Lessons Learnt From 18-641 Project 2

Da Wang, dawang(andrew id)

1. What is the first thing I need to do to start a brand new project?

We have several team meetings and brainstorm several ideas. Then we analyze each one and find the most proper idea to start out project.

2. What is Obejct Theory?

In android system, different activities are different object. So are the framents. They use similar structure to organize the data and classes to make the whole program easy to understand and modify.

3. What is Inner Class? Why should we use it?

Inner class is the class inside another class. In this project, `Note` class used to be the inner class. However, as the modification of our design, we decided not to use inner class.

4. What is containment for classes?

Basically, containment means one class is the compoment of another class. In different activities, they contain different models such as `Note`, `User`, `Comment`.

5. What is encapsulation for classes?

Encapsulation can be regarded as a method for modular design. Mostly, we use encapsulation to hide the complexity of different classes as well as protect data from inproper access. For example, the `Request` class encapsulate all the network operation so that we can use them easily.

6. How to design the core data structures so that they can be highly reusable and extensible?

The key of this issue is modularity. Different parts of the same object should not be highly related. It's better to break a big object into different small objects and each object only minds its own business.

As the logic of our app is quite clear, different models can be separated easily without repetition.

7. Why should we use Serialization?

As a way to make objects persistent, Serialization is a good way to protect your data. If we store everything in plain text, the data will be of high risk as plain text files are so easy to be modified or deleted. Also, serialization plays an important part in network.

In our project, we use serialization to transfer the images between server and client.

8. What is the importance of drawing class diagrams?

It is difficult to read other people's code from scratch. But the class diagrams can be the maps for other people. They can build a basic concept of your project with the help of the class diagrams as they contain different classes and the relationships among them.

In our project, class diagram is included in the first stage so that every member can easily get the whole structure.

9. Why good coding style can make code better

With the help of package, the organization of different codes can also be treated like different objects. That is to say, highly related codes should be put into the same package and every package has its own target and only cares about itself.

Also a unified structure for different classes can help people understand the code quickly. What's more, the comment should be consice.

In our project, we create different util classes to do the repeated work. As it is, the code for different acitvities can be quite consice.

10. How to test the data as well as the program?

The easist way to test the program is outputing every detail of the objects and operations. We can also use some of the famous unit test framework such as JUnit.

In our project, we make input for user as less as possible and test different function with basic case, corner case as well as invalid input case.

11. What is the relationship between containment and encapsulation (as applied in this project), when building components?

From my opinion, containment is a way to implement encapsulation. When two different classes can be regardes as "has-a" relationship. Then we can say that one class contains another. Different ways of containment helps organize the data structure to become encapsulation.

In different activities, they contain different models such as `Note`, `User`, `Comment`.

Also, the `Request` class encapsulate all the network operation so that we can use them easily.

12. What are some ways to analyze data (presented in requirements) to design Objects?

The most common way is the top-down method. First we have the whole design called `Note` and then we break the big class into small classes as User`and`Comment‘.

We can also classify different data by its usage. For example, the `Request` class only cares about loading and saving object. It can be regarded as design by function.

13. What are good conventions for making a Java class readable?

There are several conventions to achieve readability:

1. Declaration first, method second.
2. Private first, public second.
3. Static first, non-static second.
4. Each line can only have at most 80 characters.
5. Class description, method description as well as key variable description.
6. Keep constant with the format.

In our project we create different class with the same conventions.

14. What are the advantages and disadvantages of reading data from sources such as text files or databases in a single pass and not use intermediary buffering?

Loading all the data in a single pass is an efficient way as IO operations takes time so it is better to put the together and finishs it once for all. However, sometimes we don't have that much memory so it may cause overflow in the memory. If the data set is large, it is better to use intermediary buffering.

In our project, as we create a different thread to do the IO operation, we choose to fetch them once for all.

15. What issues can occur, when using Serialization with Inner classes?

Inner class is a nested class that is not explicitly or implicitly declared static.Serialization of inner classes (including local and anonymous classes) is error prone.

Serailizing an inner class declared in a non-static context that contains implicit non-transient references to enclosing class instances results in serialization of its associated outer class instance.

In our project we use a 3rd party toolkit to handle different exceptions.

