

Sumar

Lista de Figuri.....	1
1. Instalarea plugin-ului SonarLint.....	1
2. Configurarea utilizării SonarLint.....	2
3. Utilizarea SonarLint pentru analiza statică a codului sursă, bazată pe reguli.....	2

Lista de Figuri

Figure 1 instalarea plugin-ului pentru utilizarea SonarLint.....	1
Figure 2. Setări generale pentru SonarLint	2
Figure 3 Utilizarea SonarLint la nivelul unei clase Java	3

1. Instalarea plugin-ului SonarLint

- În IntelliJ IDEA, meniul **File** ---> **Settings** ---> secțiunea **Plugins**, se completează în câmpul *Search* numele plugin-ului **SonarLint**;
- Dacă rezultatele căutării nu apar imediat, se alege opțiunea *Search in Marketplace*;
- se instalează plugin-ul (vezi Figure 1), cu repornirea IDE-ului.

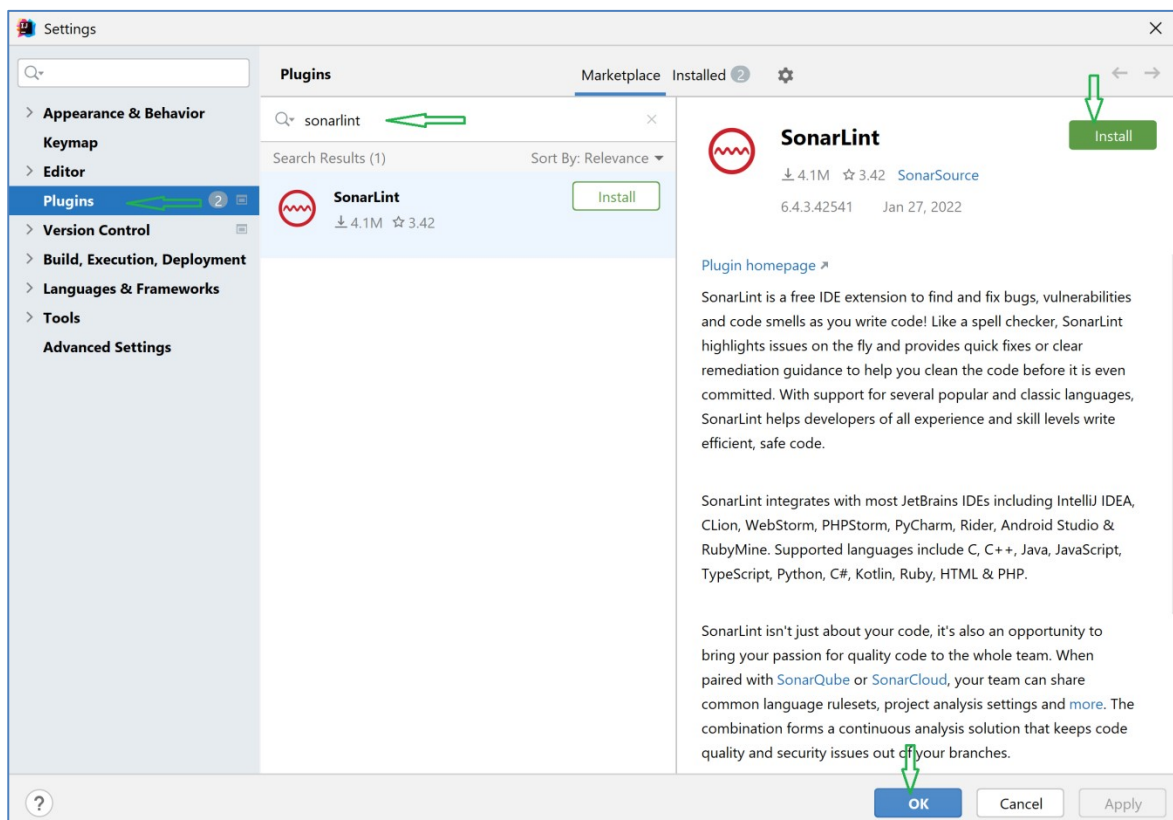


Figure 1 instalarea plugin-ului pentru utilizarea SonarLint

2. Configurarea utilizării SonarLint

- În IntelliJ IDEA, meniul **File** ---> **Settings** ---> **Tools** ---> **SonarLint**, se poate configura utilizarea **SonarLint**:
 - la nivel general;
 - la nivelul proiectului curent;
- În tab-ul **Rules** se poate realiza (vezi Figure 2):
 - vizualizarea** regulilor de evaluare a calității codului sursă asociate unui limbaj de programare, e.g., Java, cât și a descrierilor acestora;
 - activarea/dezactivarea** unor reguli particulare.

