

Class Notes

Friday, April 24, 2015 3:21 PM

Understanding Portfolio:

- ✱ Used to assess your level of fluency
 - Technology Topics
 - UML Diagrams
 - Examples of Professionalism
- ✱ You Provide Direct Evidence to me

Hibernate:

1. The Singleton Pattern
2. The Proxy Pattern
3. The Factory Pattern
4. HQL
5. Many To Many definition
6. Many To Many example and diagrams
7. Full Hibernate annotations declaration and description
8. Many-To-One and One-To-Many Relationships
9. Bi-directional One-ToMany Relationships

UML

- Class
- Sequence
- State
- Use case

How to use git:

- git add

- `git commit -m "This is my first commit"`
- `git push`

hashmaps map keys to values

Weekly Journal

Saturday, April 25, 2015 4:55 PM

April 25, 2015

The week has been kind of weird. I broke my arm, and I have been behind in all my work this week due to sleeping a lot from the pain meds. I am not even sure exactly what I am supposed to be doing on this class, but I am sure I will get into the rhythm this week. I am not going to type much this week considering I have only one hand to type at the moment. I think I have everything done that I need to get done this week. I hope my semester in this class is a great one. I hope to end my final semester in college on the best note that I can.

May 2, 2015

This week I simply did what was required on iLearn. This is my first full week back after breaking my finger, and I can seriously see a LOT of the limitations that comes from typing with only 1 finger on one hand. Its really hard! Besides that, I have worked on really nothing else this week because of that. I just simple got to know a little bit more about my class.

May 9, 2015

This week we actually got to meet our team. I was excited to see that our team was all on campus. We met together and just spent some time quickly studding UML Diagraming so that we could present on it. I did activity diagrams, only to learn that I did not need to do that.

May 16, 2015

This week we meet again, but I did not really do anything new. I read up on some Java Collections. I didn't really do anything else that week.

May 23, 2015

This week I think is my breaking point! I haven't had time to do anything in this class, and time is going to get even more tight. I need to find time to at least get to a B in this class.

May 30, 2015

This week I feel like got a while lot done. I got most of the UML Diagramming finished as well as MVC and Hibernate. I was grateful to see that our group was willing to all work together to get at least a B in the class.

Team Meetings

Wednesday, May 6, 2015 10:11 PM

May 6, 2015

Using GitHub for the first.

GUI

- drag file
- open GUI
- commit
- sync

PWR

git pull
git push

What will I be learning this week?

- Java Collections - ~~Just~~ L the basics

Other Items

- What you did today?
yesterday?
tomorrow?

Elevator Pitch - Project Idea

Wednesday, April 29, 2015 12:12 PM

Wall Paper switcher

How will it switch?

- Based on location
- Based on time of day
- Based on battery Percentage

How many wall papers can we switch?

- 5 Wall papers

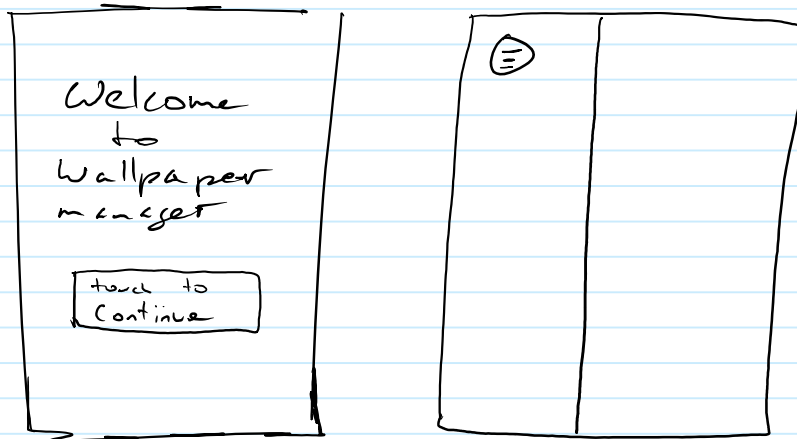
Why?

- For design
- Saving battery
- Can't decide what wall paper you like

Other Items:

- App will be process that runs in the background
- Process that runs in the background
 - Make sure that it runs smoothly in the background
 - If Battery option is selected, then process should shut itself off so it doesn't continue to run the background.