16. Where can inheritance be used?

When you get a "Is-a" relationship between two different classes, inheritance can be used so that the child class can be a specific type of the parent class. For example, `BaseRequest` can be the parent class for `GetNoteRequest`, `RegisterRequest`, `LoginRequest`.

17. Where can polymorphism be used?

Use the example from the last question, as the `GetNoteRequest`, `LoginRequest` are all child class for `BaseRequest` they can have the same function call `execute` but with different behavior. For polymorphism, we can use `BaseRequest` variable to call the `execute` method. The object itself will find the proper method to execute.

18. How can you design objects, which are self-contained and independent?

These two property means that an object should have all the data and methods to get its own staff done. For designing, it is always a safe choice to use the top-down method. Seperate big class into small parts and keep them independent.

In this project we design `Note` object using this manner.

19. How to handle Exceptions from different classes?

First we use try catch method to handle most of the exception. Other will be handled in the corresponding acitivities.

20. Why and when should we use abstract class?

When different concepts can be derived from the same concept but we do not need the origin concept, it is the great time to use abstract class. Abstract class can be treated as a implicit connection between different object to have a unified behavior.

Use the example from the last question, as the `GetNoteRequest`, `LoginRequest` are all child class for `BaseRequest` they can have the same function call `execute` but with different behavior.

21. What is the limitations of interface?

Interface is just another way to help different classes to have the similar behavior. However, it is so convinient that sometimes we may break the border for OOP design because of it. So it's better to use interface wisely and carefully.

In this project, the googlemap interface is easy to use and we don't have to the detail of how other class works.

22. What role(s) does an interface play in building an API?

Interface in Java is like a protocal that classes should follow. One class can inherit several interfaces and the user will know what the class can do.

In our project, we use GoogleMap api to show the maps and markers. Before using it, we must implement its corresponding interface so as to provide enough information. So interface can be used as a rule to regularize api.

23. What is the best way to create a framework, for exposing a complex product, in a simple way and at the same time making your implementation extensible?

Using Interface is a great way to achieve this goal. With a set of good designed interfaces, different classes can cooperate well without making the logic of the program complex.

In this project, we separate different functions to different package and within each package, we try our best to make the classes self-contained.

24. What is the advantage of exposing methods using different interfaces?

It is simple to implement and highly extensible. Different interfaces can be regarded as different sets of related functions that help to keep the code organized.

For example, we can show different data in a listview without implementing different classes but use the listviewadapter interface.

25. How to hide code in abstract class?

Just like the `BaseRequest` class, the other `Request` class extends the abstract class without having to code anything as everything is done in the abstract class!

In the base class, we store the server information and basic interface so that we don't need to repeat them in the children classes.

26. Is there any advantage of creating an abstract class, which contains interface method implementations only?

It sets a clear border for internal code and external code. Also, as an abstract class can not be instantiated, this mechenism makes sure that no internal instance will be built.

In our project, the `BasicRequst` class is the class meets the need above.

27. How can you create a software architecture, which addresses the needs of exception handling and recovery?

We can create the exception management class instead of using default try catch clause. Also we can define different types of exception so that we can indicate them correctly. Also, we should create a package name exception to handle all the exception issue.

In our project, we handle all the exception within its own activities as most of the problems are caused by the network and hardware. We thought it's better not to merge these two different categories of exceptions together.

28. What is the advantage of exposing fix methods for exception management?

With an exception managment class, we can have more freedom to handle different exceptions with different manners.

In our project, we handle all the exception within its own activities as most of the problems are caused by the network and hardware. We thought it's better not to merge these two different categories of exceptions together.

29. Why did we have to make the Automobile object static in ProxyAutomobile class?

To share the `Automobile` object via the whole program.

In our project, we do not make anything static. Instead, we use sharedpreference to store the useful information.

30. What is the advantage of adding `marker` variable in `MainActivity` class?

It is used for further extension of the project. We need a way to track all the markers on the map.

31. What measures had to be pass `marker` to other Activity and Fragments?

We use `putArrayListExtra` to do the job. First use custom method to transform the marker class to a ArrayList and then rebuild it in corresponding activities.

32. When implementing ArrayList in MainActivity, how to consider the CRUD operations

As we fetch almost everything from the server, the client side do not need to modify any database. So the method from ArrayList is enough for us.

