Student Schedule Table .JAVA :

/unitime/timetable/gwt/client/sectioning/StudentScheduleTable.java

1 BUG and 7 Code enhancements.

**(1) Bugs:**

1)line 327:

break;

A loop with at most one iteration is equivalent to the use of an if statement to conditionally execute one piece of code. No developer expects to find such a use of a loop statement.If the initial intention of the author was really to conditionally execute one piece of code, an if statement should be used instead.but in this code will not modify it because it causes the loop to exit after processing the first ClassAssignment, preventing the creation of additional rows for subsequent ClassAssignment objects.intended to display information about the first class section in the list.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**(2)enhancements:**

1)

line 113:

@Override

public void onSuccess(Boolean result) {}

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.

Methods should not be empty.

////////////////////////////////////////////

2)

line 130:

@Override

public void onSuccess(Boolean result) {}

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.

Methods should not be empty.

//////////////////////////////////////////////

3)

line 182:

@Override

public void onFailure(Throwable caught) {}

Add a nested comment explaining why this method is empty, throw an UnsupportedOperationException or complete the implementation.

Methods should not be empty.

/////////////////////////////////////

3)line 217:

public void populate(Collection<CourseAssignment> data){---}

Cognitive Complexity is a measure of how hard the control flow of a method is to understand. Methods with high Cognitive Complexity will be difficult to maintain.

////////////////////////////////

4)

line231:

ArrayList<WebTable.Row> rows = new ArrayList<WebTable.Row>();

Java 7 introduced the diamond operator (<>) to reduce the verbosity of generics code. For instance, instead of having to declare a List's type in both its declaration and its constructor, you can now simplify the constructor declaration with <>, and the compiler will infer the type.

///////////////////////////////

5)

line 248:

new WebTable.Cell(firstClazz ? course.isFreeTime() ? MESSAGES.freeTimeSubject() : course.getSubject() : "")

Extract this nested ternary operation into an independent statement.

Nesting ternary operators results in the kind of code that may seem clear as day when you write it, but six months later will leave maintainers can't understand anything and the code will be less maintainable.

Instead,on the side of clarity,use another line to express the nested operation as a separate statement.

//////////////////////////////

6)line 260:

new WebTable.Cell(firstClazz ? course.isFreeTime() ? MESSAGES.freeTimeCourse() : course.getCourseNbr() : "")

Extract this nested ternary operation into an independent statement to be more maintainable.

////////////////////////////

7) line 311:

unassignedMessage += ", ";

there are no major differences between StringBuilder and '+'. and it this a minor code effect.