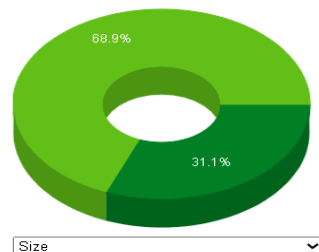
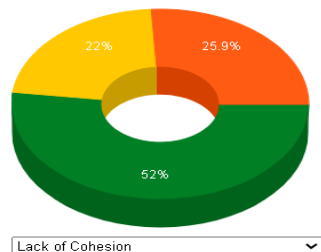
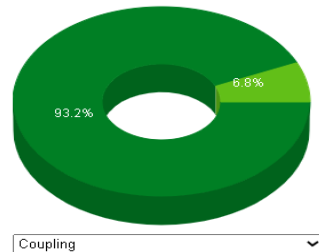
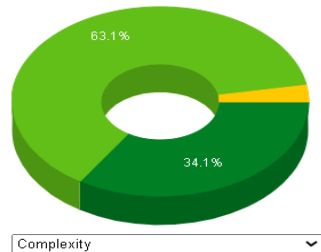


## JavaDesktopSudoku Code Analysis

These diagrams represent the results of the code analysis √ this specific program using codeMr tool of the JetBrains IntelliJ idea.

### Distribution of Quality Attributes

Complexity, Coupling, Cohesion, and Size



### Complexity

Implies being difficult to understand and describes the interactions between a number of entities. Higher levels of complexity in software increase the risk of unintentionally interfering with interactions and so increases the chance of introducing defects when making changes.

### Coupling

Coupling between two classes A and B if:

- A has an attribute that refers to (is of type) B.
- A calls on services of an object B.
- A has a method that references B (via return type or parameter).
- A has a local variable which type is class B.
- A is a subclass of (or implements) class B.

Tightly coupled systems tend to exhibit the following characteristics:

- A change in a class usually forces a ripple effect of changes in other classes.
- Require more effort and/or time due to the increased dependency.
- Might be harder to reuse a class because dependent classes must be included.

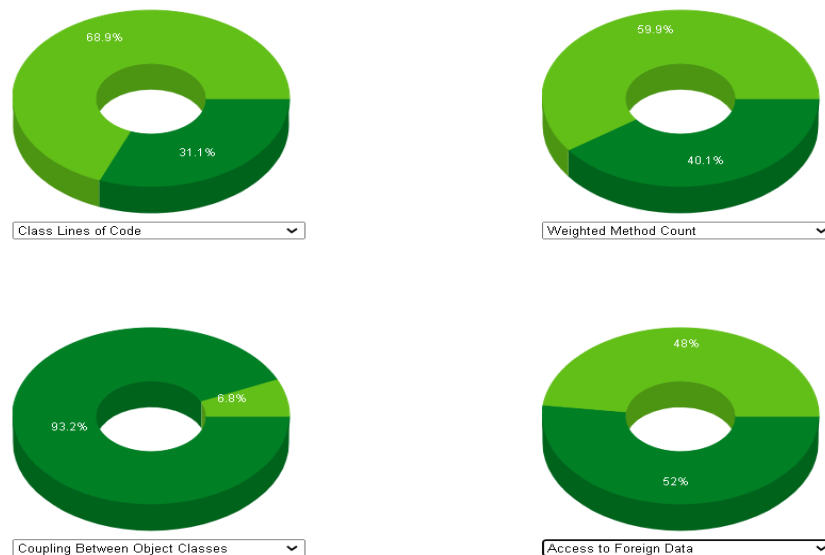
## Lack of Cohesion

Measure how well the methods of a class are related to each other. High cohesion (low lack of cohesion) tend to be preferable, because high cohesion is associated with several desirable traits of software including robustness, reliability, reusability, and understandability. In contrast, low cohesion is associated with undesirable traits such as being difficult to maintain, test, reuse, or even understand.

## Size

Size is one of the oldest and most common forms of software measurement. Measured by the number of lines or methods in the code. A very high count might indicate that a class or method is trying to do too much work and should be split up. It might also indicate that the class might be hard to maintain.

Distribution of Quality Attributes  
Complexity, Coupling, Cohesion, and Size



## Class Lines of Code

Related Quality Attributes: Size

The number of all nonempty, non-commented lines of the body of the class. CLOC is a measure of size and also indirectly related to the class complexity.

## Weighted Method Count

Related Quality Attributes: Complexity, Size

The weighted sum of all class' methods an represents the McCabe complexity of a class. It is equal to number of methods, if the complexity is taken as 1 for each

method. The number of methods and complexity can be used to predict development, maintaining and testing effort estimation. In inheritance if base class has high number of method, it affects its' child classes and all methods are represented in sub-classes. If number of methods is high, that class possibly domain specific. Therefore they are less reusable. Also these classes tend to more change and defect prone.

