

Lessons Learned of Mini2



Yi Cao (Andrew Id: yc2)

Spring of 2016

Lessons Learned of 18642

yi cao (andrew id :yc2)

Abstract

Mini 2 of Java smart phone(18641) mainly focused on the android development. Through mini2, I applied OO design concepts, Database normalization, and of course android basics. Besides, in the backend, I also implemented Servlet as my server and deployed it on AWS. For this article, in the first part, I would introduce lessons learned in mini 1 would be applied in mini 2; in the second part, I would introduce lessons learned in android development; in the third part, I would introduce overall understanding.

PART 1: Lessons Learned 1

For this part, I would introduce lessons learned in mini2 that would be used in mini 2.

1. what is object: Objects can be modeled according to the needs of the application. An object may have a physical existence or a conceptual existence.
2. what is class: A class represents a collection of objects having same characteristic properties that exhibit common behavior. It includes the operations and attributes of object.
3. what is encapsulate: It is the process of binding both attributes and methods together within a class. Through encapsulation, the internal details of a class can be hidden from outside. And customers can only access the elements of class outside.
4. what is inherent: It permits new classes to be created out of existing classes by extending and refining its capabilities.
5. what is polymorphism: a object could be implemented in many forms but with a common interface. Inheritance would be one of the important way to implement polymorphism.
6. what is association: It describes the relations between objects with common structure or common behavior.
7. what is the key points of OOD: low coupling and high cohesion.
8. what role(s) does an interface play in building an API: Interface encapsulates all operations of the API, but it doesn't have function body to explain the operations in a detail. It has to be implemented by proxy classes to explain the details of the operations
9. Facade design pattern: The BuildAuto has no specific operations inside but it is an outer interface for customers to call and it almost contain all CRUD operations for models. It is a kind of "Facade Design Pattern". Customers can only call functions to operate but do not the implementation details. It is good to maintain the safety of the whole system.
10. What is http protocol: Http is a protocol based on TCP. User enters information into form fields or controls. The browser then packages the data, opens an HTTP connection and sends the data to a server or an email address.
11. What is servlet and why we need servlet: Servlets provide a component-based, platform-independent method for building Web-based applications.

12. What is database normalization: Database normalization is the process of organizing the columns and tables of a relational database to minimize data redundancy.
13. What is transaction ACID:
 - ATOMICITY: A transaction should be done or undone completely and unambiguously
 - CONSISTENCY: A transaction should preserve all the invariant properties (such as integrity constraints) defined on the data
 - ISOLATION: Each transaction should appear to execute independently of other transactions
 - DURABILITY: The effects of a completed transaction should always be persistent.
14. How to select the primary key and foreign key of relation db: as for project1, I choose the tables' id as primary key or foreign key, because id is unique and would be valid to avoid conflicts
15. How to delete a relation db with cascade: if we delete the upper db which serves as reference for lower db, then the lower db would be deleted as the upper db deleted.

PART 2: Lessons Learned 2

For this part, I would introduce lessons learned in android development.

16. What is android :Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.
17. What is android api level: API Level is an integer value that uniquely identifies the framework API revision offered by a version of the Android platform.
18. What is activity: An activity represents a single screen with a user interface,in-short Activity performs actions on the screen.
19. What is service: A service is a component that runs in the background to perform long-running operations.
20. What is broadcast receivers: Broadcast Receivers simply respond to broadcast messages from other applications or from the system.

21. What is content provider: A content provider component supplies data from one application to others on request.
22. What is fragment: A Fragment represents a behavior or a portion of user interface in an Activity.
23. What is adapter: Android adapter is used to transform the data into view
24. Why we need database normalization: Normalization involves decomposing a table into less redundant tables without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.
25. What are the common steps to design database:
- *Determine the data to be stored in the database.
 - *Determine the relationships between the different data elements.
 - *Superimpose a logical structure upon the data on the basis of these relationships
26. Why we'd better to create table from text file: we read commands from text file or properties file to create tables because we can adjust the process of creating table easily by changing content of input file.
27. How to addresses the needs of exception handling and recovery: we could write a package to contain all customer exceptions and its handling or recovery.
28. What is relation table: relation table mainly means tables have foreign key and reference key and so that those table would not be that redundant.
29. Strategies to design core classes so that they are reusable, extensible and easily modifiable: We could use interface or abstract classes to make our design reusable, extensible and easily modifiable;
30. what is the disadvantage of reading file like text files or database in a single pass and not use intermediary buffering: If we read file in a single pass, the speed of accessing data would be much slower than using intermediary buffering.
31. Why we have to define the primary key for a table: If we have a primary key, then we could use the primary key as the unique identity for the row element, it would avoid conflicts.
32. Why we need activity:Activity is part of the application and could be used to dictate the UI and handle the user interaction to the smart phone screen. With Activity, we could implement different actions with smart phone like dial phone