The Application



Use Case Document

Monday, May 25, 2015 9:45 PM

UML

Saturday, May 23, 2015 4:12 PM

Class Diagrams

- A class is basically a template for which objects are created.
- Classes define attributes, information that are relevant to their instances, operations, and functionality that the objects support. Some of the more important guidelines

Use Case Diagram

Useful Links:

- <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm>

General Notes

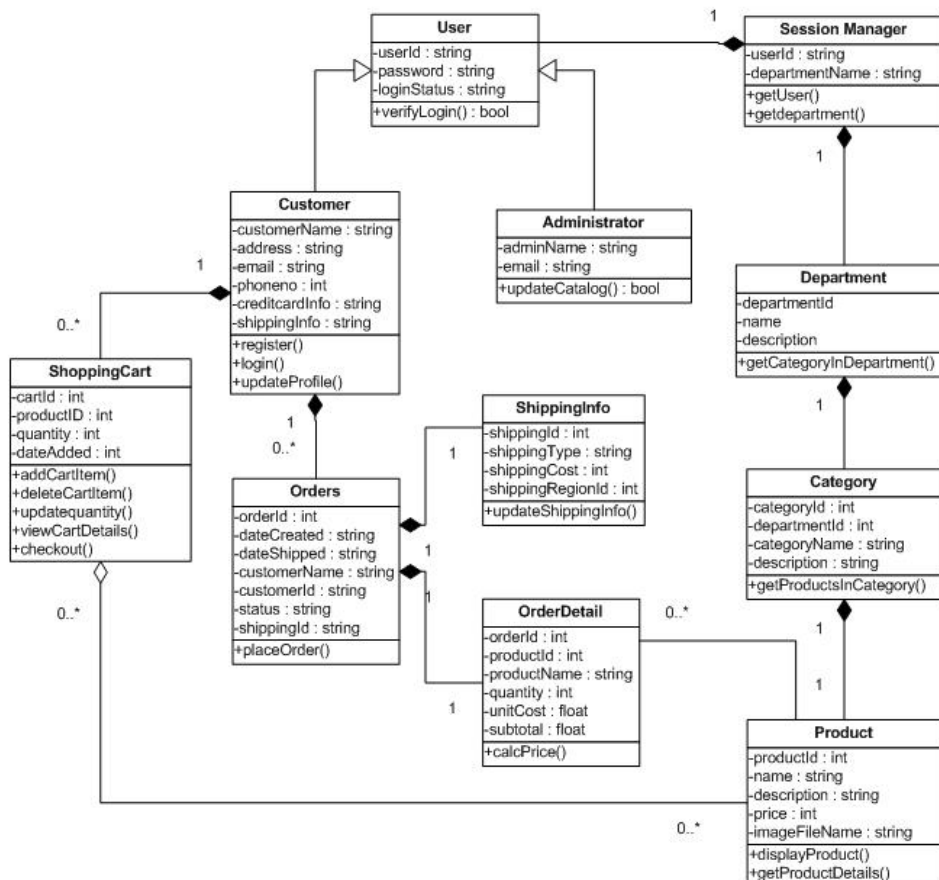
- Purpose of a use case diagram is to capture the dynamic aspect of a system.
- Use case diagrams are used to gather the requirements of a system including internal and external influences.

Class Diagram:

Useful Links:

- http://people.cis.ksu.edu/~reshma/798_ClassDiagram.htm

General Notes:



