Niezawodność Oprogramowania

Sprawozdanie z zadania projektowego

Prowadzący: dr hab. inż. Kazimierz Worwa

Student: Maciej Rudnicki

Nr indeksu: 59955

Spis treści

[1 Wstęp 3](#_Toc24666366)

[1.1 Opis problemu i sformułowanie zadania 3](#_Toc24666367)

[2 Specyfikacja wymagań 3](#_Toc24666368)

[2.1 Wymagania niefunkcjonalne 3](#_Toc24666369)

[2.2 Wymagania funkcjonalne 3](#_Toc24666370)

[3 Specyfikacje projektowe 4](#_Toc24666371)

[4 Opis implementacji 4](#_Toc24666372)

[4.1 Wykorzystane technologie i narzędzia 4](#_Toc24666373)

[4.2 Wykorzystane techniki 5](#_Toc24666374)

[5 Wyniki 6](#_Toc24666375)

[5.1 Omówienie wyników badań 6](#_Toc24666376)

[5.2 Opis instalacji 6](#_Toc24666377)

# Wstęp

## Opis problemu i sformułowanie zadania

Zaprojektować i zaimplementować (w dowolnym języku i środowisku) aplikację programową, która dla zadanej dokładności obliczeń i wskazanego zbioru danych, zawierającego 240 odstępów czasowych pomiędzy wykryciem kolejnych błędów, umożliwia wyznaczenie wartości estymatorów parametrów N, Φ następujących modeli:

1. Jelińskiego-Morandy
2. Schicka-Wolvertona

Wykorzystując wyznaczone wartości parametrów N, Φ dla każdego z ww. modeli obliczyć wartość oczekiwaną czasu, jaki upłynie do momentu wykrycia kolejnego (241.) błędu.

# Specyfikacja wymagań

## Wymagania niefunkcjonalne

Wymagania niefunkcjonalne określają sprecyzowane cechy wytwarzanego systemu lub produktu oraz sposób, w jaki system powinien prezentować się pod względem jakościowym. Dla projektu laboratoryjnego zostały sprecyzowane następujące wymagania niefunkcjonalne:

1. Interfejs aplikacji zaimplementowany jako interfejs konsolowy
2. Aplikacja przeznaczona jest dla pojedynczego użytkownika
3. Aplikacja działa na systemie operacyjnym z zainstalowanym środowiskiem Java(TM) SE Runtime Environment
4. Aplikacja działa w sposób wydajny

## Wymagania funkcjonalne

Wymagania funkcjonalne określają wyniki zachowania się wytwarzanego systemu, który powinien zostać dostarczony przez jego funkcje. Dla projektu laboratoryjnego zostały sprecyzowane następujące wymagania funkcjonalne:

1. Aplikacja dla zadanej dokładności obliczeń i wskazanego zbioru danych umożliwia wyznaczenie wartości estymatorów N oraz Φ
2. Aplikacja na podstawie wyznaczonych wcześniej parametrów N i Φ może obliczyć wartość oczekiwaną czasu, jaki upłynie do momentu wykrycia kolejnego błędu
3. Do obliczania w/w wartości aplikacja korzysta z modeli Jelińskiego-Morandy oraz Shicka-Wolvertona

# Specyfikacje projektowe

Projekt stanowi program napisany w technologii Java 9 przy użyciu następujących bibliotek:

|  |  |  |
| --- | --- | --- |
| Nazwa pakietu / biblioteki | Wyróżnione podpakiety / podbiblioteki | Opis |
| Package java.io | - | Zapewnia komunikację wejście-wyjście w aplikacją przy pomocy tzw. streamów, serializacji oraz systemu plikowego |
| Package java.nio | java.nio.file.Files  java.nio.file.Paths | Biblioteka wspomagająca komunikację wejścia wyjścia, pozwalająca na intensywne czasowo i zasobowo operacje |
| Package java.util | Package java.util.logging | Biblioteka wspomagająca rejestrowanie działania programu oraz logowania postępów lub problemów podczas pracy |
| Package java.util.Arrays  Package java.util.LinkedList  Package java.util.List  Package java.util.Map  Package java.util.HashMap | Pakiety zawierające w sobie dedykowane struktury danych oraz ich implementacje |
| Package java.util.stream  Package java.util.stream.Collectors  Package java.util.stream.Stream | Pakiet pozwalający w łatwy sposób na przetwarzanie i obróbkę danych |

Tabela 3.1 Wypis bibliotek/pakietów zastosowanych w projekcie

# Opis implementacji

## Wykorzystane technologie i narzędzia

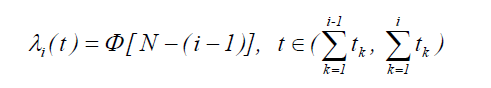
Przy implementacji aplikacji posłużono się następującymi technologiami oraz narzędziami:

|  |  |
| --- | --- |
| Nazwa | Opis |
| Intellij IDEA | Zintegrowane środowisko programistyczne dla Javy firmy JetBrains |
| Java 9 | Język programowania i platforma do tworzenia oprogramowania komputerowego (w wersji 9) |

Tabela 4.1 Wypis narzędzi i technologii zastosowanych w projekcie

## Wykorzystane techniki

Do implementacji programu oraz wyznaczenia wartości estymatorów parametrów N oraz Φ posłużono się wymaganymi modelami (pokazującymi poprawę niezawodności modułu lub systemu jako wynik poprawiania defektów w czasie). Przyjmując funkcję intensywności występujących błędów w postaci



można pokazać, że w oparciu o metodę największej wiarygodności estymatory MLE (Maximum Likelihood Estimation) parametrów N oraz Φ wyznaczyć można z zależności charakterystycznych dla danego modelu:

|  |  |
| --- | --- |
| Model Jelińskiego-Morandy | Oraz wartość oczekiwana czasu, jaki upłynie do momentu wykrycia kolejnego (w tym przypadku 241.) błędu: |
| Model Shicka-Wolvertona | Oraz wartość oczekiwana czasu, jaki upłynie do momentu wykrycia kolejnego (w tym przypadku 241.) błędu: |

# Wyniki

## Omówienie wyników badań

Badania zostały przeprowadzone na podstawie danych dostarczonych przez prowadzącego. Zawierają one informacje o odcinkach czasu (w minutach) pomiędzy wykryciem kolejnych błędów.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 1 | 772 | 822 | 903 | 1003 | 1067 | 960 | 1069 | 838 | 1055 | 906 | 1061 | 1241 | 1025 | 1001 | 962 |
| 2 | 1272 | 1322 | 1403 | 1503 | 1567 | 1460 | 1569 | 1338 | 1555 | 1406 | 1561 | 1741 | 1525 | 1501 | 1462 |
| 3 | 1472 | 1522 | 1603 | 1703 | 1767 | 1660 | 1769 | 1538 | 1755 | 1606 | 1761 | 1941 | 1725 | 1701 | 1662 |
| 4 | 1672 | 1722 | 1803 | 1903 | 1967 | 1860 | 1969 | 1738 | 1955 | 1806 | 1961 | 2141 | 1925 | 1901 | 1862 |
| 5 | 1772 | 1822 | 1903 | 2003 | 2067 | 1960 | 2069 | 1838 | 2055 | 1906 | 2061 | 2241 | 2025 | 2001 | 1962 |
| 6 | 1872 | 1922 | 2003 | 2103 | 2167 | 2060 | 2169 | 1938 | 2155 | 2006 | 2161 | 2341 | 2125 | 2101 | 2062 |
| 7 | 1972 | 2022 | 2103 | 2203 | 2267 | 2160 | 2269 | 2038 | 2255 | 2106 | 2261 | 2441 | 2225 | 2201 | 2162 |
| 8 | 2072 | 2122 | 2203 | 2303 | 2367 | 2260 | 2369 | 2138 | 2355 | 2206 | 2361 | 2541 | 2325 | 2301 | 2262 |
| 9 | 2172 | 2222 | 2303 | 2403 | 2467 | 2360 | 2469 | 2238 | 2455 | 2306 | 2461 | 2641 | 2425 | 2401 | 2362 |
| 10 | 2272 | 2322 | 2403 | 2503 | 2567 | 2460 | 2569 | 2338 | 2555 | 2406 | 2561 | 2741 | 2525 | 2501 | 2462 |
| 11 | 2372 | 2422 | 2503 | 2603 | 2667 | 2560 | 2669 | 2438 | 2655 | 2506 | 2661 | 2841 | 2625 | 2601 | 2562 |
| 12 | 2472 | 2522 | 2603 | 2703 | 2767 | 2660 | 2769 | 2538 | 2755 | 2606 | 2761 | 2941 | 2725 | 2701 | 2662 |
| 13 | 2572 | 2622 | 2703 | 2803 | 2867 | 2760 | 2869 | 2638 | 2855 | 2706 | 2861 | 3041 | 2825 | 2801 | 2762 |
| 14 | 2672 | 2722 | 2803 | 2903 | 2967 | 2860 | 2969 | 2738 | 2955 | 2806 | 2961 | 3141 | 2925 | 2901 | 2862 |
| 15 | 2772 | 2822 | 2903 | 3003 | 3067 | 2960 | 3069 | 2838 | 3055 | 2906 | 3061 | 3241 | 3025 | 3001 | 2962 |
| 16 | 2972 | 3022 | 3103 | 3203 | 3267 | 3160 | 3269 | 3038 | 3255 | 3106 | 3261 | 3441 | 3225 | 3201 | 3162 |

Tabela 6.1 Zbiór danych o czasach wykrycia kolejnych błędów

## Opis instalacji

Aplikacja oparta jest na technologii Java w wersji 9, dlatego też jej instalacja (lub wersje następne, ponieważ kompatybilność wsteczna jest wspierana) jest wymagana do poprawnego uruchomienia skryptu. Program jako źródło prawdy przyjmuje plik config.properties, czyli plik konfiguracyjny (w przykładzie jest to config.properties), który musi zawierać dwie podstawowe wartości:

* Wzlędną ścieżkę dostępu do pliku z danymi wejściowymi (plik z roszerzeniem .txt musi posiadać informacje o kolejnych czasach jako liczby stałoprzecinkowe oddzielone od siebie białym znakiem, w dowolnej konfiguracji liczbowej, zarówno wierszowej, jak i kolumnowej)
* wartość różnicy między obliczonymi parametrami N (po obu stronach równania) przy wyliczaniu zależności dla estymatorów MLE (Maximum Likelihood Estimation).

Podstawowa wersja pliku konfiguracyjnego:

data-source=data.txt  
accepted-difference=0.001

Przy uwzględnieniu wszystkich przesłanek program należy uruchomić następującą komendą:

java -jar Niezawodnosc\_Oprogramowania\_Rudnicki.jar