32. What is the advantage of LinkedHashMap over HashMap

They all support key-value pair insertion and search. However, `HashMap` only uses a hash function which means that it cannot keep the data in order. In the same time, `LinkedHashMap` will kepp the order of the objects so it is very efficient for time based iteration in the program.

In our project we use the sharepreference which is quite similar to LinkedHashMap

33. How to log operations and exceptions?

The most important thing for the log system if the timestamp. When something bad happens the log can help us find out the reason. It is critical for non-stop server program.

We use `log` quite a lot for the network request package to track what's going on.

34. What is enumeration used for?

Enumeration can be regarded as a way to represent integer with meaningful phrases. Each phrase has a unique number which helps the program to select in the switch clause.

In our project we create a `ConstantValue` class to store all the static value so as to keep them unified thourgh the whole develping process.

35. What is the best way to setup multithreading in an Enterprise Class application?

Multithreading is a method that sometime do more harm than good. So it is wise to limit its effective range to make sure that even if something bad happened for multithreading code, it won't break the whole program.

In our project, the network request is implemented through the `asyncTask` interface.

36. What Strategy is used for synchronizing, so you end up with a scalable application?

We can use mutex to make sure that even if there are lots of threads operating at the same time, only one thread can operate on one data in order to avoid race conditions.

In our server side we use atomic operation to make sure the CRUD operation on the database will not cause conflict.

37. What implementation strategy can be used for creating a race condition for testing Multithreading?

Race condition means that different threads want to access the same part of memory at the same time. For example, if two threads want to update the note of the same `User`, race condition will occur.

38. How does Synchronization work in JVM? What are the performance consequences of Synchronizing?

Lock the mothod for one object or lock the object for one thread. Synchronization will slow down the program while it can avoid race conditions.

This is also the strategy we used in the project.

39. What is producer-consumer problem? How to solve it?

It's a typical thread problem in multithreading. If the consumer continues to fetch item from the shelf even if it is empty, errors will occurs. So we need a way to tell the consumer whether it is ready for it to fetch new item.

We don't encounter this problem in the client development. As android handle it well.

40. Why should we not synchronize every method in class?

As the synchronizing process is time consuming so it will cause great performance problem if every method is synchronized even it is unneccessary.

We should only sync the ones needed to reduce the affect to the performance.

41. How to read shared preference

We create a util class to handle all the complex operation on shared preference. So in the other classes we can just use the util class like normal get/set method.

42. What is socket?

Socket is an abstraction of "Communication link" among different devices. For example, the client and server in this project uses socket for communication.

In our project we use `httppost` to handle the network operation.

43. How to transfer file via httppost?

Just like transfering item in real life, first we need to know the address, which is hostname(port). And then we can use socket object stream to transfer data from different sources.

44. What is protocol?

Protocol can be regarded as a pre-defined communication pattern for communication. So we can create our own protocal to set up communication. Actually what we do in this project is a kind of protocol.

In our project we have our own protocol document so all the connection can have a unified standard.

45. How could one server handle several clients?

We can create many threads on the same server, each thread can handle several requests from one client. In this manner can the server handle several clients.

Our server has lots of servlets and with the help of the multithread handling by Tomcat, we can support multiple users.

46. Are JSP and Java Servlet the same?

No, servlet is precompiled java application while JSP is just a kind of HTML webpage with some specific syntax.

Our server has lots of servlets and with the help of the multithread handling by Tomcat, we can support multiple users.

47. Why do we need to use Tomcat?

Because servlet, as its name implies, is the application running on the server. So we need Tomcat to be the container for our servlets.

Our server has lots of servlets and with the help of the multithread handling by Tomcat, we can support multiple users.

48. How to show different content via servlet?

Mostly, we can output what we need in the `doGet()/doPost()` method. From this process, we can learn from the different variables passed by the request and choose what to do accordingly.

Different url can be regarded as different interface.

49. How to connect to MySQL via Java?

We can use the JDBC package from the MySQL community. It provides basic methods for connecting to the database.

In our client we don't use any database.

50. Why do we need a primary key?

A primary key is a field in the table which is unique. It can be regarded as the identifier for different records. We need it to find the correct `Note`, `User`, `Comment` object record.

51. What are the three rules of Normalization?

1NF: arrays or other repeating fields should not be used.

2NF: in 1NF, and all the data in the table must be dependent on the primary key

3NF: in 2NF, each column except for keys, must not be interdependent.

We design our database based on the 2NF.