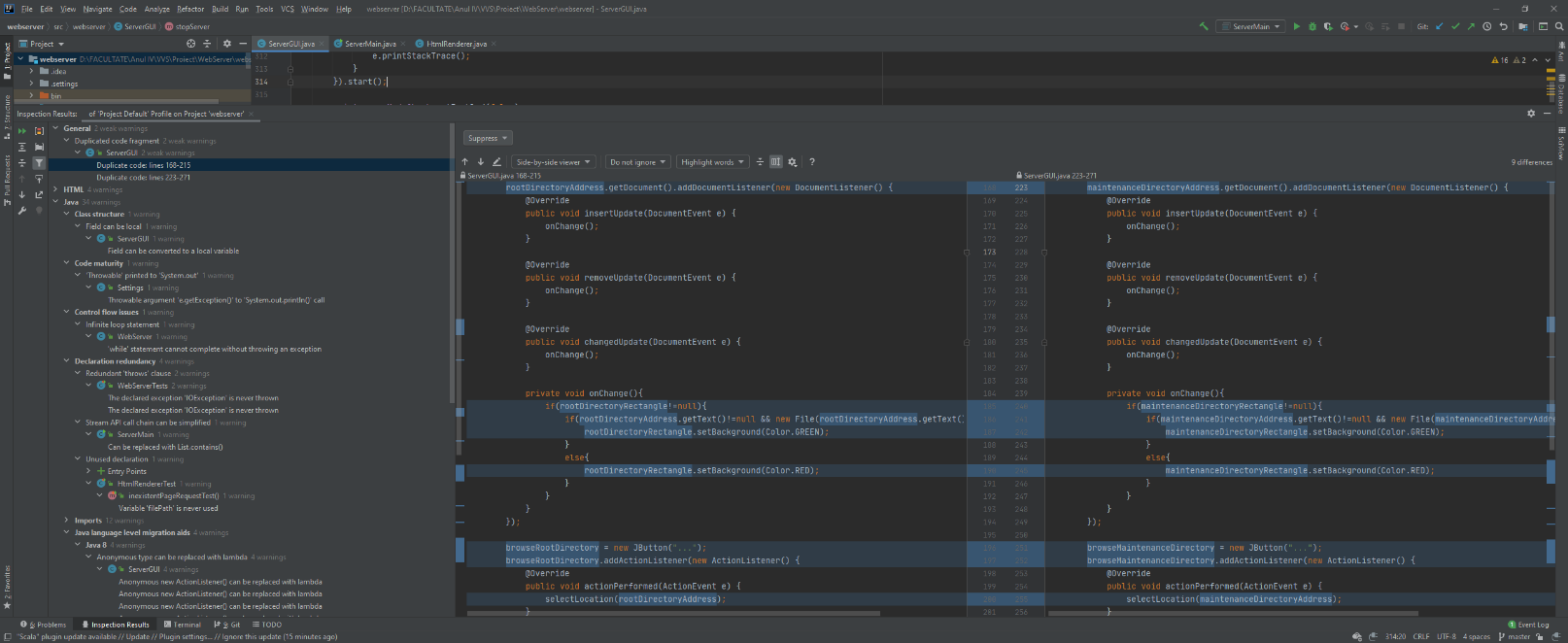
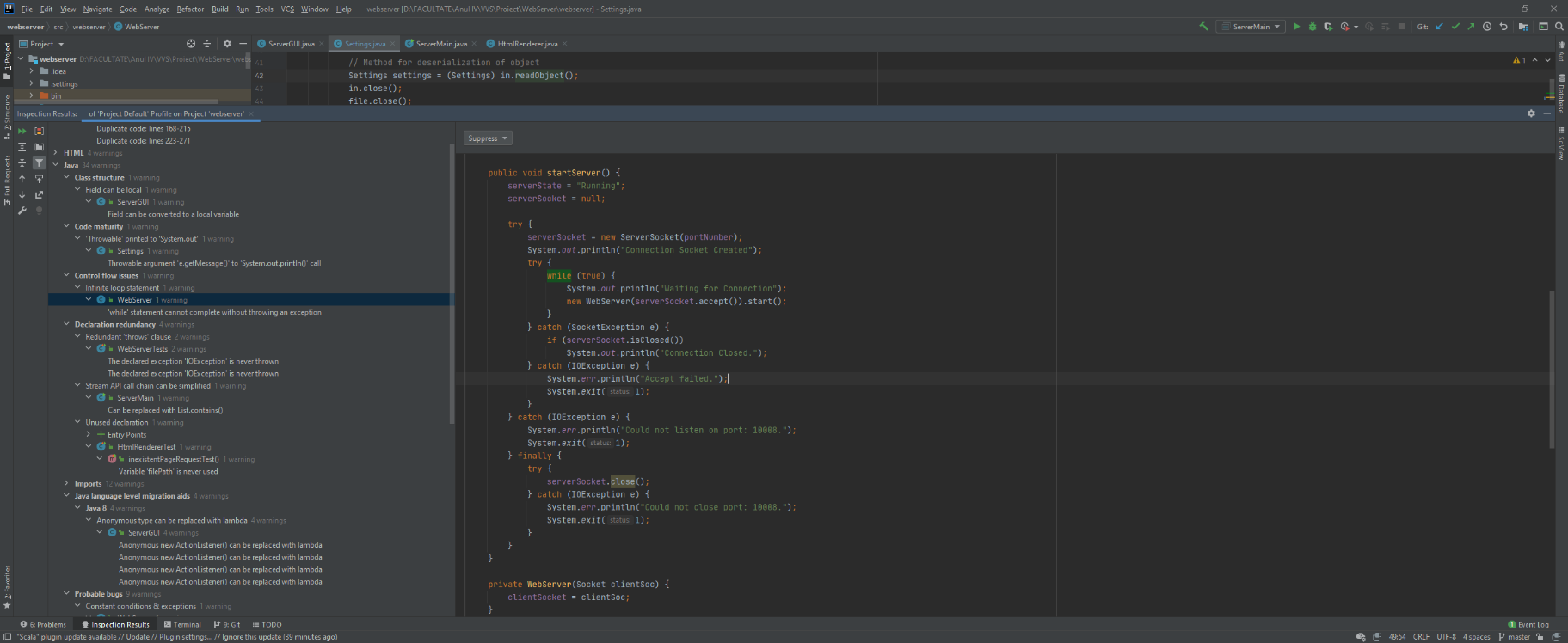