number, search map information , send messages and so on. One activity can also interact with other activities to achieve certain goals. They run in the back stack and they have lifecycle. Without activity, we can not conveniently interact with smart phone. We may have to bridge the users' operations and the smart phone. However, without the help of activity, it might be quite hard to make machine understand the meaning of users' operations.

33. What is the lifecycle of activity: The Android OS uses a priority queue to assist in managing activities running on the device. Based on the state a particular Android activity is in, it will be assigned a certain priority within the OS. This priority system helps Android identify activities that are no longer in use, allowing the OS to reclaim memory and resources. `onCreate()`: Called to create the activity. `onStart()`: Called when the activity is becoming visible to the user. It is followed by `onResume()` if the activity comes to the foreground, or `onStop()` if it becomes hidden. `onResume()`: Called when the activity will start interacting with the user. Always followed by `onPause()`.

`onPause()`: Called as part of the activity lifecycle when an activity is going into the background, but has not (yet) been killed.

`onStop()`: Called when the activity are no longer visible to the user. You will next receive either `onRestart()`, `onDestroy()`, or nothing, depending on later user activity. `onRestart()`: Called after your activity has been stopped, prior to it being started again. Always followed by `onStart()`.

`onDestroy()`: The final call you receive before your activity is destroyed. This can happen either because the activity is finishing (someone called `finish()` on it, or because the system is temporarily destroying this instance of the activity to save space. You can distinguish between these two scenarios with the `isFinishing()` method.

34. What is android volley: Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley is available through the open AOSP repository.

35. What is android SDK: The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications.

36. What is android studio: Android Studio is the official IDE for Android app development, based on IntelliJ IDEA

37. What is android manifest file: The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code.
38. Android development is primarily done in Java, which is great because it is already an established, stable language with a large community around it in addition to the Android community itself.
39. What is Singleton: In our app, we have android singleton class called “activity” to initiate our SDK when we first open our app.
40. What is android JSON: In our design, we try to use JSON as our format to access data so that our data could be processed more quickly.

PART3: Overall Conclusion

41. Why we need database normalization: Normalization involves decomposing a table into less redundant tables without losing information; defining foreign keys in the old table referencing the primary keys of the new ones. The objective is to isolate data so that additions, deletions, and modifications of an attribute can be made in just one table and then propagated through the rest of the database using the defined foreign keys.
42. What kind of data type can be saved in db: db can save almost all kinds of data type, the important thing is how to save those datatypes , how to index them and how to query them.
43. Why we have to define the primary key for a table: If we have a primary key, then we could use the primary key as the unique identity for the row element, it would avoid conflicts.
44. How Synchronization works in JVM: JVM puts a lock on the monitor of the object or the piece of the synchronized code, which gives it exclusive access to that part of code or object.
45. How can android bridge the communication between front-end and backend: We can use volley to bridge the communication.
46. How can android realize data exchange between activities: We can use asynchronous way to save data in interfaces and then we could get the data from those interfaces
47. How can android bridge the communication between fragments: We could navigation view to direct the app to different fragments
48. How can android use servlet as its server: We could use servlet get and post method to save and process data in the backend database

49. What are the common steps to design database:

- *Determine the data to be stored in the database.

- *Determine the relationships between the different data elements.

- *Superimpose a logical structure upon the data on the basis of these

50. What kind of data type can be saved in db: db can save almost all kinds of data type, the important thing is how to save those datatypes , how to index them and how to query them.

Overall:

Through this course, I got familiar with android development and I implemented lessons learned in Mini1 to Mini2. After this course, I got very interested in android development as well as the details of Java language.

Finally, I want to say thanks for all instructors' work and help!