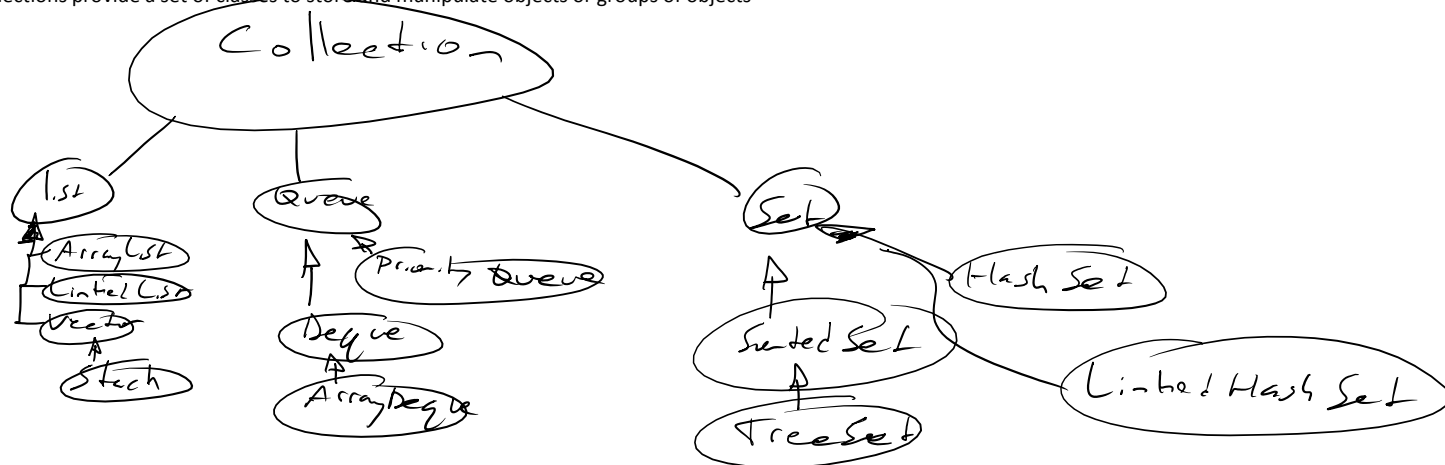
JAVA Collections

Wednesday, May 13, 2015 1:05 PM

Useful Links

- <https://docs.oracle.com/javase/tutorial/collections/intro/index.html>
- <https://docs.oracle.com/javase/tutorial/collections/interfaces/map.html>
- http://www.tutorialspoint.com/javaexamples/collection_sublist.htm
- <https://docs.oracle.com/javase/tutorial/collections/>
- <http://www.java2novice.com/java-collections-and-util/>
- http://www.tutorialspoint.com/javaexamples/java_collections.htm
- <http://math.hws.edu/javanotes/c10/s4.html>
- <https://www.youtube.com/watch?v=mkCTxtLe7XU>
- <http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>
- <http://docs.oracle.com/javase/7/docs/api/java/util/Collections.html>

Java Collections can sometimes be called a container. It is simply an object that groups multiple elements into a single unit. Collections provide a set of classes to store and manipulate objects or groups of objects



— What is a collection??

- Similar stuff

↳ collection of anything

ex/ employee

Hibernate

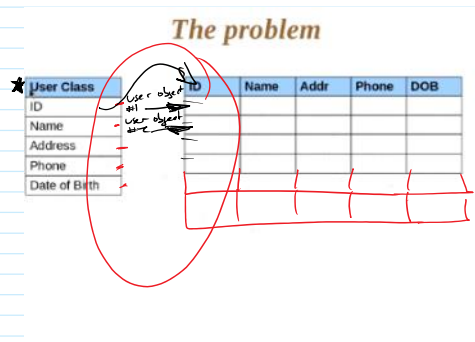
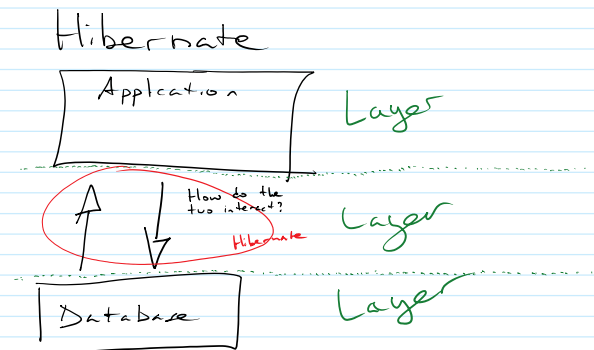
Tuesday, May 19, 2015 10:47 PM

Useful Links:

- http://en.wikipedia.org/wiki/Hibernate_%28Java%29
- Used in the data layer of applications
- http://www.theregister.co.uk/2005/12/06/hibernate_object_relational_mapping_2/
- <https://www.youtube.com/watch?v=66XbyTM7tM4>
- <https://www.youtube.com/watch?v=wNT-EZsaC98>
- <http://www.vnrgroups.com/articles/java-hibernate-tutorial-part-1/>
- <http://www.vnrgroups.com/articles/java-hibernate-tutorial-part-2/>
- <http://www.javaworld.com/article/2072999/data-storage/get-started-with-hibernate.html>

