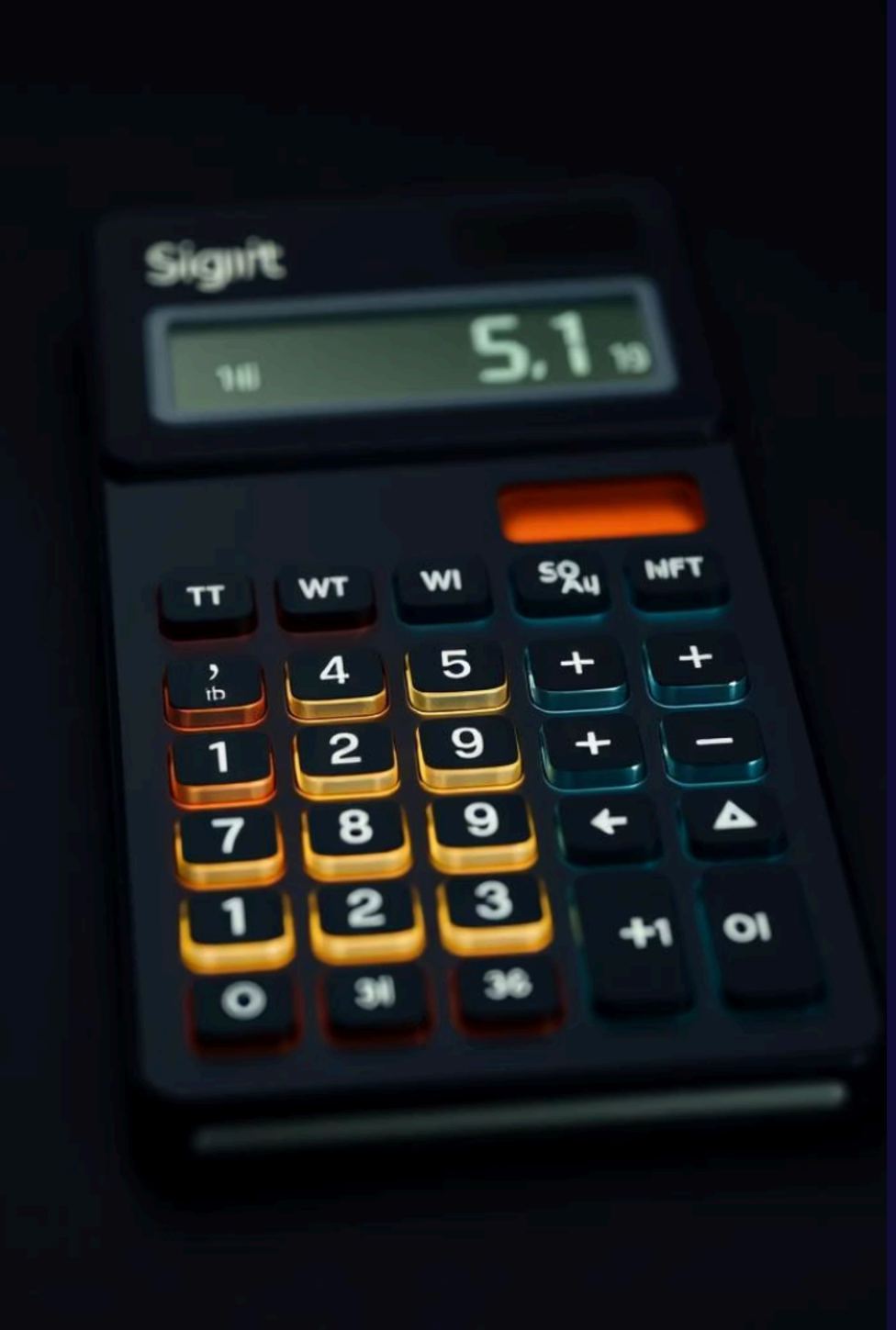


Scientific Calculator: From Concept to Creation with Method Overloading

This presentation explores developing a scientific calculator using method overloading.

We'll cover project setup, database design, JDBC, and UI development.

