Software Engineering, Assignment 8

Group 96:

Yuri Manna, i6316134  
Helena Posyniak, i6303009

SMELL 1 - Primitive Obsession

To solve smell 1, we have implemented classes Player, Question and Category. This way we could get rid of primitive variables like places, purses, inPenaltyBox, isGettingOutOfPenaltyBox. Instead of many Linked Lists we create Categories which have Linked lists as properties. Method add() which we renamed to addPlayer() for more clarity now has only 3 lines of code instead of 7. To simplify a variable Player currentPlayer was created which stores which player is the current one, it does so in the GameRunner class by setting the current player with a method - setCurrentPlayer(), which keeps track of a token as a index pointer.

SMELL 2 & SMELL 3 - Long Method & Duplicate Code

This smell was targeted at the method roll() which had around 30 lines of code, after creating move() method, which is responsible for moving the player on the “board”, the old code was duplicated and used in another place as well thus creating one method seemed fit. Similar thing was done in the correctAnswer() method, where we deleted the duplicate code and created the once-AnsweredCorrectly(Player player) method, which executes what happens after a player answers a question correctly and then returns whether the current player is a winner.

SOME CLEANUPS:

* Method add() was of type boolean, which was unnecessary so we changed it to type void
* Method isPlayable() was never implemented, so we implemented it in the GameRunner class
* We rearranged the methods, so they are in executable order, and we added comments/documentation for cosmetic purposes
* We fixed some naming inconsistencies:
  + int[] Purses -> int coins
  + int[] Places -> int place
  + add(String playerName) -> addPlayer(Player player)
  + wasCorrectlyAnswered() -> correctAnswer()
* Reduced number of if statements in currentCategory() method by using &&
* Deleted createRockQuestion(int index) method, because it had no special use and was inconsistent with how other questions were created
* Hardcoded parts of the code were made flexible (number of players, amount of questions)

The results of final unit testA screenshot of a computer

Description automatically generated: