Raport z testowania.

# 1. Co to za program/biblioteka jest przetestowana, do czego służy?

Jest to biblioteka z wieloma rodzajami algorytmów, działających na liczbach, stringach i tablicach.

# 2. Co jest planowane do przetestowania?

W planach mamy zamiar przetestować następujące paczki:

* TriangleFunc (9 JUNIT + analiza kodu)
* SumZero (analiza kodu, 3 testy białoskrzynkowe)
* StringOps (26 JUNIT)
* PermutationsStr (6 testow bialoskrzynkowych)
* maxDist (4 białoskrzynkowe, 5 JUNIT)
* MathOps (36 JUNIT, 5 białoskrzynkowych)
* longPalindromicString (5 białoskrzynkowych)
* isNumber (16 JUNIT)
* intArrayOps (17 JUNIT)
* allSubsets (2 białoskrzynkowe)
* ActivityZero (4 JUNIT, 3 białoskrzynkowe)

# 3. Wymagania środowiskowe i programowe do uruchomienia programu i testów

Wystarczy Eclipse. Ja testy pisałem na Intellij.

# 4. Instrukcja jak zbudować + uruchomić projekt i testy

Jeśli w jakiejś paczce zostały zbudowane testy jednostkowe to trzeba wejść do tej paczki i kliknąć run tests.

# 5. Ryzyko/problemy związane z testowaniem tego programu/biblioteki i dlaczego? Jeśli nie ma to dlaczego?

Problemem był prawie całkowity brak dokumentacji i nieczytelny kod.

# 6. Co zostało przetestowane i dlaczego, co nie zostało przetestowane i dlaczego. Wykryte błędy (jeśli wykryto), sytuacje w których występują błędy.

Zostały przetestowane paczki wymienione w punkcie 2. Ciężko było przetestować coś więcej, gdyż w większości algorytmów nie wiadomo o co chodzi – co mają robić i dlaczego.

Wykryte błędy

|  |  |
| --- | --- |
| Paczka | Błąd |
| TriangleFunc | * Brak walidacji zakresu bokow trojkata (3 x bład) – łatwo przekroczyć zakres int * Trójkąt równoboczny jest trójkątem równoramiennym – program tego nie pokazuje |
| SumZero | W dużej części przypadków zwraca częściowe wyniki zamiast pełnych |
| StringOps | * Brak walidacji inputu przy metodzie GenerateWordsFromPhone() * Błąd na pustym zbiorze w LRS() * Błędny wynik w LRS() * Błąd na pustym zbiorze w LRS1() * Błędny wynik w LRS1() * Błędny wynik dla LCS1() * Wypuszczanie poza zakres dla reverse() * Błąd na zbiorze pustym dla LRS * Błędny wynik dla IsRexMatch() * Błędny wynik dla IsRexMatch1() |
| PermutationsStr | Brak walidacji inputu (przyjmuje liczby ujemne) |
| maxDist | * Błąd dla tablicy jednoelementowej * Brak walidacji inputu (przyjmuje stringi) * Kluczowa funkcja ustawiona na private (zmienione do testów) * Błąd dla przykładowej tablicy w JUNIT |
| MathOps | * Przekracza zakres int w działaniach * Błąd w pow(), 0^0 jest symbolem nieoznaczonym * Przyjmuje ujemne liczby dla 0 w potegowaniu * Wiesza sqrt dla liczb < 0 * Dopuszcza dzielenie przez 0 w divide1() * Zakres wiesza divide() * Dzielenie maksymalnych zakresów daje błędny wynik |
| longPalindromicString | Błąd dla pustego stringa |
| isNumber | Błąd dla używania w potędze notacji z e, a to przecież również liczba |
| intArrayOps | * Błąd na pustych tablicach w FindMedianSortedArrays() * Błąd na pustych tablicach w FindMedianSortedArrays1() * Błąd na pustych tablicach w FindK1() * Brak walidacji n w FirstMissingPositive * Błąd na pustej tablicy w LIS() |
| allSubsets | Powtarzające się elementy powinny być usunięte |
| ActivityZero | * Brak walidacji inputu (można wprowadzić “asdasd”,”asdasd” |

# 7. Stosowane techniki testowania i dlaczego?

Testowanie biało oraz czarnoskrzynkowe (tam gdzie na przykład były voidy albo główny algorytm był w mainie, albo nie było zrozumiałe jak działa algorytm tylko było wiadomo jakie ma dawać wyniki) oraz JUNIT wszędzie indziej.

# 8. Metryki podsumowujące testowanie:

* liczba testów: 141
* zaprojektowanych: 141
* wykonanych: 141
* zdanych: 86
* niezdanych: 55
* liczba znalezionych defektow: 55,
* czas testowania: 5 dni,
* efektywnosc testow (#defektow / # testow): 40%