Project Setup: JDK, IDE, and Defining the Structure

JDK Installation

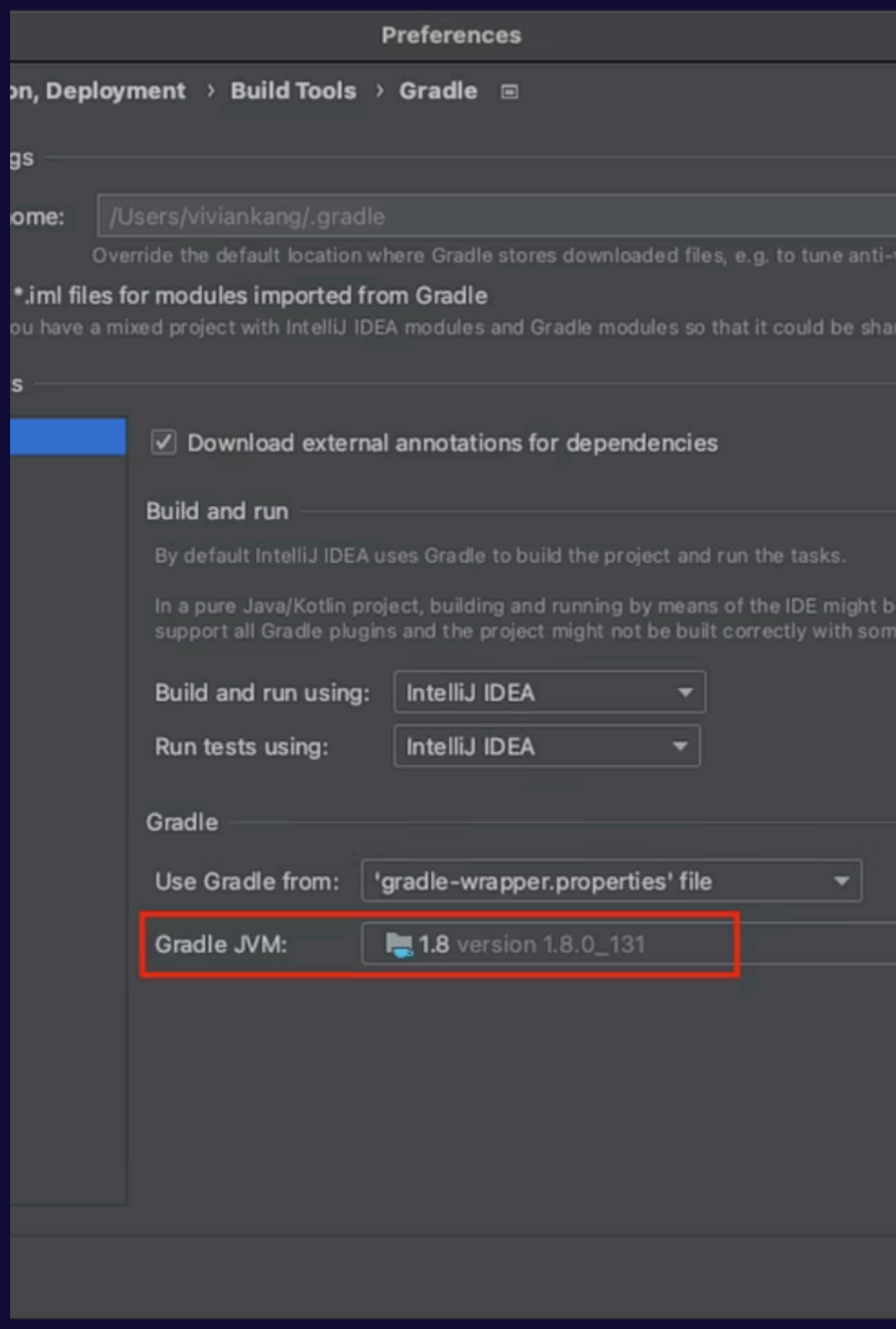
Choose a JDK version compatible with your project needs.

