завершится ошибкой.

В случае блокировки транзакции пользователь получит ошибку и ему придется вручную несколько раз повторять операцию, которая приводит к изменению баланса. Для улучшения пользовательского опыта в приложении используется `retry pattern` – механизм повтора неудавшихся транзакций. Таким образом, пользователю не придется вручную несколько раз повторять неудачные операции.

М. Д. Горошко, Р. А. Новиков
(ГрГУ им. Я. Купалы, Гродно)

ДЕЦЕНТРАЛИЗОВАННЫЙ ИНТЕРНЕТ БУДУЩЕГО WEB 3.0

Интернет был создан более 30 лет назад и с тех пор прошел долгий путь развития. В первой версии интернета (Web 1.0) пользователи могли только читать информацию на веб-страницах, а во второй версии (Web 2.0) интернет стал более динамичным и интерактивным, что привело к созданию множества социальных сетей, облачных сервисов и других онлайн-приложений.

Однако Web 2.0 имеет свои недостатки, такие как централизованный контроль над данными, уязвимость к хакерским атакам и нарушениям приватности. Децентрализованный интернет будущего Web 3.0 обещает решить эти проблемы и создать новый интернет, основанный на технологии блокчейна и децентрализации.

Web 3.0 – это концепция следующего поколения интернета, где пользователи могут контролировать свои данные, безопасно обмениваться информацией и управлять своими онлайн-активами. Одним из главных достоинств Web 3.0 является децентрализация. В отличие от Web 2.0, где данные хранятся на централизованных серверах, в Web 3.0 данные хранятся на блокчейне, что делает их более безопасными и защищенными от хакерских атак и нарушений приватности.