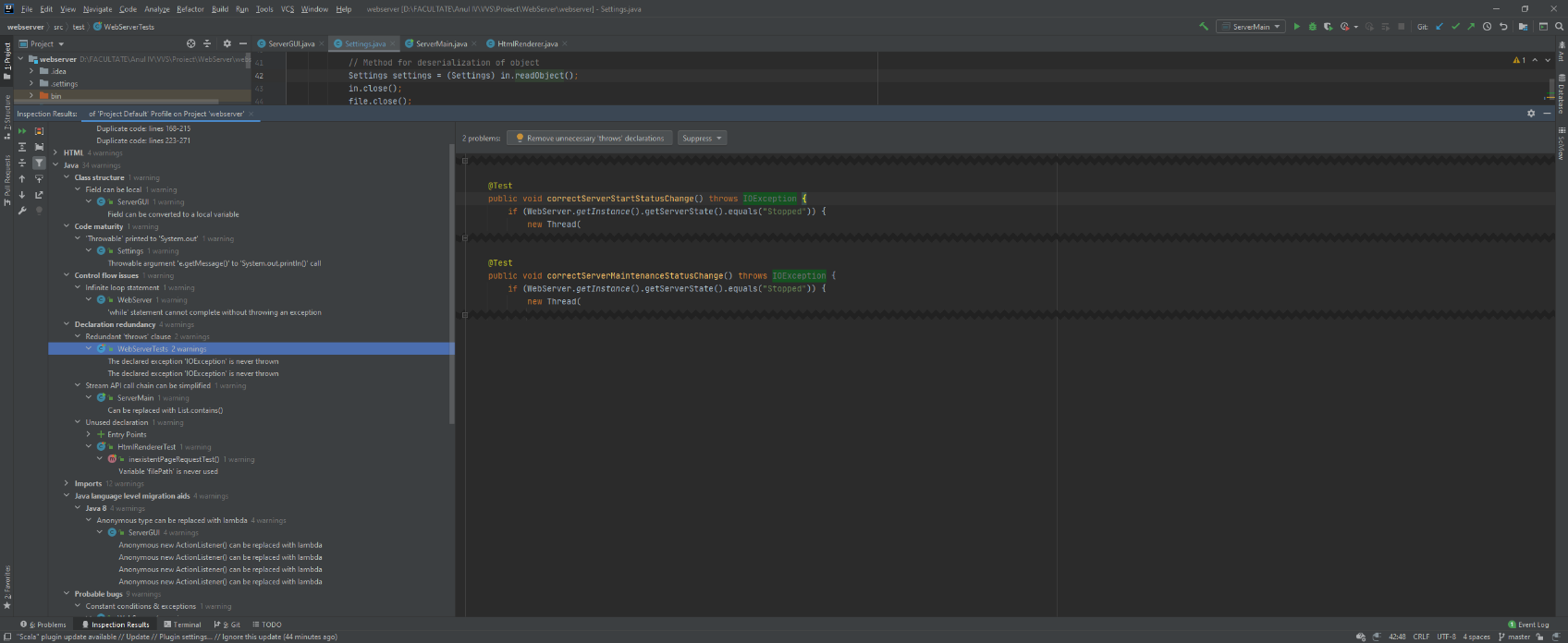
**Analiză Statică**

