* **Change in requirements**

remove unused function in class department *htmlShortLabel()* this function was never called so it's either a redundant code or it was a draft, so it should be removed to decrease complexity.

Path: target/src/org/unitime/timetable/model/Student.java

* **Change redundant code**

getExams(ExamType examType){//logic}

getExams(Integer examType){//logic}

The same function is implemented twice with the same logic that causes a redundant code.

solution is to use templates (in our case in java we will use Generics)

Path: JavaSource/org/unitime/timetable/model/Student.java

* **Remove inessential code**

*getReleasedPin()* this function was never used in the project and it consists of an if condition only so it is redundant increase the complexity of the project

Path: org/unitime/timetable/model/Student.java

* **Change in the logic of a function**

a dead code *else{}* since the first if only happens when the subject != null then it checks the same condition in second "if" nested in the first "if" so the else will never happens.

Another prediction: we can remove the first if condition so the code will work normally even the subject != null or subject = null

Path: /unitime/JavaSource/org/unitime/timetable/action/ClassesAction.java

* **Removing a code that makes a problem**

This bug happens twice in the same class in line 115 and line 131

115 break; //code

131 break;

Explanation: the break doesn't have any conditional statement to happen so it will happen leading to not iterate in the for loop after the first iteration.

Solution: remove the break or put a condition for it to occur.

Path: JavaSource/org/unitime/timetable/action/StudentSchedulingAction.java