

Laboratorium 5

1. Prezentacje:

- Sonar Cube
- Apache POI/ Commons

Po wygłoszeniu referat do wysłania mailem (format PDF + edytowalny) na adres:

skrzynia@agh.edu.pl, tytuł maila:

[PZ1] Referat Apache Commons/POI – grupa *\$grupa\$* – nazwiska autorów oddzielone przecinkiem

przy czym *\$grupa\$* przyjmuje wartości: pn. 18:30, śr. 8, śr. 9:45, śr. 11:30

np.:

[PZ1] Referat Apache Commons/POI – grupa pn. 18:30 – Nowak, Kowalski

Referat należy też KONIECZNIE udostępnić kolegom/koleżankom z grupy.

2. Przemyśl swój projekt zadania 1: zapewnij żeby można było rozszerzać funkcjonalności programu hotelowego bez zmiany istniejącego kodu (dodawać obsługę nowych komend bez jakiegokolwiek zmiany w istniejącym kodzie: wykorzystaj mechanizmy dziedziczenia i generics – zobacz na instrukcję pomocniczą).
3. Na bazie informacji z referatu z lab 4 spróbuj skonfigurować analizator statyczny kodu Sonar Cube i dokonać analizy pokrycia swojego kodu testami (kod zadania 1).
4. Podepnij analizę do *maven* (<https://docs.sonarqube.org/latest/analysis/scan/sonarscanner-for-maven>), zapewnij badania pokrycia kodu testami.
5. Dokonaj analizy raportu dla bieżącego stanu Twojego projektu nr 1.
6. Dokonaj refaktoryzacji kodu na podstawie raportu w celu wyeliminowania problemów klasy *critical* i *major*.

Wskazówki

Problem badania pokrycia testami w SonarCube (więcej o JaCoCo:

<https://www.eclemma.org/jacoco/trunk/doc/maven.html>).

1. W macierzystym pom.xml w sekcji properties dodaj:

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.target>1.8</maven.compiler.target>
  <maven.compiler.source>1.8</maven.compiler.source>
  <sonar.login>admin</sonar.login>
  <sonar.password>HasloDoSonar</sonar.password>
  <version.surefire.plugin>3.2.2</version.surefire.plugin>
</properties>
```

Oraz opcjonalnie:

```
<project.testresult.directory>${project.build.directory}/surefire-
reports</project.testresult.directory>
<sonar.jacoco.reportPaths>${project.testresult.directory}/coverage/jacoco/jacoco.ex
ec</sonar.jacoco.reportPaths>
<sonar.junit.reportsPath>${project.testresult.directory}</sonar.junit.reportsPath>
<sonar.java.codeCoveragePlugin>jacoco</sonar.java.codeCoveragePlugin>
<sonar.coverage.jacoco.xmlReportPaths>${project.build.directory}/target/site/jacoco
-aggregate/jacoco.xml</sonar.coverage.jacoco.xmlReportPaths>
```

W sekcji dependencies:

```
<dependencies>
  <dependency>
    <groupId>org.junit.platform</groupId>
    <artifactId>junit-platform-launcher</artifactId>
    <version>1.3.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-engine</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.junit.jupiter</groupId>
    <artifactId>junit-jupiter-api</artifactId>
    <version>5.3.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-csv</artifactId>
    <version>1.10.0</version>
  </dependency>
</dependencies>

<plugin>
  <artifactId>maven-site-plugin</artifactId>
  <version>3.7.1</version>
</plugin>

<plugin>
  <artifactId>maven-project-info-reports-plugin</artifactId>
  <version>3.0.0</version>
</plugin>
```

oraz pluginy w sekcji <build><pluginManagement> (główny pom i potomne):

```
<plugin>
  <groupId>org.sonarsource.scanner.maven</groupId>
  <artifactId>sonar-maven-plugin</artifactId>
  <version>${sonar.version}</version> <!-- można podać 3.9.1.2184 -->
</plugin>
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>${version.surefire.plugin}</version>
  <dependencies>
    <dependency>
      <groupId>org.junit.platform</groupId>
      <artifactId>junit-platform-surefire-provider</artifactId>
      <version>1.3.2</version>
    </dependency>
    <dependency>
      <groupId>org.junit.jupiter</groupId>
      <artifactId>junit-jupiter-engine</artifactId>
      <version>${version.junit.jupiter}</version>
    </dependency>
  </dependencies>
</plugin>

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.11.0</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
  </configuration>
</plugin>
```

```

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-javadoc-plugin</artifactId>
  <version>3.3.2</version>
  <configuration>
    <source>17</source>
    <encoding>UTF-8</encoding>
  </configuration>
  <executions>
    <execution>
      <id>aggregate</id>
      <goals>
        <goal>aggregate</goal>
      </goals>
    </execution>
  </executions>
</plugin>

<plugin>
  <groupId>org.jacoco</groupId>
  <artifactId>jacoco-maven-plugin</artifactId>
  <version>0.8.7</version>
</plugin>

```

W sekcji <profiles>:

```

<profiles>
  <profile>
    <id>coverage</id>
    <build>
      <plugins>
        <plugin>
          <groupId>org.jacoco</groupId>
          <artifactId>jacoco-maven-plugin</artifactId>
          <version>0.8.7</version>
          <executions>
            <execution>
              <id>prepare-agent</id>
              <goals>
                <goal>prepare-agent</goal>
              </goals>
            </execution>
            <execution>
              <id>report</id>
              <goals>
                <goal>report</goal>
              </goals>
              <configuration>
                <formats>
                  <format>XML</format>
                </formats>
              </configuration>
            </execution>
          </executions>
        </plugin>
      </plugins>
    </build>
  </profile>

```

2. W macierzystym pom.xml oraz pom.xml każdego modułu należy jeszcze dodać plugin na profilu.

```

<profiles>
  <profile>
    <id>attributed</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
    <build>

```

```

    <plugins>
      <plugin>
        <groupId>org.jacoco</groupId>
        <artifactId>jacoco-maven-plugin</artifactId>
        <version>0.8.11</version>
        <executions>
          <execution>
            <id>jacoco-initialize</id>
            <goals>
              <goal>prepare-agent</goal>
            </goals>
          </execution>
          <execution>
            <id>jacoco-site</id>
            <phase>package</phase>
            <goals>
              <goal>report</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
</profile>
</profiles>

```

Możesz ew. też spróbować dodać plik ***sonar-project.properties*** do głównego katalogu projektu z następującą zawartością:

```

sonar.projectKey=UnikalnyKluczProjektu
sonar.projectName=NazwaProjektu
sonar.projectVersion=1.0
sonar.sources=src
sonar.tests=src/test
sonar.java.coveragePlugin=jacoco
sonar.junit.reportPaths=target/surefire-reports

```