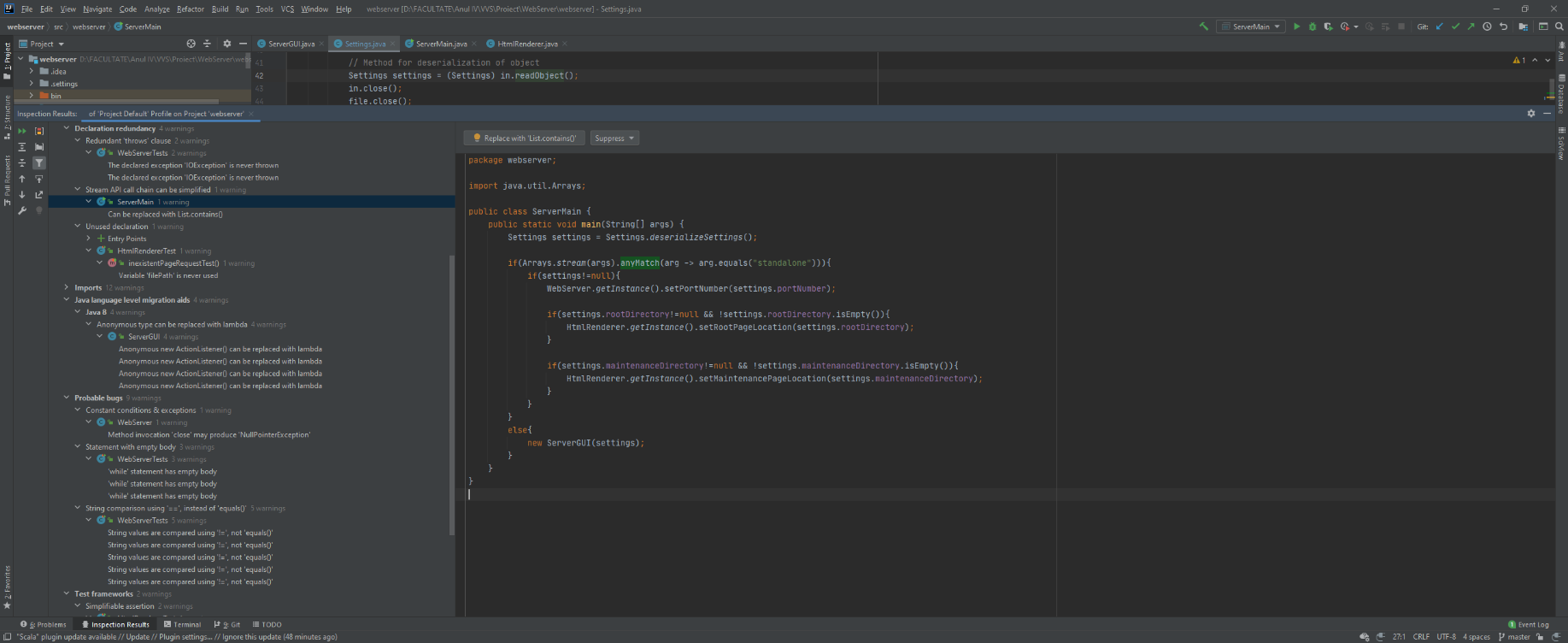
1. Fragmente de cod duplicate

Aceste fragmente de cod sunt detectate ca fiind duplicate, însă ele aparțin de două butoane diferite, cele două butoane de browse prezente în interfață. Aceste porțiuni de cod nu vor fi modificate, deoarece ele salvează datele în obiecte și atribute diferite, astfel am nevoie de ele.