### Coupling Between Object Classes

Related Quality Attributes: Coupling

The number of classes that a class is coupled to. It is calculated by counting other classes whose attributes or methods are used by a class, plus those that use the attributes or methods of the given class. Inheritance relations are excluded. As a measure of coupling CBO metric is related with reusability and testability of the class. More coupling means that the code becomes more difficult to maintain because changes in other classes can also cause changes in that class. Therefore these classes are less reusable and need more testing effort.

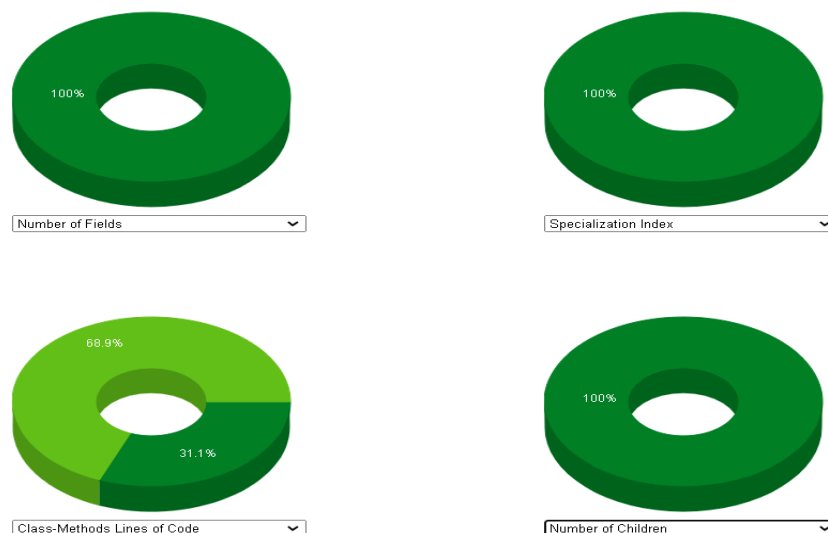
### Access to Foreign Data

Related Quality Attributes: Coupling

ATFD (Access to Foreign Data) is the number of classes whose attributes are directly or indirectly reachable from the investigated class. Classes with a high ATFD value rely strongly on data of other classes and that can be the sign of the God Class.

#### Distribution of Quality Attributes

Complexity, Coupling, Cohesion, and Size



### Number of Fields

Related Quality Attributes: Size

The number of fields (attributes) in a class

### Specialization Index

Related Quality Attributes: Complexity

Defined as  $NORM * DIT / NOM$ . The Specialization Index metric measures the extent to which subclasses override their ancestors classes. This index is the ratio between

the number of overridden methods and total number of methods in a Class, weighted by the depth of inheritance for this class. Lorenz and Kidd precise : Methods that invoke the superclass' method or override template are not included.

### Class-Methods Lines of Code

Related Quality Attributes: Size

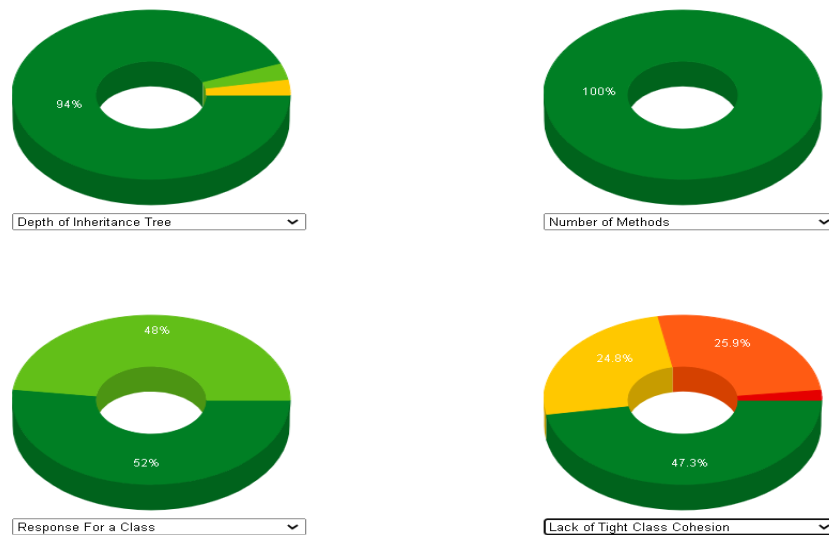
Total number of all nonempty, non-commented lines of methods inside a class.

### Number of Children

Related Quality Attributes: Coupling

The number of direct subclasses of a class. The size of NOC approximately indicates how an application reuses itself. It is assumed that the more children a class has, the more responsibility there is on the maintainer of the class not to break the children's behaviour. As a result, it is harder to modify the class and requires more testing.

