**Функция addOrder**

Функция addOrder обрабатывает процесс добавления заказа, включая генерацию уникального штрих-кода, отправку данных в стороннюю API для бронирования и подтверждения заказа, и сохранение заказа в базе данных с использованием транзакции.

**Параметры:**

* $event\_id (int): Идентификатор события.
* $event\_date (string): Дата и время события, на которое были куплены билеты.
* $ticket\_adult\_price (int): Цена взрослого билета на момент покупки.
* $ticket\_adult\_quantity (int): Количество купленных взрослых билетов.
* $ticket\_kid\_price (int): Цена детского билета на момент покупки.
* $ticket\_kid\_quantity (int): Количество купленных детских билетов.

**Алгоритм:**

1. Рассчитывается итоговая сумма заказа на основе количества и стоимости билетов.
2. Генерируется уникальный штрих-код с помощью generateUniqueBarcode.
3. Осуществляется попытка забронировать заказ через сторонний API (sendBookingRequest). Если API возвращает ошибку о дублировании штрих-кода, генерируется новый штрих-код и запрос повторяется.
4. Если бронирование успешно, отправляется запрос на подтверждение заказа (sendConfirmationRequest).
5. При успешном подтверждении заказа данные сохраняются в базе данных с использованием функции saveOrderToDatabase, которая включает транзакцию для обеспечения целостности данных.

**Функция generateUniqueBarcode**

Генерирует случайный восьмизначный штрих-код. Этот код используется для идентификации заказа.

**Возвращаемое значение:**

* Возвращает строку, представляющую собой восьмизначное случайное число.

**Функция sendBookingRequest**

Эта функция имитирует отправку данных о бронировании в сторонний API и возвращает случайный результат.

**Параметры:**

* $event\_id, $event\_date, $ticket\_adult\_price, $ticket\_adult\_quantity, $ticket\_kid\_price, $ticket\_kid\_quantity, $barcode

**Возвращаемое значение:**

* Возвращает замокированный массив с ключом message (успешный ответ) или error (ошибка бронирования).

**Функция sendConfirmationRequest**

Эта функция имитирует подтверждение бронирования у внешнего API и возвращает случайный результат.

**Параметры:**

* $barcode (string): Уникальный штрих-код заказа.

**Возвращаемое значение:**

* Возвращает замокированный массив с ключом message (успешный ответ) или error (ошибка подтверждения).

**Функция saveOrderToDatabase**

Сохраняет данные заказа в базе данных с использованием транзакции.

**Параметры:**

* $event\_id, $event\_date, $ticket\_adult\_price, $ticket\_adult\_quantity, $ticket\_kid\_price, $ticket\_kid\_quantity, $barcode, $equal\_price.

**Алгоритм:**

1. Подключение к базе данных осуществляется с использованием PDO.
2. **Начало транзакции** ($pdo->beginTransaction()): Транзакция начинается для обеспечения целостности данных.
3. **Вставка данных заказа**: Выполняется SQL-запрос на вставку данных заказа в таблицу orders.
4. **Подтверждение транзакции** ($pdo->commit()): Если данные успешно добавлены, транзакция подтверждается, записывая изменения в базу данных.
5. **Откат транзакции** ($pdo->rollBack()): В случае возникновения ошибки транзакция откатывается, предотвращая частичное добавление данных в таблицу.

**Исключения:**

* В случае ошибки при выполнении SQL-запроса происходит откат транзакции.