General Notes

- An object relational mapping framework for the Java language, providing a framework for mapping
- an object-oriented domain model to a traditional relational database
- It is an ORM tool
- Used in the data layer of applications, meaning the layer in which the applications interact with the data.
- Implements JPA - Java Persistence API - Persistence implement
 - More portable if not wanting to use hibernate
- The problem they are trying to solve
 - Saving data in a relational Database
 - Each of these objects data would have data for a particular user
- Need to use hibernate configuration in xml file

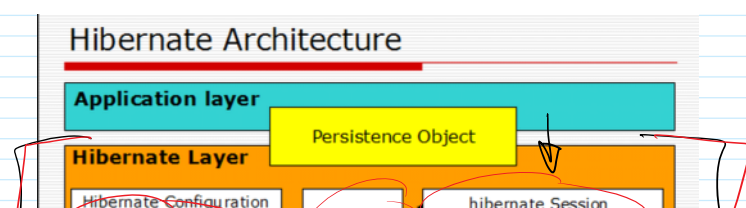
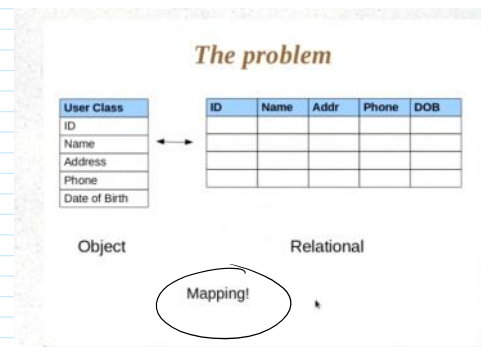


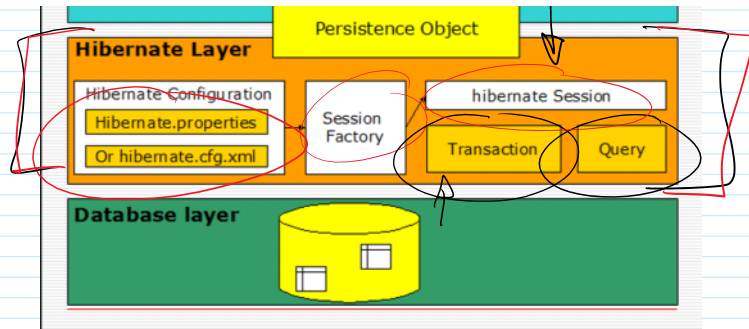
EX1 - User object #1 is saved as row #1

- A class corresponds to a table, an object corresponds to a row in that table

→ problem: objects there as java, no objects in database

* Mapping is what is a pain





JSON

Monday, May 25, 2015 12:28 PM

JavaScript Object Notation

MVC (Model View Controller)

