Escuela
Superior
de Informática

UNIVERSITY OF CASTILLA-LA MANCHA
ESI CIUDAD REAL

SOFTWARE ENGINEERING

GRADE ON COMPUTER SCIENCE

Maintenance Lab

Enrique Garrido Pozo
Pablo Mora Herreros
Adrián Ollero Jiménez
Beka Bekeri
Daniele Acquaviva
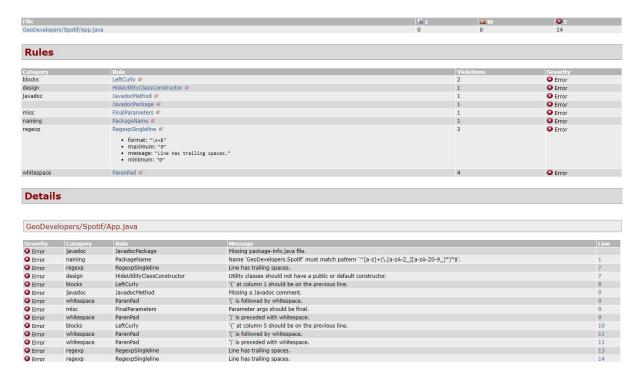
Contents

1 Setup Configuration

In order to be able to achieve our maintenance plan, we have added some plugins to obtain the errors to improve our Spotif project. To obtain this, we have created a new module called *build-tools* to get all the information in the same file inside *Spotif target*. The plugins we have added are *pmd*, *findbugs*, *checkstyle*, *jxr* and *javadoc*. Here, it is the code to add the plugins.

```xml
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-pmd-plugin</artifactId>
    <version>3.8</version>
    <configuration>
        <rulesets>
            <ruleset>pmd-ruleset.xml</ruleset>
        </rulesets>
        <printFailingErrors>true</printFailingErrors>
    </configuration>
    <executions>
        <execution>
            <goals>
                <goal>check</goal>
            </goals>
        </execution>
    </executions>
    <dependencies>
        <dependency>
            <groupId>GeoDevelopers</groupId>
            <artifactId>build-tools</artifactId>
            <version>1.0</version>
        </dependency>
    </dependencies>
</plugin>
```

```xml
    <plugin>
<groupId>org.apache.maven.plugins</groupId>
<artifactId>maven-pmd-plugin</artifactId>
<version>3.8</version>
<configuration>
  <rulesets>
     <ruleset>pmd-ruleset.xml</ruleset>
  </rulesets>
</configuration>
</plugin>
    <plugin> <groupId>org.codehaus.mojo</groupId> <artifactId>findbugs-maven-plugin</artifactId>
      <configuration> <skipEmptyReport>false</skipEmptyReport> </configuration>
      </plugin>
    <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-checkstyle-plugin</artifactId>
      <configuration> <skipEmptyReport>false</skipEmptyReport> </configuration>
    </plugin> <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-jxr-plugin</artifactId>
    </plugin> <plugin> <groupId>org.apache.maven.plugins</groupId> <artifactId>maven-javadoc-plugin</artifactId>
      </plugin>
```

2 Maintenance Plan

Firstly, we are going to run the project to obtain all the errors. When we see them, we will decide which of them are the most important to correct. We have seen the information clicking in each

type of error and the different ways to resolve them.

File		I		W		E
GeoDevelopers/Spotif/App.java		0		0		14

Rules

Category	Rule	Violations	Severity
blocks	LeftCurly	2	❌ Error
design	HideUtilityClassConstructor	1	❌ Error
javadoc	JavadocMethod	1	❌ Error
	JavadocPackage	1	❌ Error
misc	FinalParameters	1	❌ Error
naming	PackageName	1	❌ Error
regexp	RegexpSingleline	3	❌ Error
	• format: "\s+$" • maximum: "0" • message: "Line has trailing spaces." • minimum: "0"		
whitespace	ParenPad	4	❌ Error

