Interview Preparation Questions and Answers

Q: Explain the principles of OOP (Encapsulation, Inheritance, Polymorphism, Abstraction).

A: Encapsulation: Hiding implementation details while exposing only the necessary functionalities.

Inheritance: Reusing existing class properties in new classes.

Polymorphism: A single interface for different underlying forms (e.g., method overloading and overriding).

Abstraction: Hiding complex details and showing only the essentials.

Q: What is the difference between `abstract class` and `interface`?

A: Abstract Class: Can have both abstract and concrete methods; supports single inheritance.

Interface: Can only have abstract methods (before Java 8) and supports multiple inheritance.

Q: How is memory managed in Java? Explain garbage collection.

A: Java uses automatic garbage collection to manage memory. Objects no longer referenced are automatically removed to free up memory.

Q: How does exception handling work in Java? Can you create a custom exception?

A: Exception handling uses `try`, `catch`, and `finally` blocks to handle runtime errors. Custom exceptions can be created by extending the `Exception` class.

Q: What is the `final` keyword, and where can it be applied?

A: `final` prevents modification:

- `final` variable: Value cannot be changed.
- `final` method: Cannot be overridden.
- `final` class: Cannot be extended.

Q: Explain the difference between an array and a linked list with examples.

A: Array: Fixed size, fast random access.

Linked List: Dynamic size, slower due to pointer traversal.

Q: Implement Bubble Sort or Binary Search in Java.

A: Bubble Sort: Compare adjacent elements and swap if out of order.

Binary Search: Divide and conquer by comparing mid-value with target.

Q: What is the time complexity of your Flappy Bird or Alien Game logic?

A: Depends on implementation, typically O(n) for simple game loops.

Q: Describe the data structures you used in your Hotel Management System project.

A: Used arrays/lists for temporary storage and relational databases for persistence.

Q: What is a binary search tree? How is it used for searching?

A: Binary Search Tree: Binary tree where left nodes < root and right nodes > root. Efficient for searching with O(log n) complexity.

Q: How did you connect to the database in your Hotel Management System?

A: Used JDBC API with `DriverManager.getConnection` method.

Q: Explain the use of `PreparedStatement` vs `Statement`. Why is `PreparedStatement` preferred?

A: `PreparedStatement` prevents SQL injection and allows parameterized queries, making it safer and more efficient.

Q: What are transactions in JDBC? How would you implement one in your project?

A: A transaction ensures a group of operations either complete fully or not at all. Use `connection.setAutoCommit(false)` and `commit`/`rollback` methods.

Q: How do you handle SQL exceptions in your project?

A: Used `try-catch` blocks to log errors and provide user-friendly messages.

Q: Explain the use of `ResultSet` in your project. How did you retrieve and display data?

A: `ResultSet` is used to fetch data from SQL queries. Used `resultSet.next()` to iterate over rows.

Q: What is the structure of a JavaFX application? Explain the role of `Stage` and `Scene`.

A: `Stage` represents the main window. `Scene` contains UI elements organized in layouts.

Q: How did you implement animations in your Flappy Bird or Alien Game?

A: Used JavaFX `Timeline` and `Animation` classes to create smooth movements.

Q: What is the difference between `Canvas` and `Pane` in JavaFX?

A: `Canvas` is used for custom drawings, while `Pane` is a general-purpose container for UI controls.

Q: How do event handlers work in JavaFX? Can you give an example from your games?

A: Event handlers are assigned to UI components to respond to user actions like mouse clicks or key presses.

Q: How would you make your JavaFX project platform-independent?

A: JavaFX is already cross-platform. Ensure no platform-specific APIs or file paths are hardcoded.

Q: How did you handle concurrency in your games? Did you use threads?

A: Used separate threads for animations and game logic to prevent UI freezing.

Q: Explain the difference between `Runnable` and `Thread`.

A: `Runnable` is a functional interface for defining a task. `Thread` is a class to manage execution.

Q: What are `wait()`, `notify()`, and `notifyAll()` in Java? Where could you use them in your projects?

A: Used for thread communication. Example: Pause and resume game states.

Q: How can you achieve thread safety in JavaFX animations?

A: JavaFX uses a single UI thread. Use `Platform.runLater` for thread-safe UI updates.

Q: What are the advantages and challenges of using threads in game development?

A: Advantages: Smooth gameplay.

Challenges: Race conditions, debugging complexity.

Q: What database did you use for your Hotel Management System? Why?

A: Used MySQL for its reliability and integration with Java JDBC.

Q: Explain the `AUTO_INCREMENT` keyword and how it was used in your project.

A: Generates unique IDs automatically for primary keys.

Q: How would you query to find the room details for a specific reservation?

A: `SELECT * FROM reservations WHERE id = ?` using `PreparedStatement`.

Q: Write an SQL query to update a guest's phone number in your database.

A: `UPDATE reservations SET mobile_no = ? WHERE id = ?;`

Q: How would you handle large datasets in SQL to avoid performance issues?

A: Use indexing, optimize queries, and apply pagination with `LIMIT` and `OFFSET`.

Q: If asked to extend your Hotel Management System, what features would you add?

A: Features like room availability tracking, payment gateway integration, and user roles (admin, guest).

Q: How would you refactor your Flappy Bird Game to improve performance or readability?

A: Optimize physics calculations, modularize code, and reduce redundant rendering.

Q: What were the biggest challenges you faced during your project development?

A: Debugging, ensuring thread safety, and managing database connections efficiently.

Q: How would you optimize your JavaFX animations for better performance?

A: Reduce unnecessary rendering and use `Canvas` for complex animations.

Q: Can you explain the flow of data in your Hotel Management System from the UI to the database?

A: UI sends user input to the backend, which processes the data and interacts with the database using JDBC.