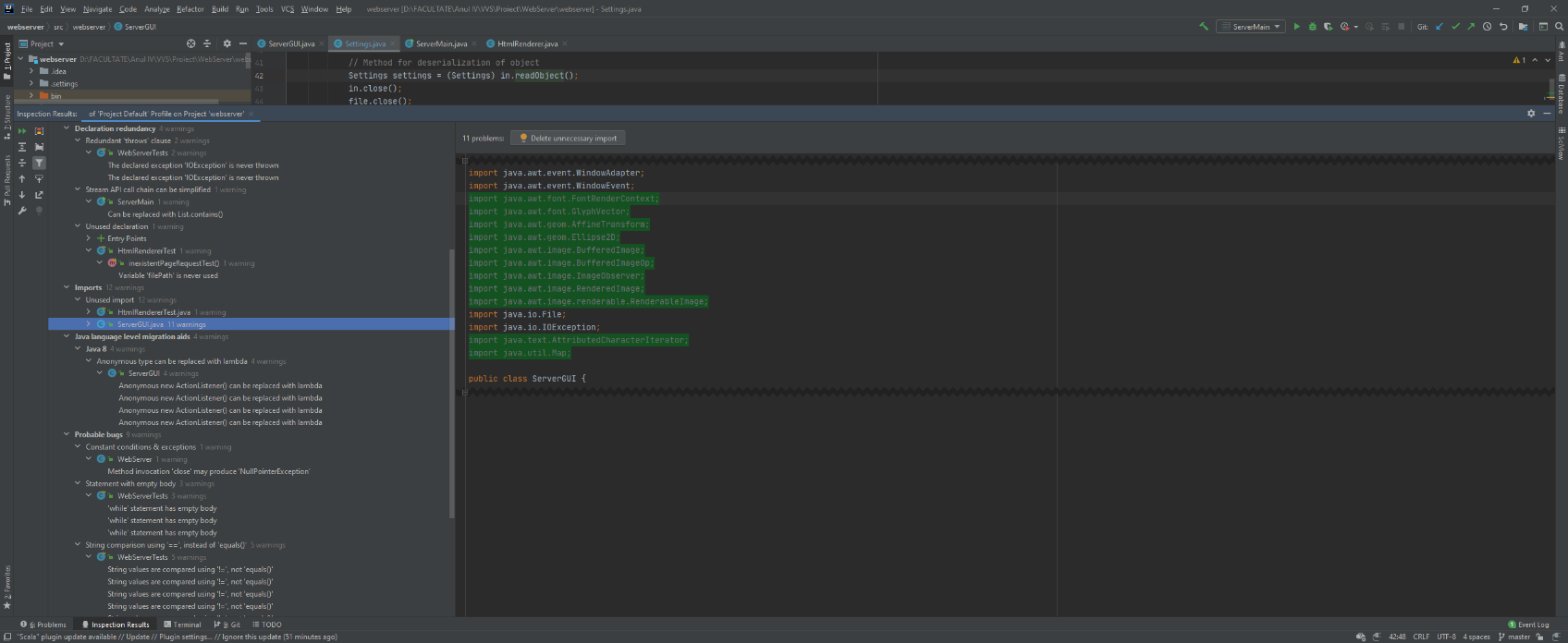
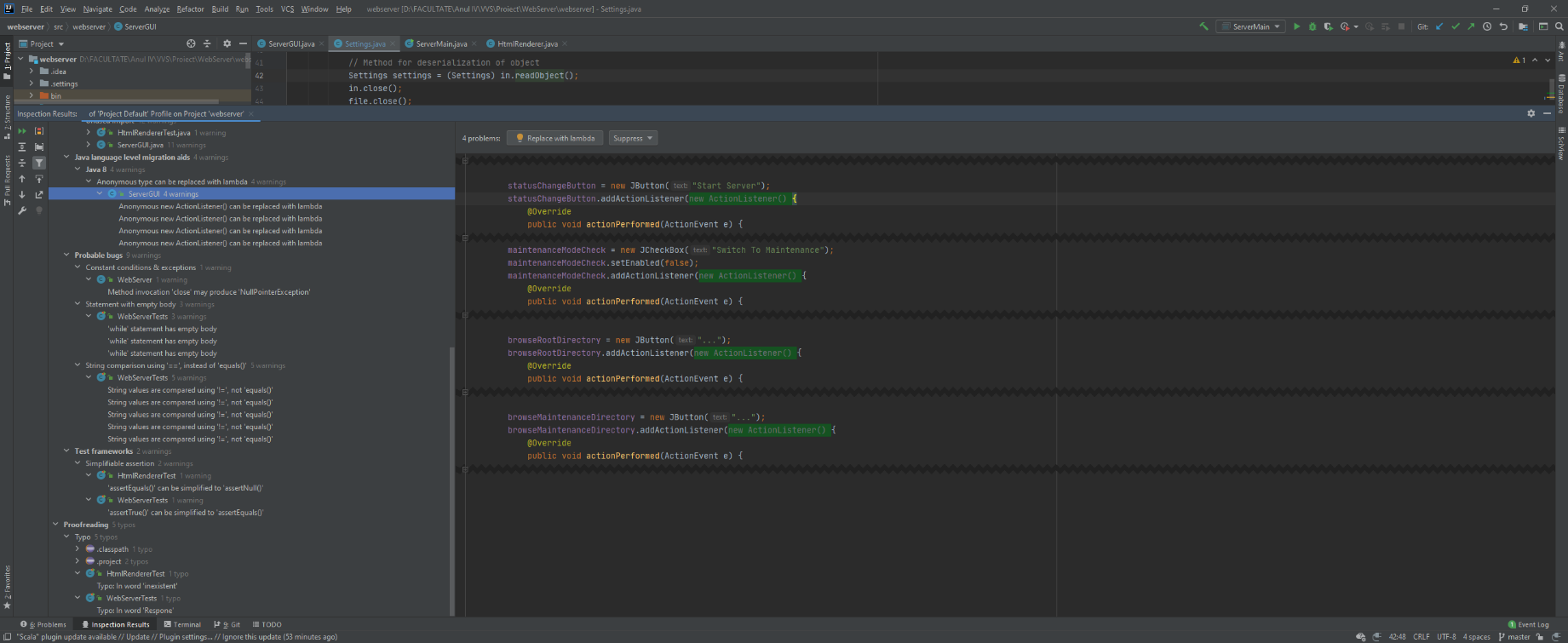
1. Control Flow

