Assignment - Metrika, pregled i statička analiza

* LOC

Broj linija koda za sve fajlove zbirno iznosi 214 (Calculator.java i Start.java 188 + 26)

* Ciklomatska složenost metoda evaluateExpression i Calculate

Koristeći alat Codalyze u Visual Studio Code izračunata je ciklomatska složenost za metodu evaluateExpression i ona iznosi 12. Također, izračunata je i ciklomatska složenost za metodu Calculate i ona isto iznosi 12.

* Kognitivna složenost metoda evaluateExpression i Calculate

Što se tiče kognitivne složenosti metode evaluateExpression možemo zaključiti da je ona relativno visoka. Metoda obavlja više koraka kako bi izračunala vrijednost izraza koji je zadat kao ulazni parametar. Složenost se dodatno povećava zbog činjenice da metoda mora obrađivati različite slučajeve ovisno o prisutnosti različitih aritmetičkih operacija u izrazu. Ovo zahtijeva pažljivo praćenje što otežava razumijevanje koda.

Kognitivna kompleksnost metode "Calculate" je, također, visoka. Metoda koristi različite algoritme za obradu matematičkih operacija (množenje, dijeljenje, zbrajanje, oduzimanje) i manipulaciju listama "numbers" i "operations" kako bi se izračunao konačni rezultat. Također, postoji nekoliko if-else blokova koji se koriste za određivanje redoslijeda izvođenja operacija, što može dodatno povećati kognitivnu kompleksnost metode.

* Pregled

Nakon neformalnog pregleda jednostavnog programa za računanje osnovnim matematičkim operacijama: sabiranje, oduzimanje, množenje, dijeljenje zaključeno je sljedeće:

1) Program je napisan Java jezikom

2) Sastoji se iz dva fajla: Start.jave i Calculator.jave. Ova dva fajla su dio istog programa, gde je Start.java klasa koja sadrži glavnu metodu "main" koja se izvršava prilikom pokretanja programa i putem nje se vrši interakcija sa korisnikoma, dok je "Calculator.java" klasa koja sadrži logiku za izvršavanje matematičkih operacija prilikom unosa korisnika u konzoli "Start.jave" i gdje se ispisuje rezultat.

3) Izvršavanje matematičkih operacija kroz unos korisnika vrši se kroz beskonačnu petlju u "Start.javi" i ona se nastavlja sve dok korisnik ne unese "exit" kako bi izašao iz programa.

4) Prilikom pokretanja programa utvrđeno je da program funkcioniše.

5) Program posjeduje MIT licencu što znači da je besplatan za korištenje.

* Statička analiza
* Razvojno okruženje IntelliJ IDEA je ukazalo na sljedeće greške prije samog pokretanja koda i to samo u “Calculator.javi”:

1) Fajl : Calculator.java- broj linije koda: 53- zapažanje: petlja ‘for’ se može zamijeniti sa naprednom ‘for’ petljom, odnosno for-each petljom

2) Fajl : Calculator.java- broj linije koda: 70 - zapažanje: promjenjljiva ‘textResult’ je suvišna

3) Fajl : Calculator.java- broj linije koda: 87 - zapažanje: Iz if bloka se može izdvojiti sljedeći dio: *Calculate*(numbers, operations); return;

jer ne utječe na samo izvršavanje uslova.

4) Fajl : Calculator.java- broj linije koda: 138 - zapažanje: Iz if bloka se može izdvojiti sljedeći dio: *Calculate*(numbers, operations); return;

jer ne utječe na samo izvršavanje uslova.

5) Fajl : Calculator.java- broj linije koda: 183 - zapažanje: ‘return’ je suvišan i treba se izbaciti

* U Visual Studio Codu korišten je Sonarlint i ovaj alat je ukazao na sljedeće nedostatke:

1) Fajl : Calculator.java - broj linije koda: 18 - zapažanje: Treba preimenovati metodu ‘ToString’ kako bi se izbjegla zamjena sa metodom ‘toString’ iz super klase "java.lang.Object". Sonarlint ovo kategoriše u veći Code smell.

2) Fajl : Calculator.java - broj linije koda: 24 - zapažanje: Treba preimenovati naziv metode ‘Run’ kako bi odgovarala regularnom izrazu.

3) Fajl : Calculator.java - broj linije koda: 70 - zapažanje: Nema potrebe dodjeljivati rezultat privremenoj varijabli "textResult". Ovo može smanjiti nepotrebno korištenje memorije i povećati čitljivost koda, posebno jer privremena varijabla "textResult" nije potrebna za daljnju obradu ili upotrebu u kodu. Ovo spada u Code smell.

4) Fajl : Calculator.java - broj linije koda: 74 - zapažanje: Treba preimenovati naziv metode ‘Calculate’ da odgovara regularnom izrazu.

5) Fajl : Calculator.java- broj linije koda: 183 - zapažanje: ‘return’ je suvišan i treba se izbaciti

6) Fajl : Start.java- broj linije koda: 8 i 19 - zapažanje: Treba zamijeniti upotrebu **System.out** ili **System.err** objekata u ovom kodu sa ‘logger’ koji je dio popularnih Java biblioteka za logiranje. Sonarlint ovo kategoriše u veći Code smell.

7) Fajl : Start.java - broj linije koda: 6- zapažanje: Treba preimenovati promjenjljivu ‘Expression’ da odgovara regularnom izrazu.