**Southern Luzon Technological College Foundation, Inc.**

**NC III – Java Programming**

**Project Documentation**

**GAME INVENTORY IN MYSQL DATABASE AND JSON FILE**

**Chris Lemuel D.G. Dela Cruz**

**Student**

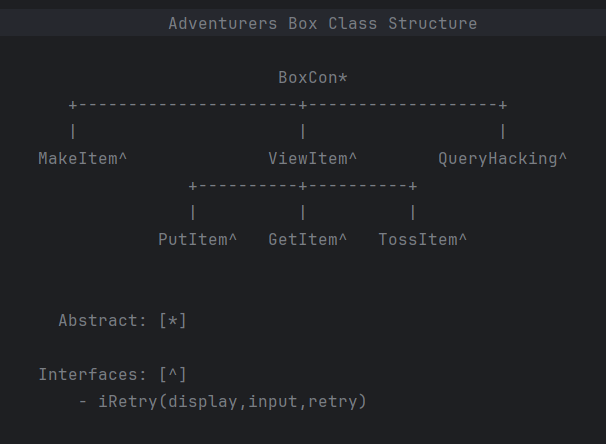
**Benz Vincent A. Geraldizo**

**Instructor**

**SUMMARY**

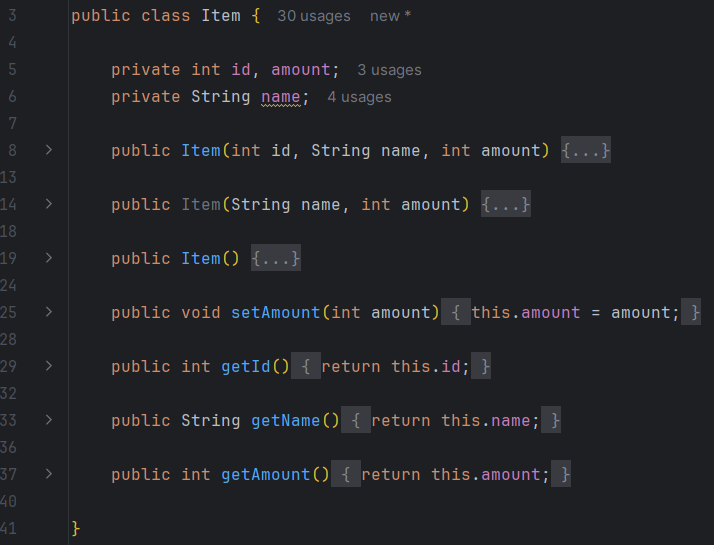
**** This project is a Command-Line Interfaces Game Inventory written in Java programming with basic CRUD functionality in MYSQL database as the game storage box and a Json file acting as the (adventurers) bag. The reason for doing this project is to show what the student learned in the past months of online training as well as incorporating some of what the student learned outside of the training.

As mentioned earlier, this project is written in Java programming language with the help of IntelliJ Integrated Development Environment, Java Development Kit version 24, and with the external Java Archive (JAR) files namely Java Database Connector (JDBC) for connecting in MySQL Database and Jackson (core, annotation, data-bind) for reading and writing Json file. And lastly a MySQL server as for the project database.

