**Code Smells**

1. **Data Class –** contains only data and no real functionality. It inherits certain data just to return it again. Perhaps it can be used as another component in the session class instead of a class.

**In:** src/net/sf/freecol/server/model/LootSession.java

public class LootSession extends Session {

    private static final Logger logger = Logger.getLogger(LootSession.class.getName());

*/\*\* The goods that are available to be captured. \*/*

    private final List<Goods> capture;

    public LootSession(Unit winner, Unit loser, List<Goods> capture) {

        super(makeSessionKey(LootSession.class, winner, loser));

        this.capture = capture;

    }

    @Override

    public boolean complete(ChangeSet cs) {

        return super.complete(cs);

    }

    public List<Goods> getCapture() {

        return capture;

    }

}

1. **Long method** – the main is too long. It has 9202 lines. It takes up most of the class and it is more complex than needs to be. We could do the Extract Method and put certain functionalities in other methods, simplifying the method and making it easier to understand.

**In**: src/net/sf/freecol/tools/ForestMaker.java – main method

public static void main(String[] args) throws IOException {

        if (args.length == 0) {

            System.out.println("Usage: ForestMaker <directory>...");

            System.out.println("Directory name should match a directory in");

            System.out.println("   " + DESTDIR);

            System.exit(1);

        }

        String riverName = "data/rules/classic/resources/images/terrain/"

                + "ocean/center0.png";

(… more code …)

*// sort by y, x coordinate*

                    trees.sort(Comparator.naturalOrder());

                    for (ImageLocation imageLocation : trees) {

                        g.drawImage(imageLocation.image, imageLocation.x, imageLocation.y, null);

                    }

                }

                g.dispose();

                ImageIO.write(base, "png", new File(destinationDirectory,

                                                    sourceDirectory.getName() + counter + ".png"));

            }

        }

3**- Speculative Generality** – the code isn’t used, and it occupies 1/3 of the class. In Agile Development, there should only be the code we currently use. Removing it will improve code clarity

In: src/net/sf/freecol/tools/GenerateDocumentation.java

*/\* Currently unused*

*private static void generateTMX() {*

*Map<String, Map<String, String>> translations = new HashMap<>();*

*for (String name : sourceFiles) {*

*System.out.println("Processing source file: " + name);*

*String languageCode = name.substring(15, name.length() - 11);*

*if (languageCode.isEmpty()) {*

*languageCode = "en";*

*} else if ('\_' == languageCode.charAt(0)) {*

*languageCode = languageCode.substring(1);*

*} else {*

*// don't know what to do*

*continue;*

*}*

(…more commented code)

*out.write("  <tu tuid=\"" + tu.getKey() + "\">\n");*

*for (Map.Entry<String, String> tuv : tu.getValue().entrySet()) {*

*out.write("    <tuv xml:lang=\"" + tuv.getKey() + "\">\n");*

*out.write("      <seg>" + tuv.getValue() + "</seg>\n");*

*out.write("    </tuv>\n");*

*}*

*out.write("  </tu>\n");*

*}*

*out.write("</body>\n");*

*out.write("</tmx>\n");*

*out.flush();*

*} catch (Exception e) {*

*e.printStackTrace();*

*}*

*}*

*\*/*

58427 – Liliane Correia