Distribution of Quality Attributes  
Complexity, Coupling, Cohesion, and Size



### Depth of Inheritance Tree

Related Quality Attributes: Complexity

The position of the class in the inheritance tree. Has 0 (zero) value for root and non-inherited classes. For the multiple inheritance, the metric shows the maximum length. Deeper class in the inheritance tree, probably inherit. Therefore, it is harder to predict its behavior. Also this class relatively complex to develop, test and maintain.

### Number of Methods

Related Quality Attributes: Size

The number of methods in a class.

### Response For a Class

Related Quality Attributes: Complexity

The number of the methods that can be potentially invoked in response to a public message received by an object of a particular class. It includes the full call graph of any method called from the originating method. If the number of methods that can be

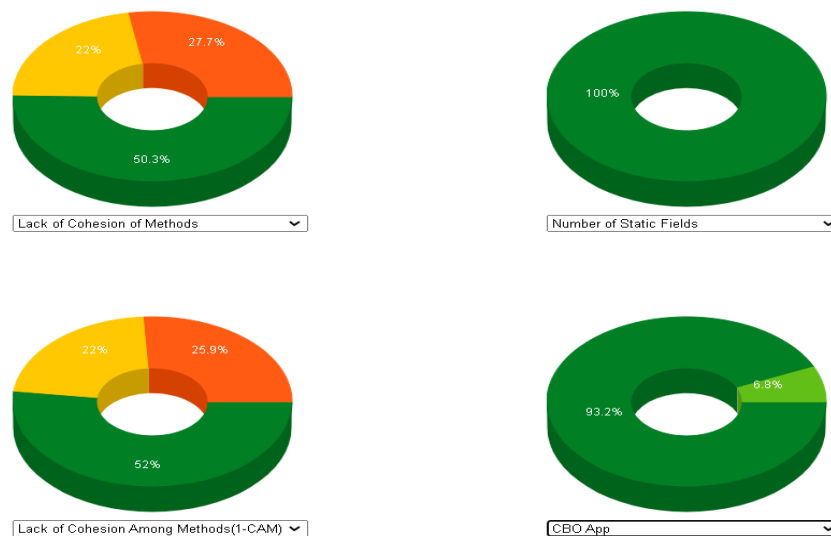
invoked at a class is high, then the class is considered more complex and can be highly coupled to other classes. Therefore more test and maintain effort is required.

### Lack of Tight Class Cohesion

Related Quality Attributes: Cohesion

The Lack of Tight Class Cohesion metric measures the lack cohesion between the public methods of a class. That is the relative number of directly connected public methods in the class. Classes having a high lack of cohesion indicate errors in the design.

Distribution of Quality Attributes  
Complexity, Coupling, Cohesion, and Size



### Lack of Cohesion of Methods

Related Quality Attributes: Cohesion

Measure how methods of a class are related to each other. Low cohesion means that the class implements more than one responsibility. A change request by either a bug or a new feature, on one of these responsibilities will result change of that class. Lack of cohesion also influences understandability and implies classes should probably be split into two or more subclasses. LCOM3 defined as follows  $LCOM3 = (m - \sum(mA)/a) / (m - 1)$  where :

- m number of procedures (methods) in class
- a number of variables (attributes) in class. a contains all variables whether shared (static) or not.
- mA number of methods that access a variable (attribute)
- $\sum(mA)$  sum of mA over attributes of a class

LCOM3 varies between 0 and 2. Values 1..2 are considered alarming. In a normal class whose methods access the class's own variables, LCOM3 varies between 0 (high cohesion) and 1 (no cohesion). When LCOM3=0, each method accesses all variables. This indicates the highest possible cohesion. LCOM3=1 indicates extreme lack of cohesion. In this case, the class should

be split.

When there are variables that are not accessed by any of the class's methods,  $1 < LCOM3 \leq 2$ . This happens if the variables are dead or they are only accessed outside the class. Both cases represent a design flaw. The class is a candidate for rewriting as a module. Alternatively, the class variables should be encapsulated with accessor methods or properties. There may also be some dead variables to remove. If there are no more than one method in a class, LCOM3 is undefined. If there are no variables in a class, LCOM3 is undefined. An undefined LCOM3 is displayed as zero  
[<http://www.aivosto.com/project/help/pm-oo-cohesion.html>]

### **Number of Static Fields**

Related Quality Attributes: Size

The number of static fields in a class.

### **Lack of Cohesion Among Methods(1-CAM)**

Related Quality Attributes: Cohesion

CAM metric is the measure of cohesion based on parameter types of methods.  $LCAM = 1 - CAM$

### **CBO App**

Related Quality Attributes: Coupling

The number of dependent classes in the application.