Code smell 1: Dead code:

Graphical user interface, text, application, email

Description automatically generated

Location:

Rationale: This class is not used or instantiated anywhere in the application.

Refactor proposal: It might be a class that is not used anymore, so it should be deleted.

Text

Description automatically generatedCode smell 2: Speculative Generality:

Location:

Line 99:

Rationale: This catch clause was written thinking in something that might happen in the future but doesn’t happen at this moment.

Refactor proposal: It should be deleted to make the code clearer.

Code Smell 3: Law of Demeter



Location:



Rationale: Here *PerChart*class in its method *updateFonts( )*is callinga method named *getFactor()*, and this method is not:

Encapsulated within the same object – the method *getFactor()*, is from an object of the type *Size*, so it obviously isn’t from the same class as *updateFonts(),*or in other words is not from*PerChart*class*.*

Encapsulated within an object that is in the parameters of *updateFonts()* - the object from which this method is from is received from the method *getSize()*of the *fontSpec* object, and this object is not from the parameters of *updatedFonts()*, it is just returned from a function.

Encapsulated within an object that is instantiated inside *updateFonts()*- the object from which this method is from is received from the method *getSize()*of the *fontSpec* object and this object is not instantiated inside *updateFonts()*, it is just returned from a function.

Encapsulated within an object that is referenced in an instance variable of the class in  *updateFonts()* - the object from which this method is from is received from the method *getSize()*of the *fontSpec* object and not from an instance variable of the class *PerChart*.

Refactor proposal: I would create a method in the FontSpec class that returns the factor since the object is received from the constructor parameters in that class.