Monday, May 25, 2015 4:17 PM

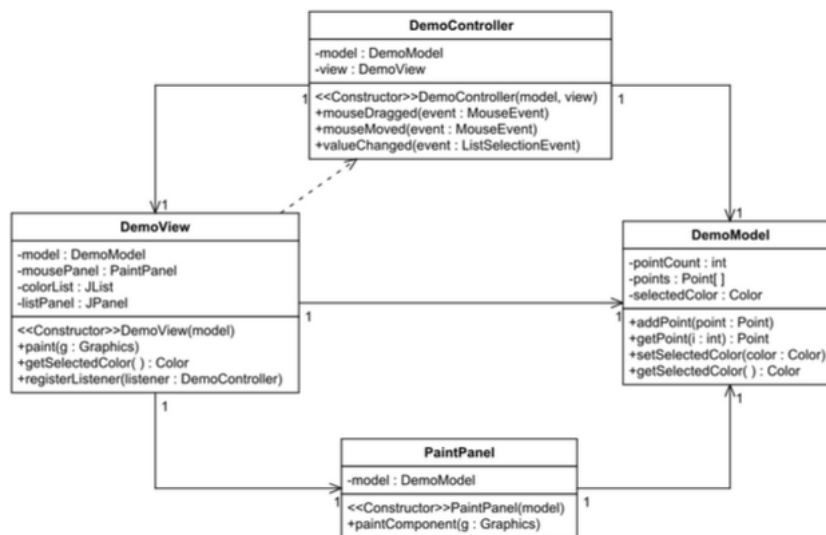
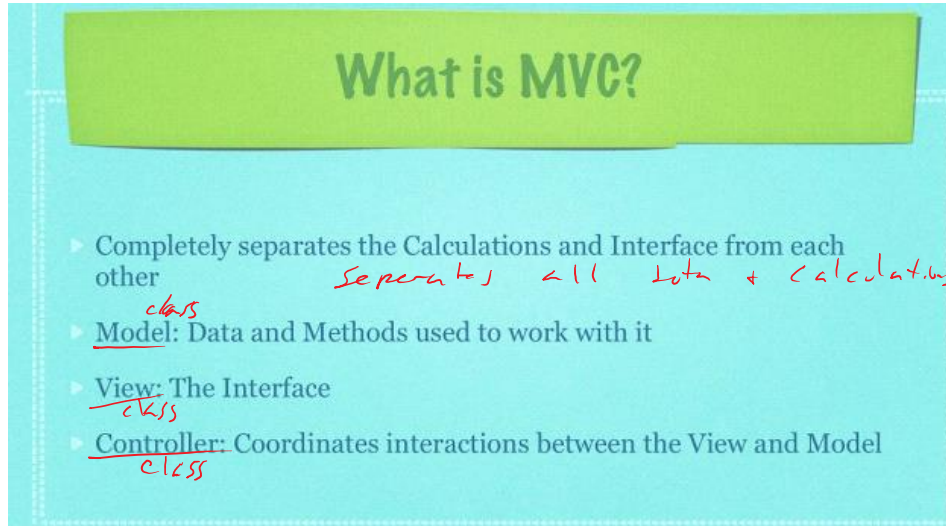
Useful Links:

<http://www.newthinktank.com/2013/02/mvc-java-tutorial/>

<http://www.cs.utsa.edu/~cs3443/mvc-example.html>

<http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design>

General Notes



- Can think of this a data interface coordinator
- Software development method that keeps it when not in use, helps all developers in the world figure out what is wrong with apokication

JUnit

Wednesday, May 27, 2015 2:35 PM

Useful Links:

- <https://netbeans.org/kb/docs/java/junit-intro.html> - a tutorial that goes through all of this
- <http://junit.org/>
- <http://hamcrest.org/JavaHamcrest/javadoc/1.3/org/hamcrest/Matchers.html>
- <http://www.vogella.com/tutorials/JUnit/article.html>
- <http://junit.sourceforge.net/doc/testinfected/testing.htm>

General Notes

- To test if the return value equals a specific value use `assertEquals()`
- Start by making our tests with `@test`
- JUnit defines how to structure your test case and provides the tools to run them.

JSON

Friday, June 5, 2015 8:11 PM

Useful Links:

- <http://json.org/>
- <https://www.youtube.com/watch?v=GF8yRG04-60>
- <https://www.youtube.com/watch?v=LDzA17uruAo>
- Jsonlint.com

General Notes:

- JSON (JavaScript Object Notation) - a lightweight data interchange format.
- Is a syntax for storing and exchanging data
- Easier to use alternative to XML
- Lightweight data change format.
- Easy for humans to read and write. Easy for machines to parse and generate.
- Based on a subset of JavaScript
- Storing JSON data creates an object, that we can access using a variable.

