

ĐẠI HỌC QUỐC GIA – ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN
KIỂM TRA PHẦN MỀM

Nhóm L02 – 03

Giảng viên hướng dẫn: PGS.TS Bùi Hoài Thắng

Sinh viên thực hiện	Mã số sinh viên	Điểm số
La Quốc Anh	1710465	
Hà Hải Thiên Sơn	1811193	
Nguyễn Tiến Đăng Khoa	1832026	
Trịnh Duy Hưng	1913652	

Tháng 1/2022



Mục lục

<u>I.</u>	Lời cảm ơn	3
<u>II.</u>	Giới thiệu	4
<u>III.</u>	Phân công nhiệm vụ.....	4
<u>IV.</u>	Báo cáo	5
	1. BTree.java	5
	2. UserDAO.java	7
	3. Controller.java	8
	4. Driver.java.....	9
	5. Model.java	10
<u>V.</u>	Tổng kết	11

I. Lời cảm ơn

Thân chào thầy và các bạn sinh viên, đây là bài tập lớn môn Kiểm tra Phần mềm với sự hướng dẫn của thầy Phó giáo sư tiến sĩ Bùi Hoài Thắng.

Chúng em sẽ cố gắng cung cấp các ý về các bài toán của đề tài một cách dễ hiểu nhất. Thay mặt lớp, nhóm xin chân thành cảm ơn thầy đã chỉ dạy và hướng dẫn nhiệt tình trong học kì vừa qua.

II. Giới thiệu

Trong bài tập lớn này, nhóm sử dụng bảng danh sách kiểm tra code được dựa trên bản gốc của công ty TMA Solution. Bảng này được chia ra làm 5 trang phần bao gồm

1. Btree.java
2. UserDAO.java
3. GPA-Calculator.zip
 - 3.1 Controller.java
 - 3.2 Driver.java
 - 3.3 Model.java

Nhóm có nhiệm vụ kiểm tra tất cả các file và tìm ra các lỗi trong chúng.

III. Phân công nhiệm vụ

No.	Family name	First name	Student Code	Role	Contribution	Job
1	La Quốc	Anh	1710465	Leader		GPA-Calculator/Driver.java GPA-Calculator/Model.java
2	Hà Hải Thiên	Son	1811193			UserDAO.java
3	Trịnh Duy	Hung	1913652			BTree.java
4	Nguyễn Tiến Đăng	Khoa	1832026			GPA-Calculator/Controller.java

Link buổi meet họp nhóm làm bài tập:

<https://drive.google.com/file/d/1YBfAE9HjvOXARRZeQHSj5du6J90Aid8/view>

IV. Báo cáo

Nhóm xin trình bày những lỗi được tìm thấy, những check case đúng xin phép được bỏ qua, chi tiết nằm trong file Assignment 1 Code Review Report.xlsx được đính kèm.

- **BTree.java**

```
Project 1 > BTree.java
1  package simpledb;
2
3  import java.io.*;
4  import java.util.*;
5
6  import simpledb.Predicate.Op;
7
8  public class BTreeFile implements DbFile {
9
10     private final File f;
11     private final TupleDesc td;
12     private final int tableid ;
13     private int keyField;
14
15     public BTreeFile(File f, int key, TupleDesc td) {
16         this.f = f;
17         this.tableid = f.getAbsolutePath().hashCode();
18         this.keyField = key;
19         this.td = td;
20     }
21
```

4. Does every method return the correct value at every method return point?

Hàm Page (dòng 34) có kiểu trả về là Page nhưng return trả về khác kiểu, có thể là class con được thừa kế từ class Page.

6. Are descriptive variable and constant names used in accord with naming conventions?

Tên biến f, td, rf, f, t, pgcateg, ipred, it, curp, nextp quá ngắn gây khó hiểu.
Tên biến tableid, retval, dirtypages, tuplenum không đúng camelCase.

15. Is every method parameter value checked before being used?

Constructor chưa có những kiểm tra tham số đầu vào có hợp lệ hay không, hơi sơ sài dễ gây ra nhiều bug tiềm ẩn.

16. Does each class have an appropriate constructor?

Constructor chưa có những kiểm tra tham số đầu vào có hợp lệ hay không, hơi sơ sài dễ gây ra nhiều bug tiềm ẩn

23. Are divisors tested for zero or noise?

BufferPool.getPageSize() chưa được kiểm tra != 0

25. Are the comparison operators correct?

Sai phép so sánh == thành phép gán =.

46. Does every attribute, variable or constant declaration have a comment?

Các thuộc tính của class thiếu comment mô tả.

47. Is the underlying behavior of each method and class expressed in plain language?

Code phức tạp nhưng lại quá ít comment giải thích.

50. Do the comments help in understanding the code?

Comment tiếng Trung Quốc, khó hiểu.

53. Is a standard indentation and layout format used consistently?

Code lùi dòng, xuống dòng bất thường, không có cách khoảng trống hợp lý.

Hàm stealFromLeftInternalPage, stealFromRightInternalPage có phần khai báo tham số được lùi dòng bất thường, khác với những hàm khác.