Details

GeoDevelopers/Spotif/App.java

Severity	Category	Rule	Message	Line
❌ Error	javadoc	JavadocPackage	Missing package-info.java file.	
❌ Error	naming	PackageName	Name 'GeoDevelopers.Spotif' must match pattern '^[a-z]+(\.[a-zA-Z_][a-zA-Z0-9_]*)*$'.	1
❌ Error	regexp	RegexpSingleline	Line has trailing spaces.	7
❌ Error	design	HideUtilityClassConstructor	Utility classes should not have a public or default constructor.	7
❌ Error	blocks	LeftCurly	'{' at column 1 should be on the previous line.	8
❌ Error	javadoc	JavadocMethod	Missing a Javadoc comment.	9
❌ Error	whitespace	ParenPad	'(' is followed by whitespace.	9
❌ Error	misc	FinalParameters	Parameter args should be final.	9
❌ Error	whitespace	ParenPad	')' is preceded with whitespace.	9
❌ Error	blocks	LeftCurly	'{' at column 5 should be on the previous line.	10
❌ Error	whitespace	ParenPad	'(' is followed by whitespace.	11
❌ Error	whitespace	ParenPad	')' is preceded with whitespace.	11
❌ Error	regexp	RegexpSingleline	Line has trailing spaces.	13
❌ Error	regexp	RegexpSingleline	Line has trailing spaces.	14

We have found that GeoDevelopers/Spotif/App.java is not necessary for us so, we are going to delete this one.

For the rest of modules in the file Checkstyle we have found 1315 errors.

Summary

Files	Info	Warnings	Errors
40	0	0	1315

Files

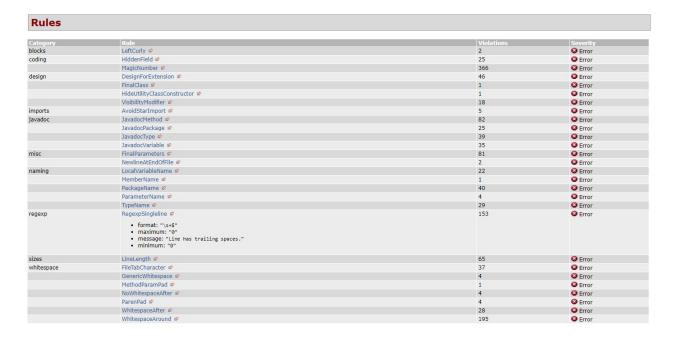| File	I	W	E
Domain/Album.java	0	0	56
Domain/Album_it3.java	0	0	45
Domain/DaoAlbum.java	0	0	34
Domain/DaoAlbum_it3.java	0	0	16
Domain/DaoAlbum_it5.java	0	0	17
Domain/DaoAlbum_it7.java	0	0	32
Domain/DaoSong.java	0	0	37
Domain/DaoSong_it3.java	0	0	16
Domain/DaoSong_it5.java	0	0	16
Domain/DaoSong_it7.java	0	0	31
Domain/DaoUser.java	0	0	11
Domain/DaoUser_it4.java	0	0	18
Domain/DaoUser_it6.java	0	0	17
Domain/DaoUser_it8.java	0	0	9
Domain/Message.java	0	0	64
Domain/Song.java	0	0	43
Domain/Song_it3.java	0	0	44
Domain/User.java	0	0	33
Domain/User_it4.java	0	0	33
Domain/User_it5.java	0	0	19
GeoDevelopers/Spotif/App.java	0	0	14
Persistence/DBBroker.java	0	0	12
Persistence/DBBroker_it2.java	0	0	83
Persistence/DBBroker_it3.java	0	0	5
Persistence/DBBroker_it4.java	0	0	6
Persistence/DBBroker_it5.java	0	0	6
Persistence/DBBroker_it6.java	0	0	5
Persistence/DBBroker_it7.java	0	0	5
Persistence/DBBroker_it8.java	0	0	8
Presentation/AppAdmin_it5.java	0	0	50
Presentation/AppAdmin_it6.java	0	0	57
Presentation/AppAdmin_it7.java	0	0	67
Presentation/AppAdmin_it8.java	0	0	77
Presentation/AppInterface.java	0	0	33
Presentation/AppInterface_it3.java	0	0	39
Presentation/AppInterface_it4.java	0	0	42
Presentation/AppInterface_it5.java	0	0	43
Presentation/AppInterface_it7.java	0	0	55
Presentation/AppInterface_it8.java	0	0	64
Presentation/LoginInterface.java	0	0	53

The most habitual errors in our project are the MagicNumber of coding, regexp and whitespaces among others. So, the priority of this plan is from the larger number of errors to the lowest one, but some of them we aren't going to correct because we couldn't check them due to the code.