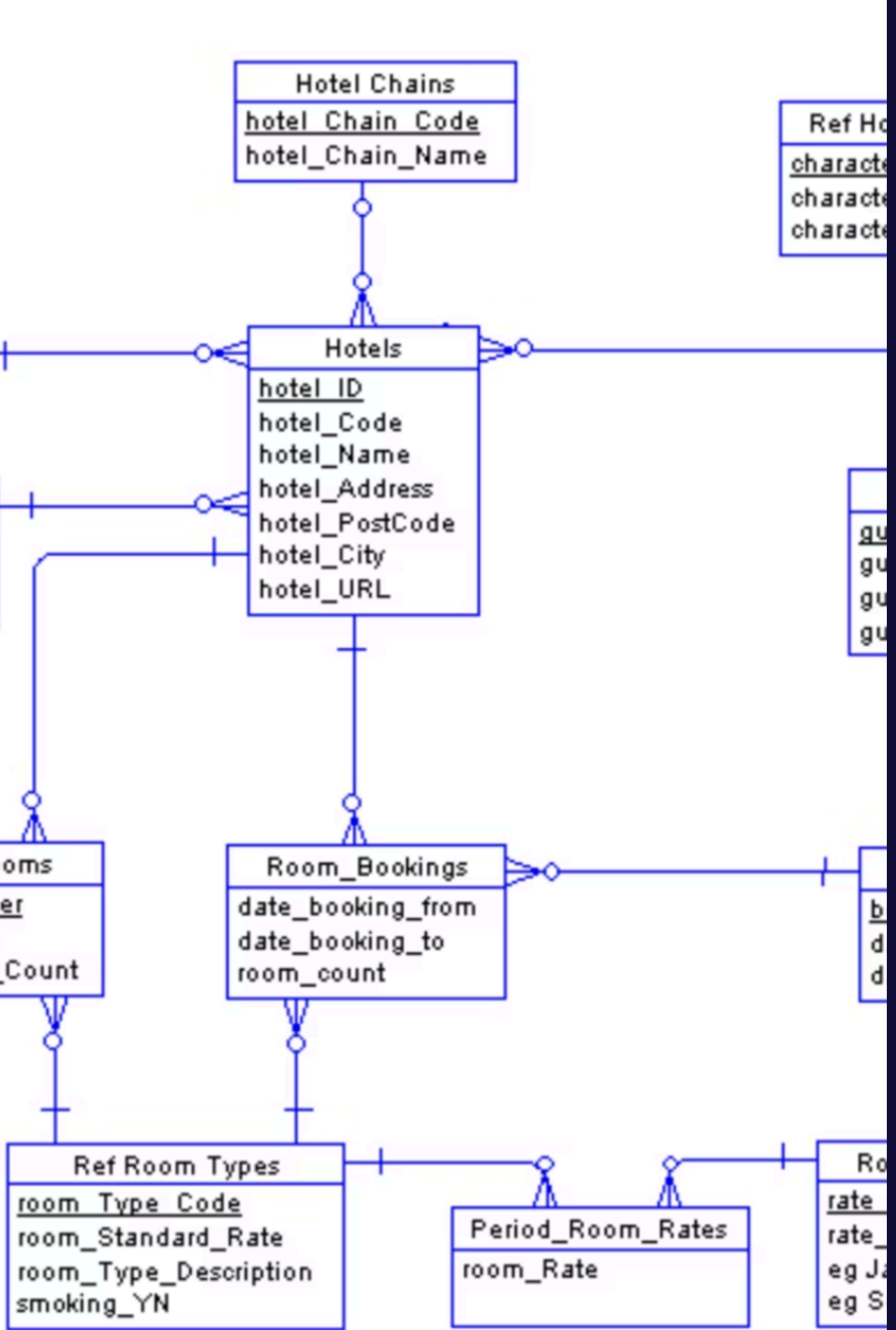
IDE Setup

Use Eclipse, IntelliJ, or NetBeans for efficient development.

Project Structure

Organize packages clearly: controllers, models, views, and utilities.





Database Design: Schema and MySQL Table Creation

Schema Planning

Define tables to store operations, user preferences, and history.

MySQL Table

Create tables with appropriate data types and keys.

Normalization

Ensure data integrity by minimizing redundancy.

JDBC Implementation: Connecting to the Database

1

Load Driver

Register MySQL JDBC driver using Class.forName()

2

Establish Connection

Connect to database using DriverManager and credentials.

3

Execute Queries

Use Statements and PreparedStatements for secure data manipulation.

4

Handle Exceptions

Catch SQLExceptions to manage database errors gracefully.

```
cernbitalse(lulli,)):
uneh fusecall beconttaltists)()))
```

Model and DAO Classes: Database Operation Logic

Model Classes

Represent database tables with Java objects encapsulating fields.

DAO Classes

Manage CRUD (Create, Read, Update, Delete) operations through JDBC.

UI Aesthetics: Creating a Visually Appealing Interface

Color Scheme

Use neutral backgrounds with accent colors for buttons.

Typography

Choose legible fonts with appropriate sizing for readability.

Visual Feedback

Highlight button presses and calculation results dynamically.

UI Component Placement: Alignment and User Experience

Button Grouping

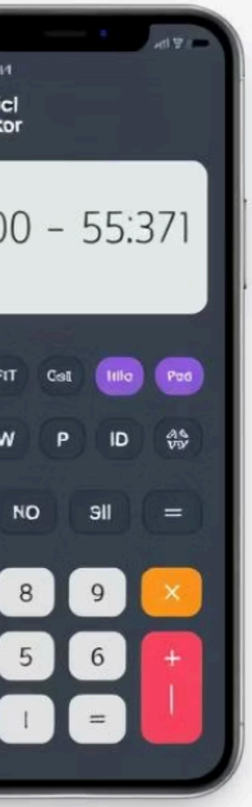
Group numeric keys, operators, and functions logically.

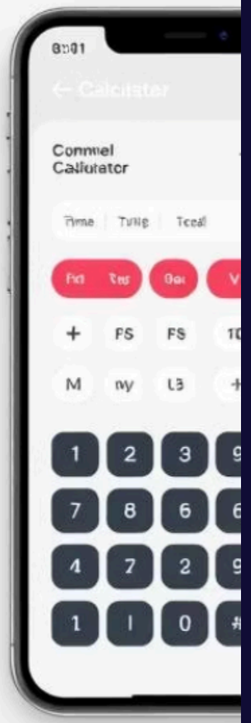
Alignment

Consistent spacing and alignment enhance ease of use.

Navigation

Arrange components to follow natural hand movement.





UI Responsiveness and Accessibility: Reaching a Wider Audience

Responsive Design

Adapt layout for desktop, tablet, and mobile screen sizes.

Keyboard Accessibility

Enable keyboard input for all functions and shortcuts.

Screen Reader Support

Use ARIA labels and semantic elements for assistive technology.


Thank
you