54. For each method: Is it no more than about 60 lines long?

Hàm readPage (34 - 94), hàm splitLeafPage (188 - 232), setEmptyPage (1063 - 1134) có nhiều hơn 60 dòng.

56. Can the cost of recomputing a value be reduced by computing it once and storing the results?

BufferPool.getPageSize() và BTreeRootPtrPage.getPageSize() được sử dụng nhiều lần.

- **UserDAO.java**

```
Project 1 > UserDAO.java
1  package com.inventory.dao;
2
3  import com.inventory.database.ConnectionFactory;
4  import com.inventory.dto.UserDTO;
5  import com.inventory.ui.Users;
6  import java.io.File;
7  import java.io.FileInputStream;
8  import java.sql.Connection;
9  import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.ResultSetMetaData;
12 import java.sql.SQLException;
13 import java.sql.Statement;
14 import java.util.Vector;
15 import javax.swing.ImageIcon;
16 import javax.swing.JOptionPane;
17 import javax.swing.table.DefaultTableModel;
18
19 public class UserDAO {
20     ResultSet rs = null;
21     PreparedStatement pstmt = null;
22     Statement stmt = null;
23     Connection CON = null;
24 }
```

6. Are descriptive variable and constant names used in accord with naming conventions?

DeleteUserDAO (không đúng quy tắc)
rs (tên biến quá ngắn gây khó hiểu)
COL không rõ ràng
userdto không tuân thủ camelCase.

9. Are all for-loop control variables declared in the loop header?

Hai biến đếm của hai vòng lặp tại dòng 187 và 195 không được khai báo trong vòng lặp.

37. Are all exceptions handled appropriately?

Không có hàm xử lý ngoại lệ

46. Does every attribute, variable or constant declaration have a comment?

Không có comment.

51. Are there enough comments in the code?

Không có comment.

53. Is a standard indentation and layout format used consistently?

Lùi dòng dư một tab.

62. Are there any redundant or unused variables or attributes?

CON.

63. Could any non-local variables be made local?

rs.

- **Controller.java**

```

Controller.java X
Project 1 > GPA-Calculator > Controller.java
1  package application;
2
3  import java.net.URL;
4  import java.util.ResourceBundle;
5  import java.util.Set;
6  import javafx.event.ActionEvent;
7  import javafx.event.EventHandler;
8  import javafx.fxml.FXML;
9  import javafx.fxml.Initializable;
10 import javafx.scene.Node;
11 import javafx.scene.chart.AreaChart;
12 import javafx.scene.chart.PieChart;
13 import javafx.scene.control.Button;
14 import javafx.scene.control.ComboBox;
15 import javafx.scene.control.TextField;
16 import javafx.scene.control.Alert;
17 import javafx.scene.control.Alert.AlertType;
18 import javafx.scene.input.KeyEvent;
19 import javafx.scene.layout.GridPane;
20 import javafx.scene.text.Font;
21 import javafx.scene.text.Text;
22 import java.util.ArrayList;
23 import java.util.HashSet;
24
25 public class Controller implements Initializable
26 {
27     @FXML
28     private GridPane inputGrid;
29     @FXML

```

9. Are all for-loop control variables declared in the loop header?

Không khai báo biến đếm trong loop header.

25. Are the comparison operators correct?

Phép so sánh sử dụng dấu "=".

28. Has an "&" inadvertently been interchanged with a "&&" or a "|" for a "||"?

Phép or dùng dấu |.

29. Is every three-way branch (less, equal, greater) covered?

Dùng else.

54. For each method: Is it no more than about 60 lines long?

Hàm newSemester nhiều hơn 60 dòng.

56. Can the cost of recomputing a value be reduced by computing it once and storing the results?

model.getNumOfRows() được dùng nhiều lần trong hàm addRow.

60. Are there variables with confusingly similar names?

Biến text22 dễ nhầm với text2 ở dòng 50.

Biến text11 dễ nhầm với text1 ở dòng 40.

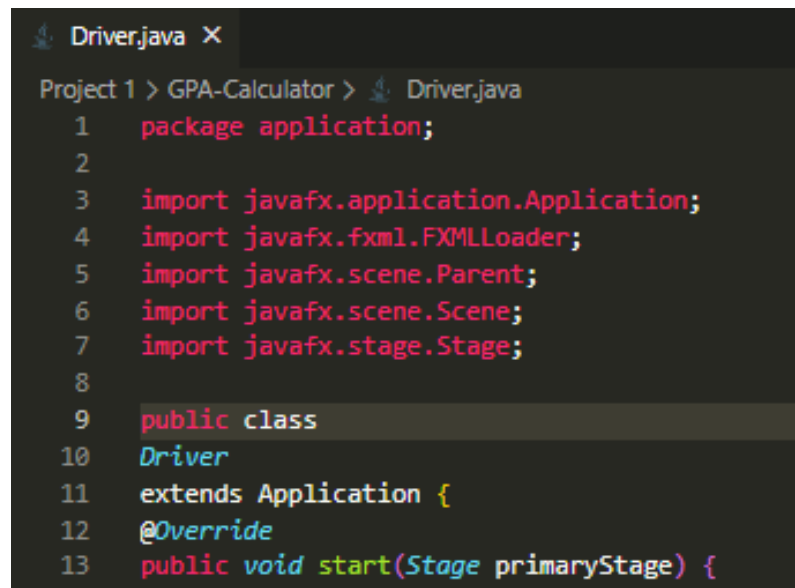
62. Are there any redundant or unused variables or attributes?