Serverul rulează într-ul ciclu infinit, acceptând conexiuni pe socketul pe care e pornit în cazul în care e pornit. Astfel acest ciclu infinti este dorit.

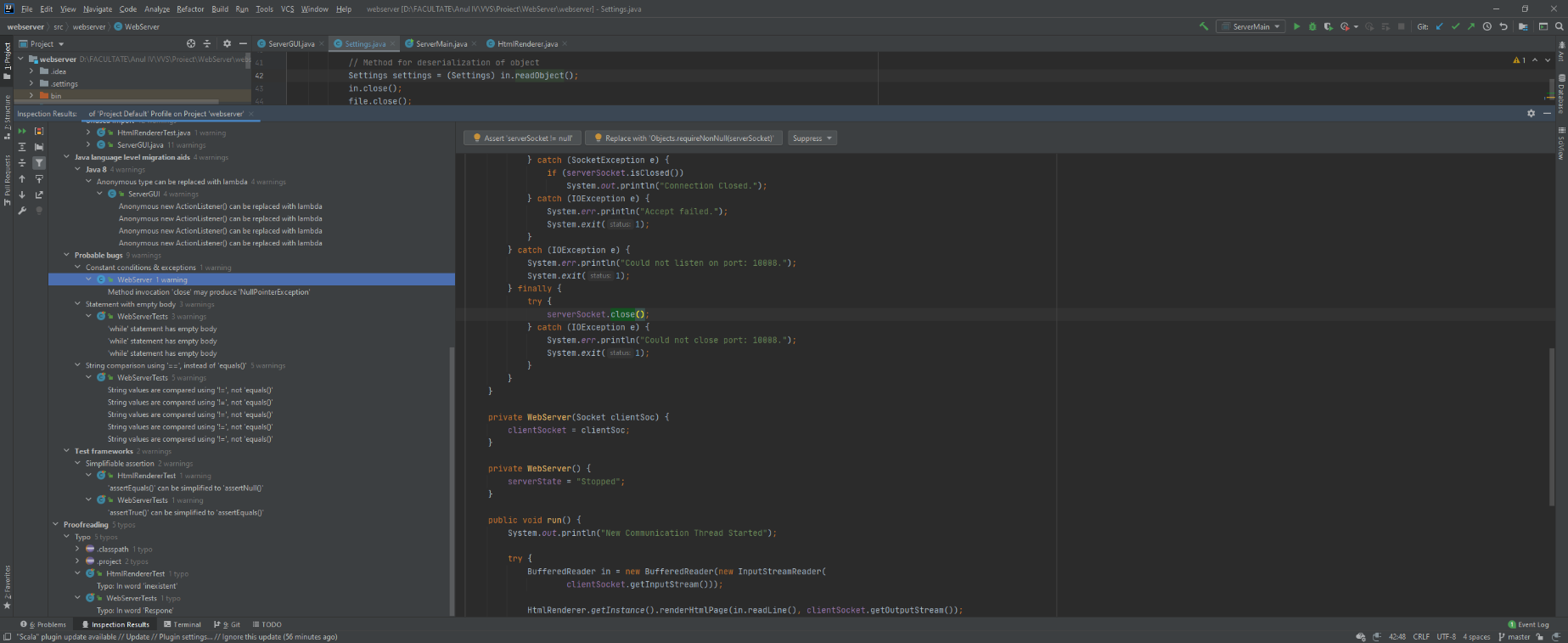
