Design a Parking Lot

Starting an object-oriented design interview for a parking lot scenario involves several key steps to ensure a structured and thorough approach. Here's a suggested outline:

1. **Clarify Requirements:**
    - Begin by asking clarifying questions to fully understand the requirements and constraints of the parking lot system. This may include:
        - The size and capacity of the parking lot.
        - Types of vehicles allowed (e.g., cars, motorcycles, trucks).
        - Parking rules and regulations (e.g., reserved spaces, handicapped spots).
        - Payment methods and pricing models.
        - Operational considerations (e.g., entry/exit points, security measures).
        - Any additional features or functionalities required.
2. **Identify Objects and Responsibilities:**
    - Based on the requirements gathered, identify the main objects and their responsibilities within the parking lot system. This may include:
        - ParkingLot: Representing the parking lot itself.
        - Vehicle: Representing different types of vehicles.
        - ParkingSpace: Representing individual parking spaces.
        - Ticket: Representing parking tickets issued to vehicles.
        - Entrance/Exit: Representing entry and exit points.
        - PaymentSystem: Handling payment processing.
        - SecuritySystem: Managing security measures (e.g., surveillance).
    - Define the attributes and behaviors (methods) of each object.
3. **Define Relationships:**
    - Establish relationships between the identified objects. For example:
        - ParkingLot has ParkingSpaces and Entrance/Exit points.
        - Vehicle occupies ParkingSpace(s) and obtains Ticket(s).
        - PaymentSystem interacts with Ticket(s) to process payments.
        - SecuritySystem monitors ParkingLot and Entrance/Exit points.
    - Determine the multiplicity and cardinality of relationships (e.g., one-to-one, one-to-many).
4. **Design Class Diagram:**
    - Create a class diagram to visually represent the relationships between objects. Use UML notation to depict classes, attributes, methods, and associations.
    - Ensure that the class diagram accurately reflects the identified objects and their interactions.
5. **Consider Design Patterns:**
    - Evaluate whether any design patterns (e.g., Factory, Singleton, Strategy) are applicable to optimize the design and address specific requirements.
    - Integrate relevant design patterns into the class diagram as needed.

6. **Discuss Trade-offs and Scalability:**
    - Discuss any trade-offs involved in the design decisions made, such as performance vs. simplicity, flexibility vs. efficiency, etc.
    - Consider how the design can accommodate future scalability and extensibility requirements.
7. **Validate and Iterate:**
    - Validate the design with the interviewer, ensuring that it aligns with the requirements and addresses all key aspects of the parking lot system.
    - Be prepared to iterate on the design based on feedback and further discussions with the interviewer.

By following this structured approach, you can effectively tackle an object-oriented design interview question for designing a parking lot system, demonstrating your ability to analyze requirements, identify objects, define relationships, and create a well-structured design.

Revision #1
Created 27 February 2024 04:52:07 by victor
Updated 27 February 2024 04:52:15 by victor