Results of Example Code:

```
run:
Enter Student name: Paul
Enter course name: Java
Enter grade: 89
{"courses":[{"grade":89,"name":"Java"}],"name":"Paul"}
BUILD SUCCESSFUL (total time: 14 seconds)
```

Client and Server Sockets and their I/O streams

Wednesday, May 27, 2015 7:00 PM

Useful links:

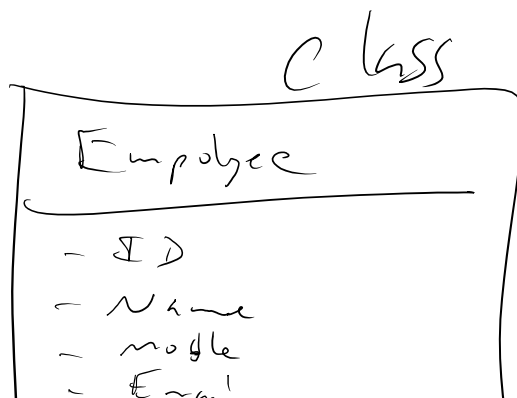
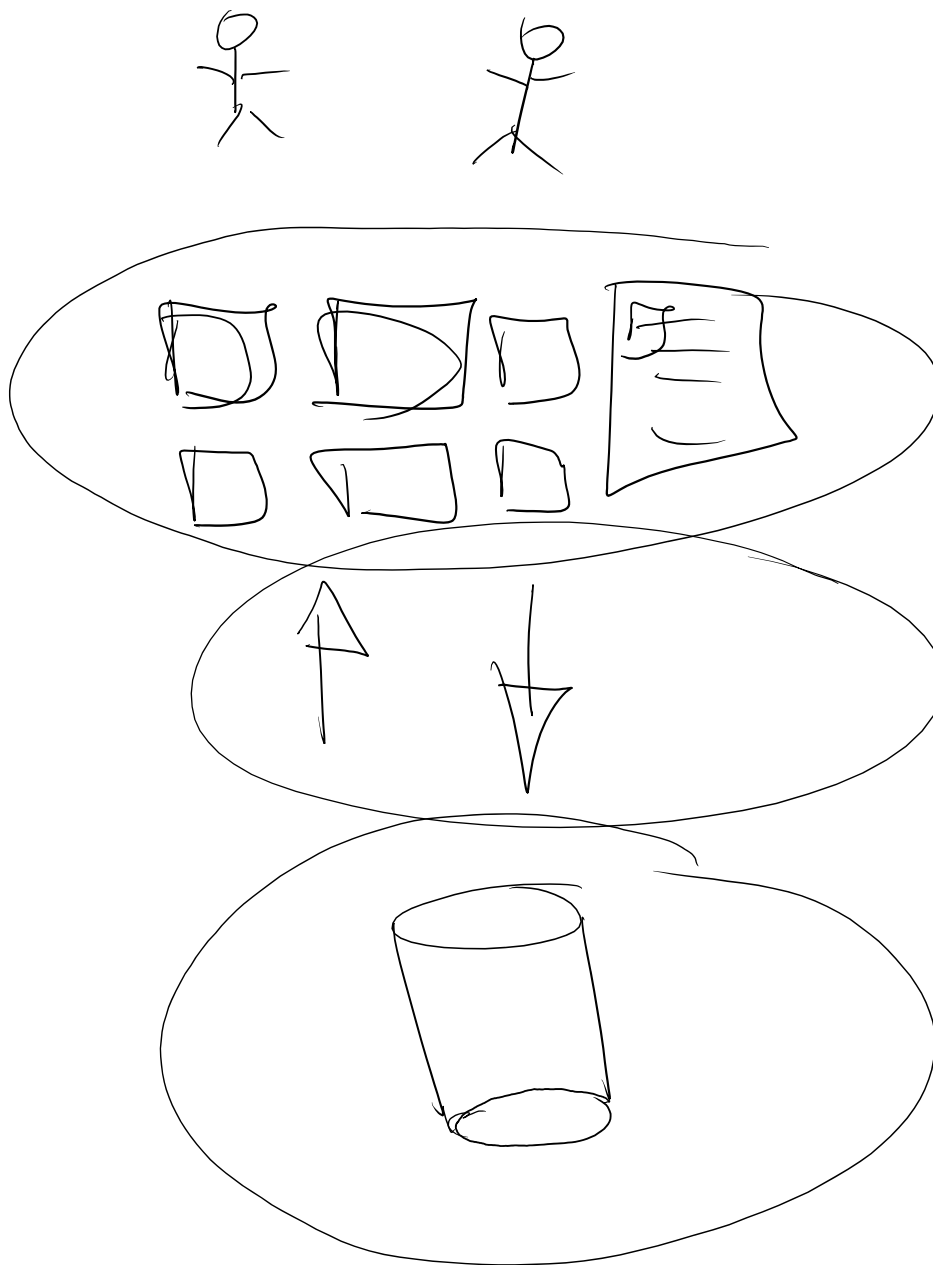
- <https://www.youtube.com/watch?v=pr02nyPqLBU>