1. Probleme de reduntanță

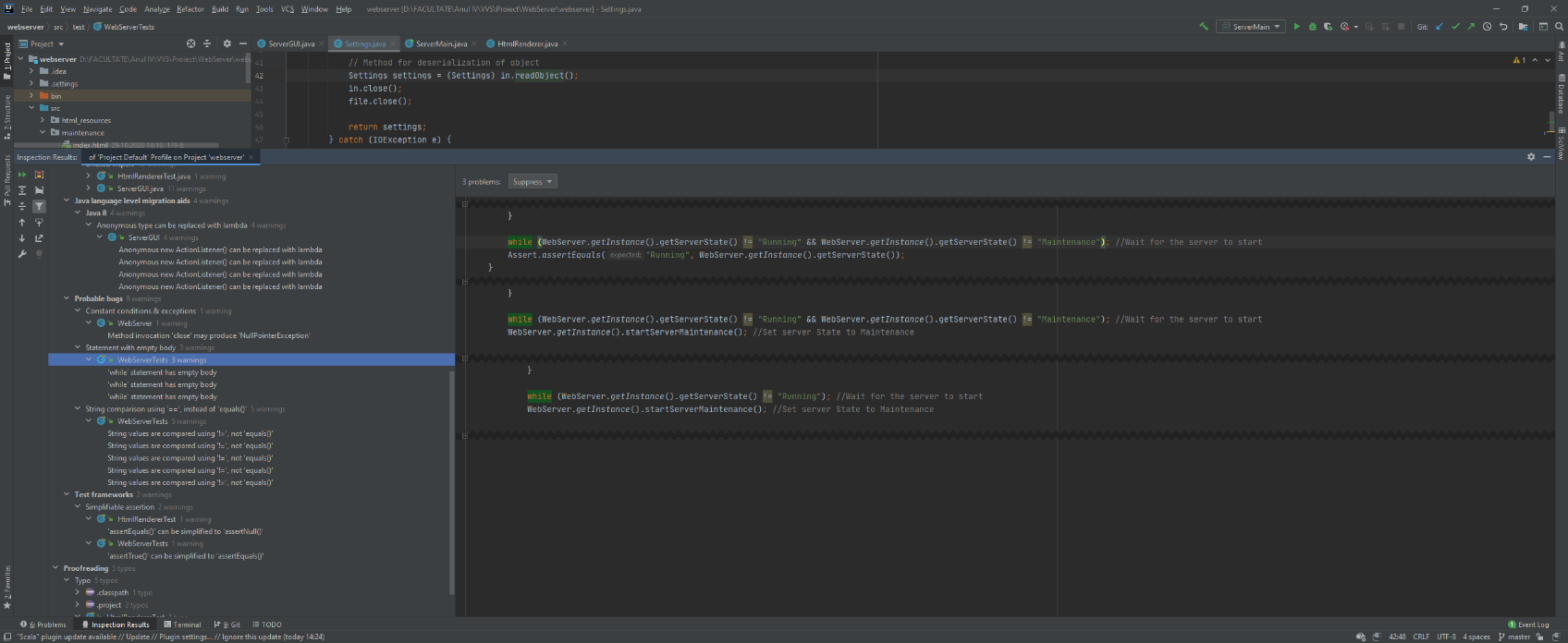
Clauzele throws nu mai sunt folosite, asta făcându-le degeaba. Astfel ele vor fi scoase.