Nhiều biến không được dùng đến.

66. Are there any blocks of repeated code that could be condensed into a single method?

Hàm saveButton.setDisable và newButton.setDisable được sử dụng nhiều lần với nhau.

- **Driver.java**



```

Driver.java X
Project 1 > GPA-Calculator > Driver.java
1  package application;
2
3  import javafx.application.Application;
4  import javafx.fxml.FXMLLoader;
5  import javafx.scene.Parent;
6  import javafx.scene.Scene;
7  import javafx.stage.Stage;
8
9  public class
10 Driver
11 extends Application {
12 @Override
13 public void start(Stage primaryStage) {

```

6. Are descriptive variable and constant names used in accord with naming conventions?

Parent Root.

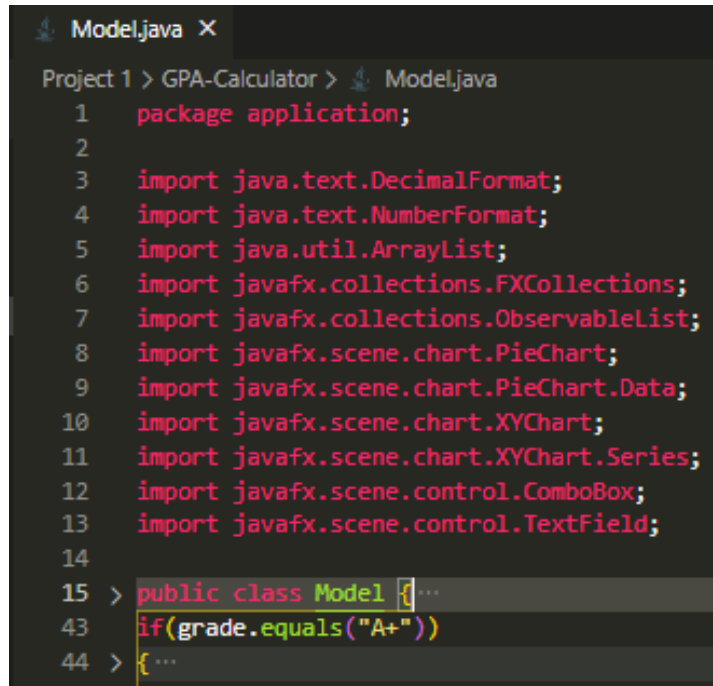
25. Are the comparison operators correct?

Phép so sánh = là sai.

53. Is a standard indentation and layout format used consistently?

Code không được lùi dòng hợp lệ, code nằm trong khối con phải lùi dòng vào so với bên ngoài

- **Model.java**



```

Model.java X
Project 1 > GPA-Calculator > Model.java
1  package application;
2
3  import java.text.DecimalFormat;
4  import java.text.NumberFormat;
5  import java.util.ArrayList;
6  import javafx.collections.FXCollections;
7  import javafx.collections.ObservableList;
8  import javafx.scene.chart.PieChart;
9  import javafx.scene.chart.PieChart.Data;
10 import javafx.scene.chart.XYChart;
11 import javafx.scene.chart.XYChart.Series;
12 import javafx.scene.control.ComboBox;
13 import javafx.scene.control.TextField;
14
15 > public class Model { ...
43 if(grade.equals("A+"))
44 > { ...

```

6. Are descriptive variable and constant names used in accord with naming conventions?

GradeScaled.

24. For every boolean test: Is the correct condition checked?

So sánh $i \leq \text{numberOfRows}$ là sai, sẽ thừa một vòng lặp.

34. Can any nested if statements be converted into a switch statement?

Nhiều if else.

53. Is a standard indentation and layout format used consistently?

Không lùi dòng đúng.

54. For each method: Is it no more than about 60 lines long?

Hàm calculateGPA nhiều hơn 60 dòng.

56. Can the cost of recomputing a value be reduced by computing it once and storing the results?

listOfCredits.get(i).getText() được tính toán nhiều lần trong if.

63. Could any non-local variables be made local?

GradeScaled, classHours, grade.

V. Tổng kết

Trong bài báo cáo lần này, nhóm đã thực hiện kiểm tra lỗi trên 5 file code mẫu. Từng đoạn code có những đặc thù riêng nhưng nếu thiết kế một checklist review code một cách hợp lý thì chúng ta có thể kiểm tra lỗi một cách dễ dàng hơn, hạn chế trong việc sửa lỗi và tránh được những lỗi không đáng có. Nhóm đã sử dụng checklist code review của TMA Solution nhưng đây chỉ là bảng rút gọn, vì vậy có thể còn nhiều sai sót mà nhóm chưa tìm ra. Tuy nhiên có thể hiểu được phần nào về công việc kiểm tra lỗi code.