mynt
finance for all

IN PARTNERSHIP WITH

apper.ph

GCash

FINTECH

CADETSHIP PROGRAM

Java™ Mentorship Program

with the Apper Cloud Enablement Team

apper.ph

Intro to Java - Part 2

Intro to **Java**

Properties

Securing Data

Database

Asynchronous vs Synchronous

Properties

Properties

- **Store config files**
- **Loadconfig file**

Securing Data

Securing Data

- Personal Identifiable Data
- Verify Data Integrity - Message Digest
- Cipher - Encrypt / Decrypt
- 3rd Party Library : Apache Commons

Message Digest

- Supports SHA-1, SHA 256, MD5
- Compare Hash

Cipher

- **Encrypt Data**
- **Decrypt Data**

Database

Importing Library

- **Adding manually**
- **maven**
- **gradle**

Synchronous and Asynchronous

Synchronous

- **Blocking**
- **Waiting**
- **Used when there is dependency**

Asynchronous

- Non-Blocking
- Execute multiple process
- Use Threads and Runnable
- Use Future
- Used when multiple task have no dependence

Any Questions?

Java Day 2 - Labs

Java Day 2

Lab Notes :

Error Checking must be performed

Determine the data types

Determine function names

Determine the data for the classes

Must use an interface or abstract

GOAL

- Store and Retrieve data from the database
- Design and create Classes appropriate for the scenarios

Lab 03.0.1 A

Classes - SMS

Create a class sms with the following fields.

MSISDN
RECIPIENT
SENDER
SHORT CODE
TRANSACTION ID (contains number and characters)
TIMESTAMP

Note: TRANSACTION ID is auto generate by another system and is not generated.

Intro to Java Labs

Lab 03.0.1 B

Classes - Promo

Create a class Promo with the following fields.

Promo Code
Details
Short Code
Start Date
End Date

Intro to Java Labs

Lab 03.0.2

DATABASE

Create a SMS Database
Create a table for SMS
Create a table for Promos

Lab 03.1.1 A

SMS CHECKER

Create a function that will accept a map with exactly 3 items.
The first item is the mobile number.
The second item is the sms
The third is the short code that will send the sms

Example:
Mobile number : +639563026795
Message : PROMO CODE ACCEPTED
Short Code : 1234

Lab 03.1.1 B

SMS CHECKER

This will check the sms sent and tag the sms for which promo the sms is for.

It will tag if the sms is SUCCESS SMS or FAILED SMS based on the promo rules.

Identify the test scenarios and create the corresponding JUnit test scripts

Lab 03.1.2

Classes

Create an interface for managing SMS then implement these in a class

These are the required functions :

-Insert SMS
-Retrieve SMS given a start date and end date
-Retrieve SMS given a promo code
-Retrieve SMS given an msisdn
-Retrieve SMS sent by the system
-Retrieve SMS receive by the system
-Retrieve SMS given several msisdn

Lab 03.1.3 A

Scenario

Insert a Promo in the Database called “Piso Pizza”
It will run from Feb 01 2021 10:00 am - June 30, 2021 23:59

User needs to send an SMS to the short code **1234555**

Lab 03.1.3 B

Scenario

Users send “PROMO” to the shortcode to get the Pizza that will cost 1 peso

Users send “REGISTER” to the shortcode to be able to avail of the promo. The system will reply

To complete the promo registration, please send Lastname, Firstname to 1234555

Example : Valmores, Marco

Lab 03.1.4 C

Data Population

Insert 30 SMS for the “PISO PIZZA” promo, make it a mix of sms the user can send.

Insert 2 other promos.

Insert 30 SMS for other promos.

Lab 03.1.5

Test Cases

Identify the test scenarios and create the corresponding JUnit test scripts for checking the sms for the promo mentioned.

Lab 03.1.6

Database

Generate the following report for the “PISO PIZZA” promo :

List of Failed Transactions
List of Failed Transactions per SMS Type
List of Successful Transactions
List of Successful Transactions per SMS Type
List of Persons who joined the Promo
Total Count of SMS received
Total Count of SMS sent

NOTE : SMS Type - Sent and Received

Lab 03.1.7

Codecommit

Upload the code to Codecommit

Use the following for repository name - “java_<LastName>”
Ex. java_valmores

Use the following description - “SMS Exercise - Marco Valmores”

apper.ph