CSC207 A4 Design discussion

Are creating many button classes an efficient design for PhotoRenamer?

In my opinion, the efficiency of having button classes outside PhotoRenamer is strongly

depended on how many buttons there are in the GUI and the complexity of their functions.

If PhotoRenamer contains many buttons (about 10 or more) or each button has a complicated

function, then it is better and useful to have different classes. The strength of this approach is

that it allows developers to fast track the errors that may occur within a specific button without

having to interfere the working codes in other button methods, which greatly decreased the

chance of accidentally changing the working code to not working.

In the other case, if PhotoRenamer only contains a few buttons that does similar things, then a

better alternative can be considered, which is to contain all formatting and the action listeners

within PhotoRenamer, with an additional actionPerformed constructor called every time a

button has been declared and added an action listener. Another way similar to the alternative is

to add action listener with “this” as the argument, and under PhotoRenamer.java, implement an

unique actionperformed method that organizes all button events within one method through

cases. Personally, I prefer the second approach as it has a cleaner format than the first.

In conclusion, I cannot judge the design of having separate button classes as extreme good or

extreme bad as this design has a varying efficiency with dependencies on the buttons and

functions. Nevertheless, I think the major guideline for this design is to use separate classes if

there are many buttons with complicated functions, otherwise use other alternatives.