Funcția anyMatch nu este necesară, o abordare mult mai ușoară ar fi folosirea funcției contains() din biblioteca List.

1. Importuri nefolosite
2. Simplificări posibile de cod

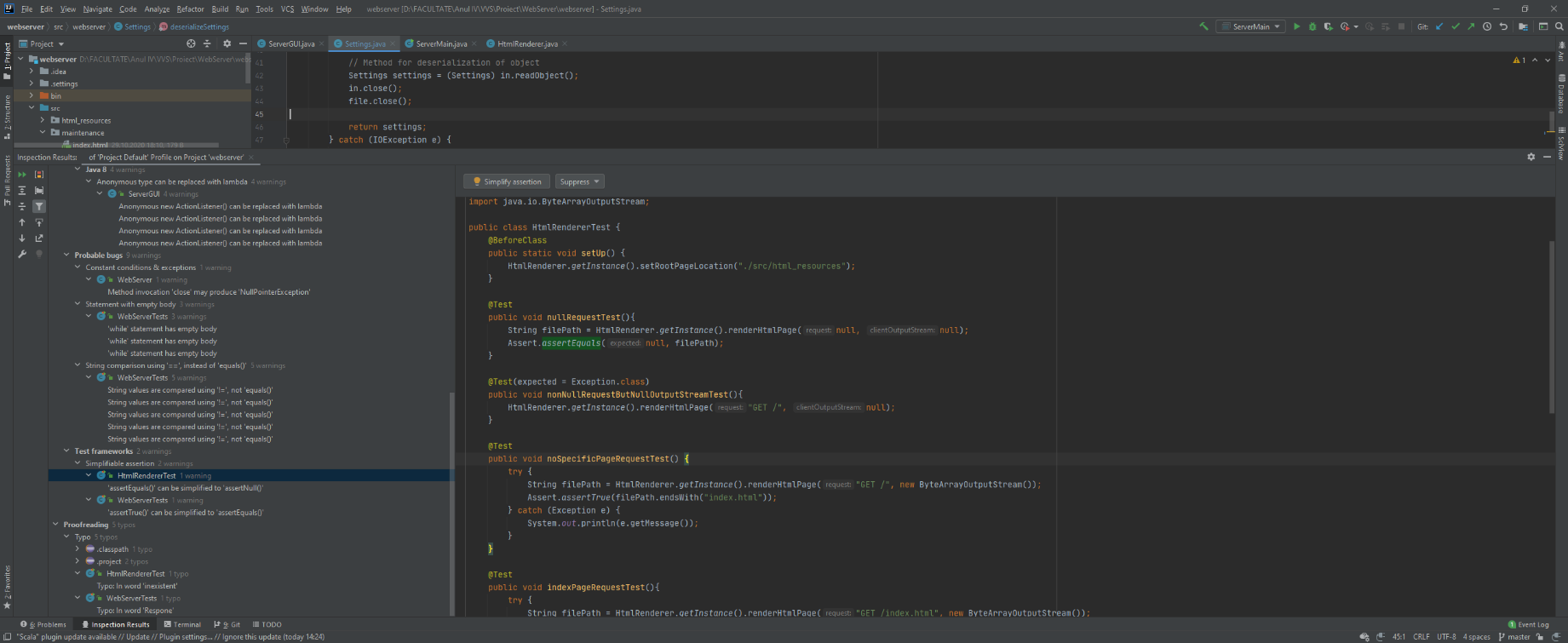
Aceste Action Listenere pot fi înlocuite cu un lambda expression, pentru a nu creea obiecte degeaba.

