Plugin-uri testare structurală Java în Intellij IDEA https://www.jetbrains.com/help/idea/running-test-with-coverage.html#coverage-run-configurations:

- integrated coverage runner
- Emma
- JaCoCo

Există tool-uri similare pentru alte IDE-uri, respectiv limbaje.

EclEmma https://www.jacoco.org/index.html - is Eclipse plugin based on Java Code Coverage Library called **JaCoCo** that performs analysis of Java bytecode.

Description of coverage counters provided by JaCoCo can be found in its documentation www.jacoco.org/jacoco/trunk/doc/counters.html.

As you can see in it - JaCoCo and hence EclEmma provide

- instructions coverage
- branch coverage
- line coverage and
- cyclomatic complexity

Setări PIT IntellijIDEA (după executarea cu succes a testelor unitare):

Run -> Edit -> Configurations -> Templates -> PIT Runner -> + -> PIT Runner ->

Name: PIT

Target classes: Pachet1.*

Source dir: .../src

Report dir: default

Other params: --outputFormats XML,HTML --targetTests Pachet2.*

A se vedea imaginile `set_pit_intellij_1.png` și `set_pit_intellij_2.png`

Setup SDK11 dacă este implicit SDK13.

PIT funcționează pentru un proiect Java cu structura indicată în imaginea `structura_proiect_run_junit.png` dacă se utilizează JUnit4.

Pentru JUnit5 urmați instrucțiunile de configurare pentru un proiect cu Maven, respectiv Gradle:

https://github.com/pitest/pitest-junit5-plugin

https://www.baeldung.com/java-mutation-testing-with-pitest

Pentru a omorî mutanți println, dar nu numai:

https://stefanbirkner.github.io/system-rules/

Trebuie să descărcați fișierul jar

https://stefanbirkner.github.io/system-rules/download.html apoi

să îl adăugați proiectului astfel:

Click dreapta pe nume proiect -> Open Module Settings -> Library -> + -> selectați calea fișierului -> OK

A se vedea imaginea `add_systemrules_jar_intellij.png`