# TDA.JAVA

Spis Treści

[TDA.JAVA 1](#_Toc492214742)

[Wprowadzenie 1](#_Toc492214743)

[Wymagania systemowe 1](#_Toc492214744)

[Opis techniczny 1](#_Toc492214745)

[Instrukcja użytkowania 2](#_Toc492214746)

## Wprowadzenie

Program TDA.JAVA jest aplikacją konsolową służącą do generowania diagramów persystencji na podstawie chmury punktów za pomocą Topologicznej Analizy Danych.   
Program na swoim wyjściu z podanej kolekcji współrzędnych generuje diagramy persystencji (ich ilość uzależniona jest od ilości wymiarów), interwały w formie tekstowej oraz entropię każdego z diagramów.

## Wymagania systemowe

* System operacyjny: Windows 10, Windows 8.1, Windows 7
* Zainstalowane JRE lub JDK w wersji odpowiednio 8 lub 1.8.
* 4 GB lub więcej pamięci operacyjnej RAM

## Opis techniczny

Program TDA.JAVA to aplikacja konsolowa napisana w języku JAVA wykorzystująca system budowania MAVEN. Program ma na celu umożliwienie wywoływania funkcji biblioteki javaplex implementującej algorytmy TDA oraz rozszerzenie jej o generowanie entropii diagramów w danym wymiarze. Większość plików generowanych przez TDA.JAVA jest w formacie JSON oraz PNG (dla wykresów). Aplikacja posiada prosty moduł logujący wykonywane działania i błędy możliwy do konfiguracji. Wynikowe nazwy plików są z góry ustalone, użytkownik ma możliwość podania folderu, do którego wyniki pracy aplikacji powinny zostać zapisane.

## Instrukcja użytkowania

* Budowanie aplikacji ze źródeł
  + Do budowania aplikacji używany jest MAVEN za pomocą którego można stworzyć plik JAR gotowy do uruchomienia na maszynie wirtualnej JAVA.   
    Poniższe polecenie utworzy w katalogu /dist archiwum ZIP w której znajdować się będą wszystkie zależności oraz wynikowy plik JAR.   
    *mvn clean compile package assembly:single -e*
* Uruchomienie aplikacji odbywa się poprzez wydanie komendy java -jar <nazwa\_jar>.   
  Przykładowe wywołanie:  
  *java -jar C:\tda.java\tdathesis-1.0-SNAPSHOT.jar "C:\program\_data\points\_cloud.json" 2 118 True 200 "C:\program\_data\_output"*
* Parametry programu:  
  *tdathesis-1.0-SNAPSHOT.jar <input\_file\_name> <max\_dimension> <max\_filtration> <use\_landmarks> <max\_landmarks> <output\_folder>*
  + *<input\_file\_name>* - plik wejściowy w formacie JSON zawierający kolekcję punktów
  + *<max\_dimension>* -liczba całkowita oznaczająca maksymalną ilość wymiarów która będzie procesowana. Duża ilość wymiarów powoduje mocne obciążenie pamięci RAM.
  + *<max\_filtration>* - maksymalna wartość parametru p (średnicy okręgu) w ramach którego tworzone są kompleksy symplicjalne.
  + *<use\_landmarks>* - ustawienie to pozwala zastąpić dużą ilość punktów ich reprezentantami redukując znacząco ilość mocy obliczeniowej potrzebnej do liczenia diagramów, a zachowując przy tym właściwości topologiczne.
  + *<max\_landmarks>* - liczba całkowita która oznacza jaką ilością punktów zastępczych zostaną zastąpione oryginalne współrzędne.
  + *<output\_folder>* - folder do którego program zapisze wyniki swojej pracy.
* Wyjście programu zapisane w folderze *<output\_folder>*:
  + *Barcodes\_<numer\_wymiaru>.png* gdzie <numer\_wymiaru> odpowiada numerowi wymiaru diagramu persystencji
  + *persistent\_homology\_summary.txt* – interwały dla każdego z wymiarów w formie tekstowej wraz z liczbami Betti.
  + *entropy.json –* tablica JSON z wartością entropi dla każdego z wygenerowanych wykresów persystencji.
* Dokładna dokumentacja techniczna wraz z kodem dostępna jest w formie javadoc w archiwum **tda.java.documentation.zip**.