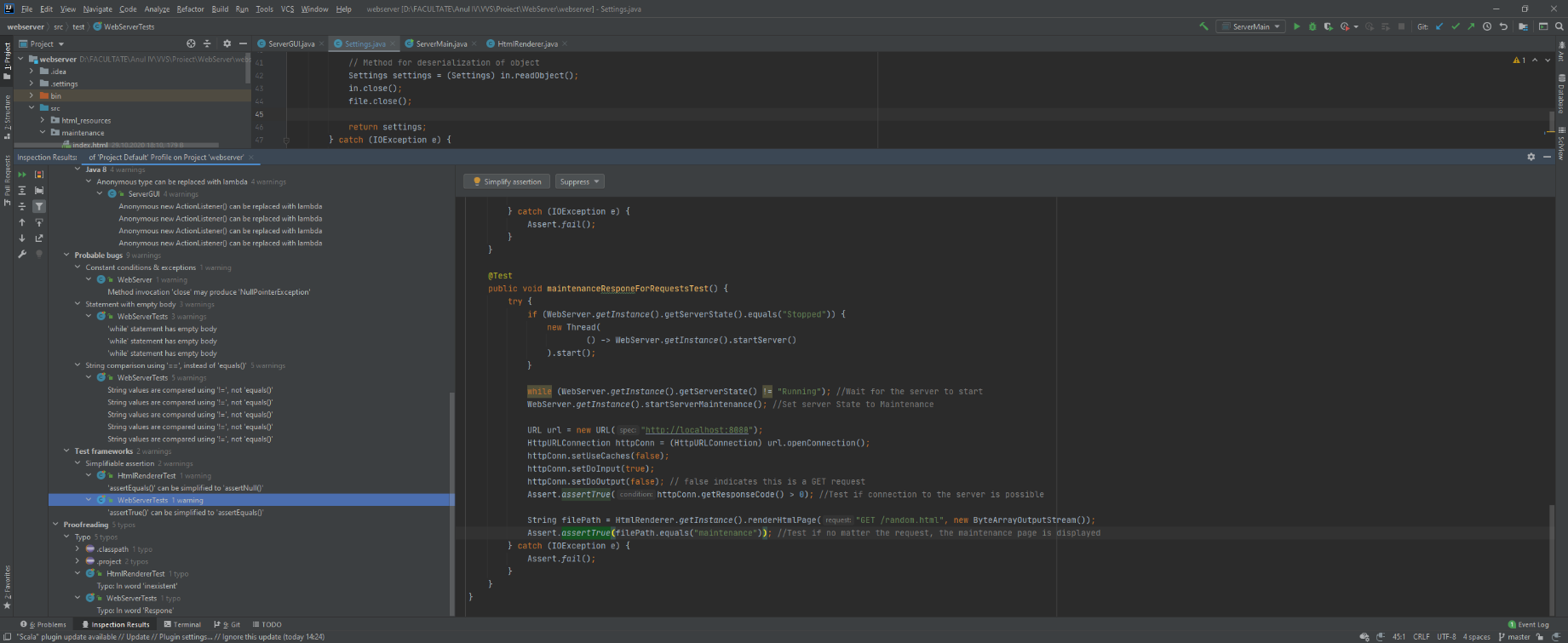
1. Posibile Bug-uri

Apelul close() pe socket poate arunca Null Pointer Exception, aceasta nefiind tratată explicit în cod.

Aceste while-uri pot prezenta două probleme. Una dintre ele e ca ele au un corp gol. Acest comportament este ceva dorit, însă, deoarece ele acționează ca niste wait-uri până în momentul în care serverul îsi schimbă starea.

Cea de-a doua problemă o prezintă condițiile puse pe logica negativă. Această abordare este o problemă, deoarece, pot apărea cazuri neașteptate în care acea condiție să fie îndeplinită. Aceste condiții vor fi modificate astfel încât ele să se îndeplinească doar când starea serverului este una anume, nu atunci când starea serverului este diferită de alte stări decât cea dorită, cum sunt puse acum.

1. Probleme de testare

Acest assertEquals poate fi înlocuit cu un assertNull.

Acest assertTrue poate fi înlocuit cu un assertEquals.