**PROJECT STRUCTURE**

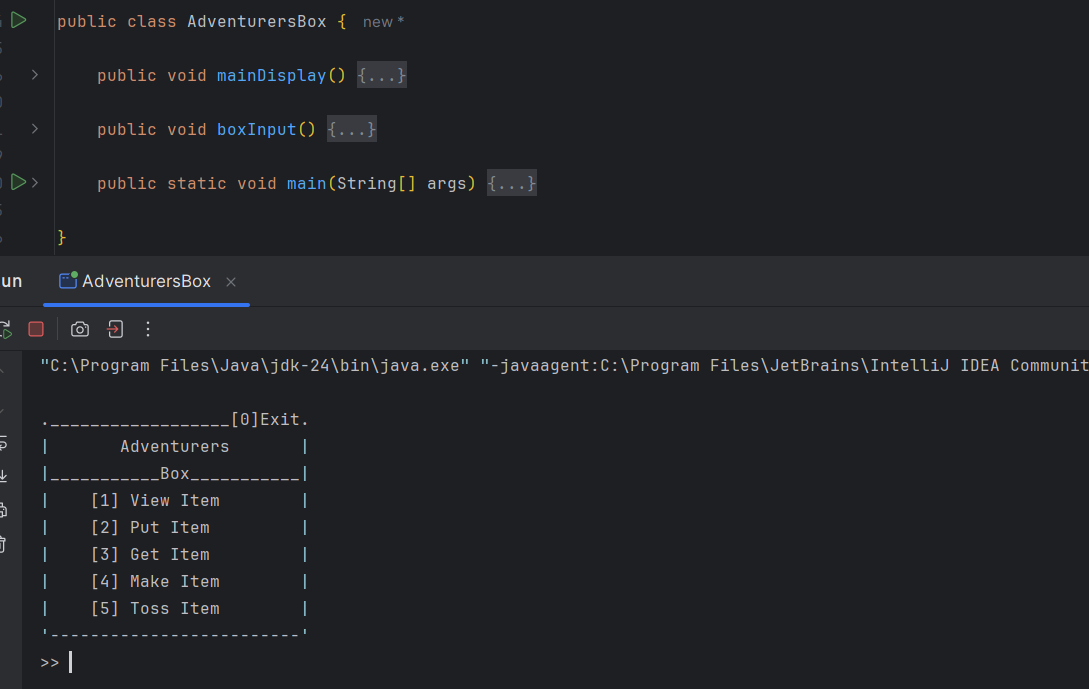
**Image 1: Image representation of project structure**

The following will be a walkthrough in all the classes and alike that is used for the project named Adventurers Box as a representation for a game inventory. This also includes some classes that are not visible in the image above (Image 1) and will be discussed next page.

****

**Item Class**

This class contains a representation of an item consisting of three fields for id, name and amount it holds. The class has various constructors for creating items and functions for retrieving or updating the fields. The student purposely put this class in private to demonstrate encapsulation and polymorphism through constructors.