Rules

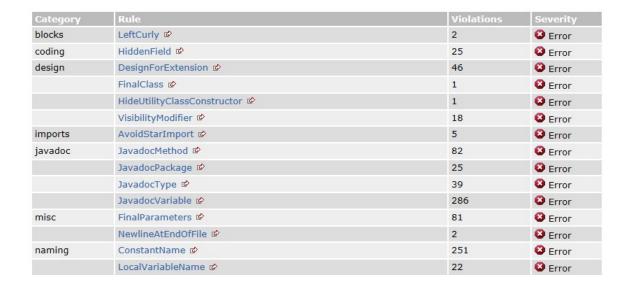| Category	Rule	Violations	Severity
blocks	LeftCurly	2	Error
coding	HiddenField	25	Error
	MagicNumber	366	Error
design	DesignForExtension	46	Error
	FinalClass	1	Error
	HideUtilityClassConstructor	1	Error
	VisibilityModifier	18	Error
imports	AvoidStarImport	5	Error
javadoc	JavadocMethod	82	Error
	JavadocPackage	25	Error
	JavadocType	39	Error
	JavadocVariable	35	Error
misc	FinalParameters	81	Error
	NewlineAtEndOfFile	2	Error
naming	LocalVariableName	22	Error
	MemberName	1	Error
	PackageName	40	Error
	ParameterName	4	Error
	TypeName	29	Error
regexp	RegexpSingleline • format: "\s+$" • maximum: "0" • message: "Line has trailing spaces." • minimum: "0"	153	Error
sizes	LineLength	65	Error
whitespace	FileTabCharacter	37	Error
	GenericWhitespace	4	Error
	MethodParamPad	1	Error
	NoWhitespaceAfter	4	Error
	ParenPad	4	Error
	WhitespaceAfter	28	Error
	WhitespaceAround	195	Error

3   Errors corrected

We start with the MagicNumber. These errors occur when we have a numeric literal that is not defined as a constant. It improves readability of the code and it's easier to maintain. This error is usually in the presentation package.

When we have corrected all the MagicNumber errors, we have discovered that the number of errors have increased with the constant names. We are going to check these errors.

Summary

Files	Info	Warnings	Errors
40	0	0	1547

In this picture we can see how this changes occur:

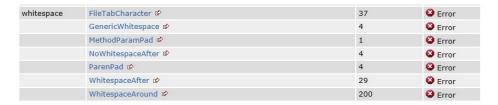| Category	Rule	Violations	Severity
blocks	LeftCurly	2	Error
coding	HiddenField	25	Error
design	DesignForExtension	46	Error
	FinalClass	1	Error
	HideUtilityClassConstructor	1	Error
	VisibilityModifier	18	Error
imports	AvoidStarImport	5	Error
javadoc	JavadocMethod	82	Error
	JavadocPackage	25	Error
	JavadocType	39	Error
	JavadocVariable	286	Error
misc	FinalParameters	81	Error
	NewlineAtEndOfFile	2	Error
naming	ConstantName	251	Error
	LocalVariableName	22	Error

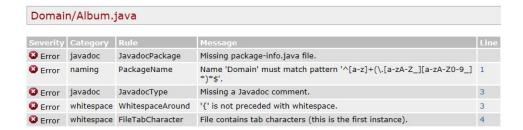To correct these new errors, we have to change the format of the naming.

```
private JFrame frame;
static final int X_FRAME = 100;
static final int Y_FRAME = 100;
static final int WIDTH_FRAME = 320;
static final int HEIGHT_FRAME = 281;
static final int X_TXT = 24;
static final int Y_TXT = 34;
static final int WIDTH_TXT = 150;
static final int HEIGHT_TXT = 197;
static final int X_LBL = 24;
static final int Y_LBL = 9;
static final int WIDTH_LBL = 46;
static final int HEIGHT_LBL = 14;
static final int Y_ADDSONG = 34;
static final int Y_ADDALBUM = 68;
static final int X_BTN = 204;
static final int WIDTH_BTN = 89;
static final int HEIGHT_BTN = 23;
```

We have seen that many classes extends from others, so we remove some constants to reduce the code.

