CS395 Lab1

Kyle Aure

Version 1.0, 2018-09-03

Project Description

Introductory lab exercise for CS395 Applied Database Management. Introduces working with external data using the Java programing language.

Course Details

- **Course** - CS385: Applied Database Management
- **Instructor** - Dr. Nicole Anderson

Project Goals

- Download student list from D2L (`Student.xlsx`).
- Choose format to work with data, such as, comma delaminated format.
- With a partner, using pair programming, or individually.
- Construct a Java program that reads from the file, then prints out the names of all of the students that have a favorite color of Green or Black in alphabetical order.
- For next week, be prepared to answer questions about how you completed this task, and about the challenges you would face if additional requirements were added to the program.

Running project

Create a local copy of this project by running the following command:

```
git clone git@github.com:KyleAure/WSURochester.git
```

Then navigate to this project directory:

```
cd WSURochester/CS385/lab1
```

Then run the following maven goals to build and run this program:

```
mvn clean install
mvn exec:java -Dexec.mainClass="com.cs385.app.App"
```

Automated testing

Four automated tests were created and run when building with maven.

- `testStudentAddedToList`: Ensures that after creating a new student that this student is then added to the student list.
- `testRemoveStudentFromList`: Ensures that after a student is added to the list it can successfully be removed using the removeStudentFromList method.
- `testSort`: Ensures that students are successfully sorted by last name.
- `testQuery`: Ensures that only students that have a 'favorite color' of black or green end up in the sorted list.

Documentation

Workflow

> How were these tasks completed?

I started off by creating a Plain Old Java Object (POJO) to represent the data contained in the Student List. From there I knew I would need to store a list of the student object and decided it best to keep everything student related in the student class.

When it came to creating the application I researched different CSV document parsers and found one I wanted to use from apache since it seemed easy to implement. It had many dependencies so I decided that my project should be built using maven to keep the program portable between different machines.

I decided to use a `JPane` and `JTable` user interface to display the data to the user. I had used these in a previous course so it was easy to implement and get running.

Since I was using maven I also decided to do some automated testing to ensure that data was successfully being created, sorted, and queried.

Challenges

> Challenges I would face if additional requirements were added to the program?

If additional requirements, such as, acceptable data sources (`.txt`, `.xlsx`, etc) were added. This program would become bloated very quickly with parses as there would not be a standard among the data sources.

Another difficult addition would be other conditions for querying the data. The data was collected in a non-restrictive way. There is no consistency on how the data is structured for the pets, movies, and shoe size columns. It would be very difficult to accurately interpret this data and query it.

Learning Outcomes

> What did I learn?

- I learned how to implement maven builds outside of Java EE.
- I learned the difficult nature of using unstructured data.