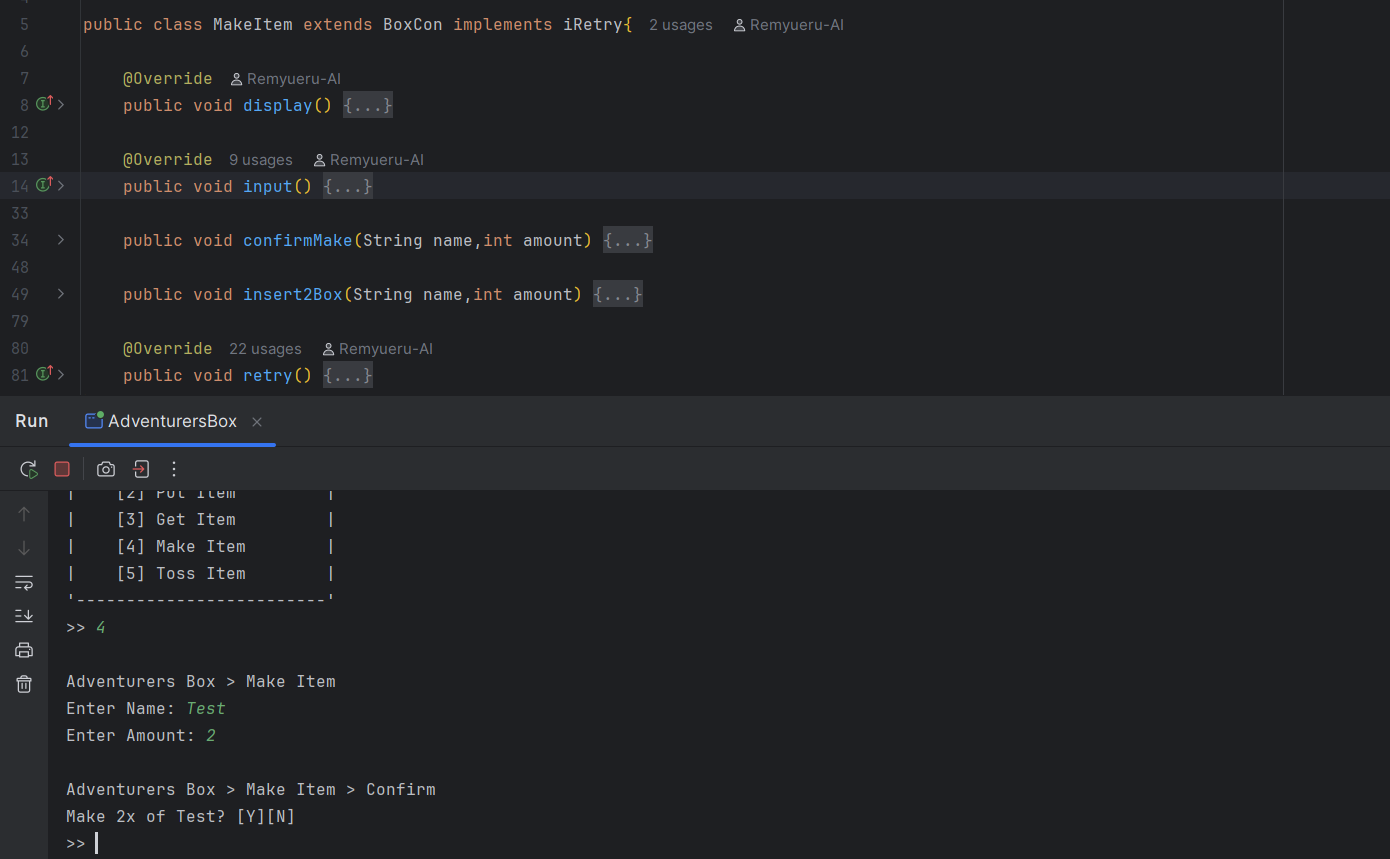
**AdventurersBox Class**

This class is mainly for executing the program, the main display and input of it. A switch case is made for accessing the other classes and entering zero will exit the program.

**BoxCon Class**

This class is created for the sole purpose of connecting the program into database and being the base class of the program, as such the student decided that making it an abstract class since the other classes would be only inheriting its functions and fields.

**iRetry Interface**

This interface holds three methods namely display, input and retry; it is needed to implement for the remaining classes in this project.

**MakeItem Class**

This class is used for two things, adding a new item in the database or adding an existing item amount if item name matched. It is done in the insert2Box function that will search and alter the amount for the item if it has already existed, otherwise it will become a new item in the database, with that it solves the problem of duplicating an existing item.

public void insert2Box(String name,int amount) {  
 System.*out*.println("Processing....");  
 qry = "select \* from boxTb where itemName like '" + name + "'";  
  
 try {  
 connectBox();  
 statement = con.createStatement();  
 resultSet = statement.executeQuery(qry);  
  
 if (!resultSet.isBeforeFirst()) {  
 qry = "insert into boxTb values (?,?,?)";  
 connectBox();  
 preStatement = con.prepareStatement(qry);  
 preStatement.setInt(1, 0);//id temp  
 preStatement.setString(2, name);  
 preStatement.setInt(3, amount);  
 }  
 else {  
 qry = "UPDATE boxTb SET amount = amount + " + amount + " WHERE itemName = '" + name +"'";  
 connectBox();  
 preStatement = con.prepareStatement(qry);  
 }  
 preStatement.executeUpdate();  
 System.*out*.println("The item is in the box\n");  
 retry();  
 }  
 catch (Exception e) {  
 System.*out*.println("Something went wrong" + e);  
 }  
}

**ViewItem Class**

This