Other error we are going to check is the whitespaces which have increased modifying the others. This is easy to reduce and it shorts the code. There are some types of whitespaces:

whitespace	FileTabCharacter ⇗	37	❌ Error
	GenericWhitespace ⇗	4	❌ Error
	MethodParamPad ⇗	1	❌ Error
	NoWhitespaceAfter ⇗	4	❌ Error
	ParenPad ⇗	4	❌ Error
	WhitespaceAfter ⇗	29	❌ Error
	WhitespaceAround ⇗	200	❌ Error

To locate the error like in the other examples, we have to see in the file, the line where the error appears. This is an example:

Domain/Album.java

Severity	Category	Rule	Message	Line
❌ Error	javadoc	JavadocPackage	Missing package-info.java file.	
❌ Error	naming	PackageName	Name 'Domain' must match pattern '^[a-z]+(\.[a-zA-Z_][a-zA-Z0-9_]*)*$'.	1
❌ Error	javadoc	JavadocType	Missing a Javadoc comment.	3
❌ Error	whitespace	WhitespaceAround	'{' is not preceded with whitespace.	3
❌ Error	whitespace	FileTabCharacter	File contains tab characters (this is the first instance).	4

At the end, we have reduce some of whitespaces errors. Other kind of error we will check is the regexp, which detect singles line. Deleting the properties disappears.

Now, we will dedicate with the javadoc category because it will give us valious information.
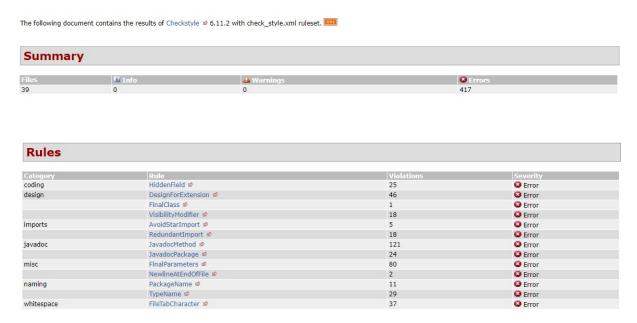
Other error we think is important for the environment is renaming the name of the packages and it is easy to change. But, we will maintain the Presentation name package because others errors will appear if we modify them. We have modified some variables names too.

In terms of length, we will change the rules to be the maximum 255 in each line in our check-style.xml file where we have put our rules.

The rest of errors to check for us they aren't important or we can't correct them. For example, the imports we will maintain because we will continue maintaining the project and we will add more imports if we need and the javadoc are uncompleted because the project will change.

4 Conclusions

Finally, we have reduced the errors in more than the half and some errors have been reduced.

The following document contains the results of Checkstyle ⏎ 6.11.2 with check_style.xml ruleset. XML

Summary

Files	Info	Warnings	Errors
39	0	0	417

Rules

Category	Rule	Violations	Severity
coding	HiddenField ⏎	25	Error
design	DesignForExtension ⏎	46	Error
	FinalClass ⏎	1	Error
	VisibilityModifier ⏎	18	Error
imports	AvoidStarImport ⏎	5	Error
	RedundantImport ⏎	18	Error
javadoc	JavadocMethod ⏎	121	Error
	JavadocPackage ⏎	24	Error
misc	FinalParameters ⏎	80	Error
	NewlineAtEndOfFile ⏎	2	Error
naming	PackageName ⏎	11	Error
	TypeName ⏎	29	Error
whitespace	FileTabCharacter ⏎	37	Error

If you have any problem seeing the reports, download all the *maintenance* branch. Although, you can find some information in the wiki.

The first version of the reports are in the *testing* branch whereas the new version is in the *maintenance* branch.