Part 5 High Level Design

Brian Lemke   Matthew McFadden   Thomas Macaulay

Summary

Our program is divided into three main portions: the user interface controller, the database access controller, and the data objects. The interface controller is launched from main() and presents various menus in a continuous loop. These menus call a function from the database access class using plain data objects, and then the database access controller constructs and issues the SQL statement. Any results are returned to the interface controller in the form of our plain data objects.

Data Objects

Our program uses one Java class for each entity set in our E/R diagram. That means that each relation has an associated class with identical attributes except for our Item class, which has three subclasses that implement the specific attributes of each of our item types. These classes are very basic Java classes with no special functionality -- they simply serve as convenient wrappers to pass data around in a higher-level form than ResultSets.

Database Access class

DatabaseAccess.java defines a class dedicated to communicating with the database. This class defines a method to perform some high-level task (e.g. “Insert a Customer” or “Find all Libraries”) and translates that action and the high-level object parameters into a SQL statement. Query results from the database are translates from ResultSets into high-level objects or ArrayLists of objects.

When the database is created, PreparedStatements are created for every statement we have programmed into the application. Then, when the user interface requests an operation, the DatabaseAccess method fills in the appropriate PreparedStatement parameters and executes the SQL statement. The methods catch any exceptions or errors and translate them into returning boolean or null values on error.

User Interface class

UserInfertace.java defines the console-based interface with the user. When the interface starts, it asks the user to select a library from a list of all libraries in the system. From then on, the application remembers the library choice and uses that library for all queries and interactions relying on a library.

The main menu runs on a continuous loop. It presents a number of different choices, and asks the user to enter the number of a choice. If the users chooses the “Exit” choice, then the loop breaks and the program terminates. Otherwise, it invokes a sub-menu function, which all follow a standard format. The submenu function takes control of the user interaction, displaying output or asking for input. When the function is completed, the submenu returns control to the main menu.