**Aplikacja**

**Używane technologie**

Do stworzenia aplikacji do Sudoku Solvera użyliśmy Java 8 oraz JavaFX 8. Do testowanie używaliśmy JUnit w wersji 5.4.2.

Zdecydowaliśmy się na wybór JavaFX w wersji 8 z kilku powodów, głównymi z nich są:

* wspiera CSS, użyteczny do stylizowania obiektów GUI
* wspiera wyrażenia lambda, które wykorzystaliśmy w kodzie, aby był on bardziej czytelny i przejrzysty
* istnieje możliwość łatwego i szybkiego przeniesnia aplikacji na system Android lub iOS, jeśli w przyszłości klient wyraźiłby taką chęć
* jest nowsza i częściej używana niż Swing

Zdecydowaliśmy się na wybór JUnit w tej wersji, ponieważ:

* zapewnia wsteczną kompatybilność z JUnit 4
* wspiera wyrażenia lambda, w przeciwieństwie do swojego poprzednika JUnit 4
* jest powszechnie używany w środowiku informatycznym

**Użyte wzorce projektowe**

* Singleton
  + Potrzebowaliśmy globalnie dostępnej zmiennej informującej nas o tym, czy w danym momencie oczekujemy na odpowiedź z serwara, żeby wiedzieć czy pozwolić użytkownikowi na wysłanie kolejnego zapytania na serwer, dlatego użyliśmy w kodzie wzorca singletona, który pozwolił nam na uzyskanie pożadanego efektu.
* Observer
* W aplikacji mamy kilka przypadków, kiedy po zmienie stanu w jednej klasie musimy poinformować o tym inne klasy i dokonać w ich zmian w odpowiedzi na tą pierwszą zmiane, więc wzorzec obserwatora jest tutaj idealny. Obserwator pozwala na przesyłanie informacji między obiektami bez żadnych zmian w klasie obserwatora i klasie ovserwowanej.