General Notes

- What is a stream? Whenever you connect to someone else's computer through networking using Java. The way we communicate is through streams; two main streams: the output stream and the input stream. The output stream is what flows away from the computer and to your friends computer
 - Output goes away from you input goes to you
 - Whenever we type text and press enter, that text is packed up and sent out and put into someone's computer.
 - Whenever some friend puts text and presses enter to send you message, that text is put into a 'package' and sent to my computer in a text stream.

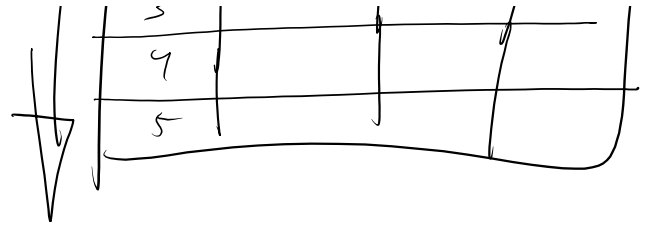
Presentation Notes

Monday, May 25, 2015 5:12 PM



ID	Name	mobile	Email
1			
2			
3			
4			

- name
- mogle
- Email



Threads, Executors, Runnables, Android Handlers

Tuesday, June 16, 2015 10:43 PM

Useful Links:

- <http://stackoverflow.com/questions/21194762/what-is-the-different-between-handler-runnable-and-threads>
- <https://developer.android.com/training/multiple-threads/run-code.html>
- <http://developer.android.com/reference/java/util/concurrent/ThreadPoolExecutor.html>
- https://www.youtube.com/watch?v=LJ_pUIWzGsc - handler
- <http://android-developers.blogspot.de/2009/05/painless-threading.html>
- <http://examples.javacodegeeks.com/android/core/os/handler/android-handler-example/>

Notes

- Runnable separates code that needs to run asynchronously, from how the code is run. Your code is kept flexible this way. Asynchronous code for example in a runnable can run on a threadpool, or a dedicated thread.
- Threads
 - They occupy a lot of memory
 - A threadpool executor takes a task from the queue and runs it on the thread.
- A handler
 - Allows to post runnables to execute on a specific thread. Behind the scenes, runOnUiThread Thread queues the runnable up with Android's UI handler so that the runnable can execute safely on the UI thread.
- Executor
 - Is an object that executes submitted runnable tasks. The interface provides a way of decoupling task submission from the mechanics of how each task will be run, including details of thread use.
-

HTTP URL Connection

Thursday, June 18, 2015 8:56 PM

Useful Links

- <http://developer.android.com/reference/java/net/URLConnection.html> -
- <https://www.youtube.com/watch?v=h3ErnJKzZU>
- <https://www.youtube.com/watch?v=JcZwlpV1rMc> - watch this one
- <https://www.youtube.com/watch?v=ryY7Dy3z-7Q>
- <https://docs.oracle.com/javase/tutorial/networking/urls/readingWriting.html>

URLConnection - an http url connection used to send and receive data over the web.

- Data may be of any type and length
- This class may be used to send and receive streaming data whose length is not known in advance

AsyncTask - used in order to do long running tasks

In the background, important job;

In application going to download application

//main purpose is to demonstrate an async task and how to connect to the internet

Application Controller Pattern

Tuesday, June 23, 2015 12:25 PM

Useful Links:

- Difference Between application controller and model view controller - <http://stackoverflow.com/questions/120781/what-should-an-application-controller-do>
- <http://www.developerfusion.com/article/9450/controller-patterns-for-aspnet/>
- <http://martinfowler.com/eaCatalog/applicationController.html>

General Notes:

- Application controller is a centralized point for handling screen Navigation and the flow of an application
- It controls the UI transition logic in a central place
- It centralizes the flow of application and screen navigation, so you can change the flow or the logic in a single location, with minimal changes or no changes to the source code of views
- For a complex process we can define multiple controllers