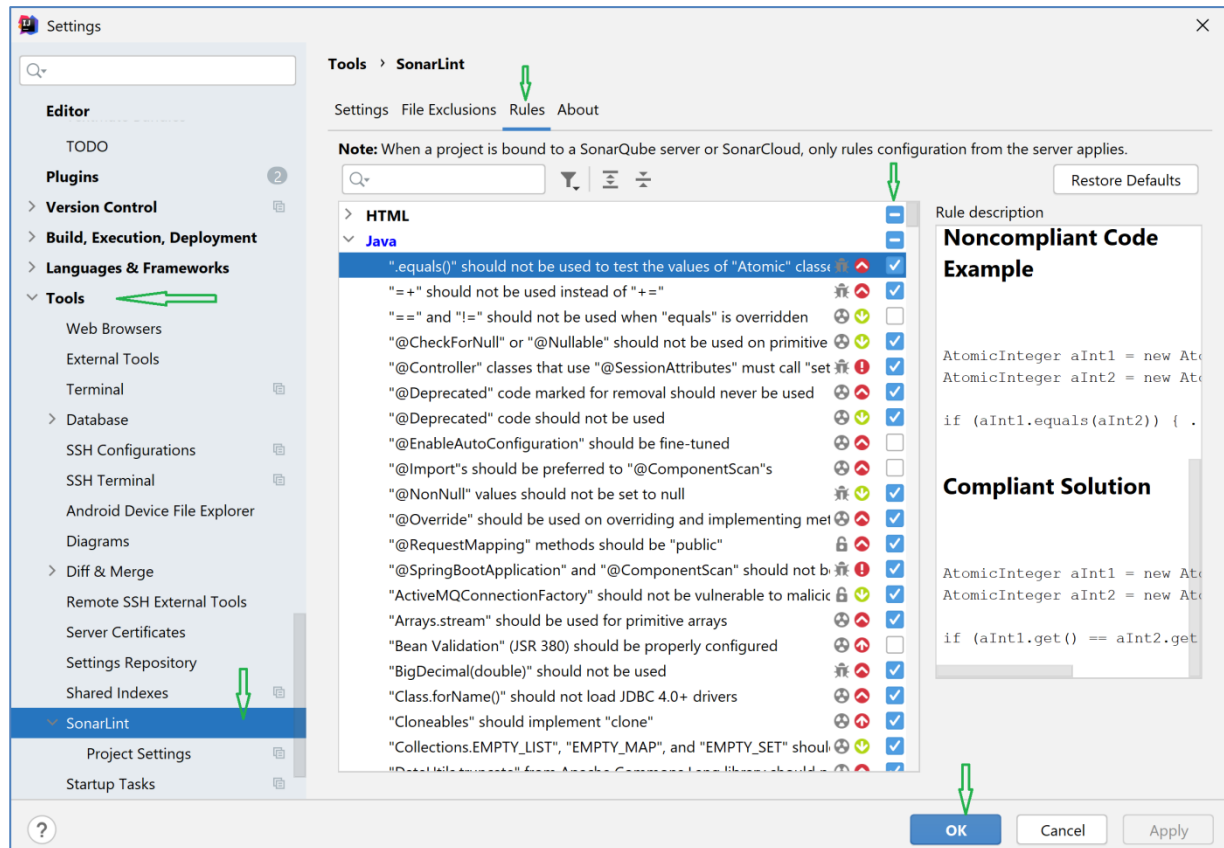


Figure 2. Setări generale pentru SonarLint

3. Utilizarea SonarLint pentru analiza statică a codului sursă, bazată pe reguli

- Fereastra **SonarLint** se poate activa din meniul **View**---> **Tool Windows**---> **SonarLint**;
- Se inițiază analiza statică a codului sursă curent folosind opțiunea **Analyze** (butonul verde);
- După finalizarea analizei rezultă o listă de *code smells*, obținute pe baza unor reguli predefinite activate (vezi Figure 2), cu niveluri diferite de severitate, e.g., *minor*, *major*, *critical*, *blocker*. Aceste code smells pot fi analizate și eliminate ulterior (Figure 3).

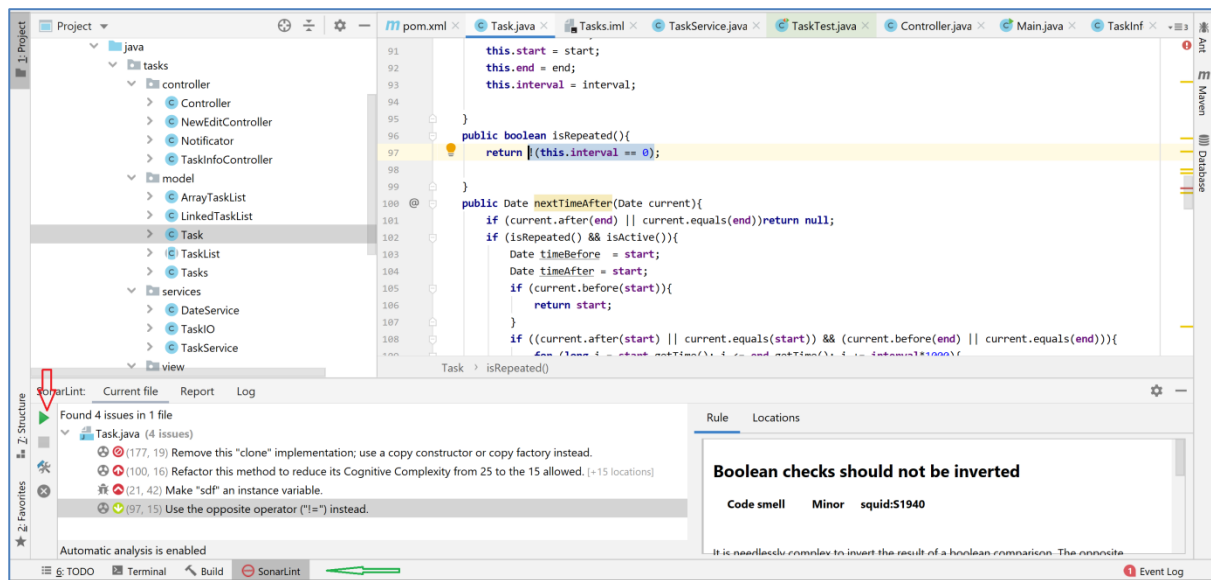


Figure 3 Utilizarea SonarLint la nivelul unei clase Java