

# SYMULATOR WNUM

## Dokumentacja do projektu z PROO

Artur Skrzypczak  
Illia Kovalenko

### Cel projektu

Celem projektu stworzono symulator obrazujący działanie różnych metod interpolacji i aproksymacji, posiadający środowisko graficzne oraz możliwość podawania parametrów przez użytkownika.

### Informacje podstawowe

Program napisany został w języku Java. Wymaga dodatkowych, niestandardowych bibliotek, które zostały załączone w odpowiednich folderach. Są to:

- org.jfree.chart.ChartFactory – biblioteka umożliwiająca tworzenie wykresów
- org.jfree.chart.ChartPanel – biblioteka umożliwiająca wyświetlanie wykresów w panelu Swing
- org.jfree.chart.JFreeChart – biblioteka umożliwiająca tworzenie bardziej zaawansowanych wykresów
- org.jfree.chart.plot.PlotOrientation – służy do ustalania orientacji wykresu
- org.nfunk.jep.JEP – umożliwia obliczenia matematyczne funkcji na podstawie wyrażeń podanych przez użytkownika

Program kompiluje się z poziomu terminala poleceniem:

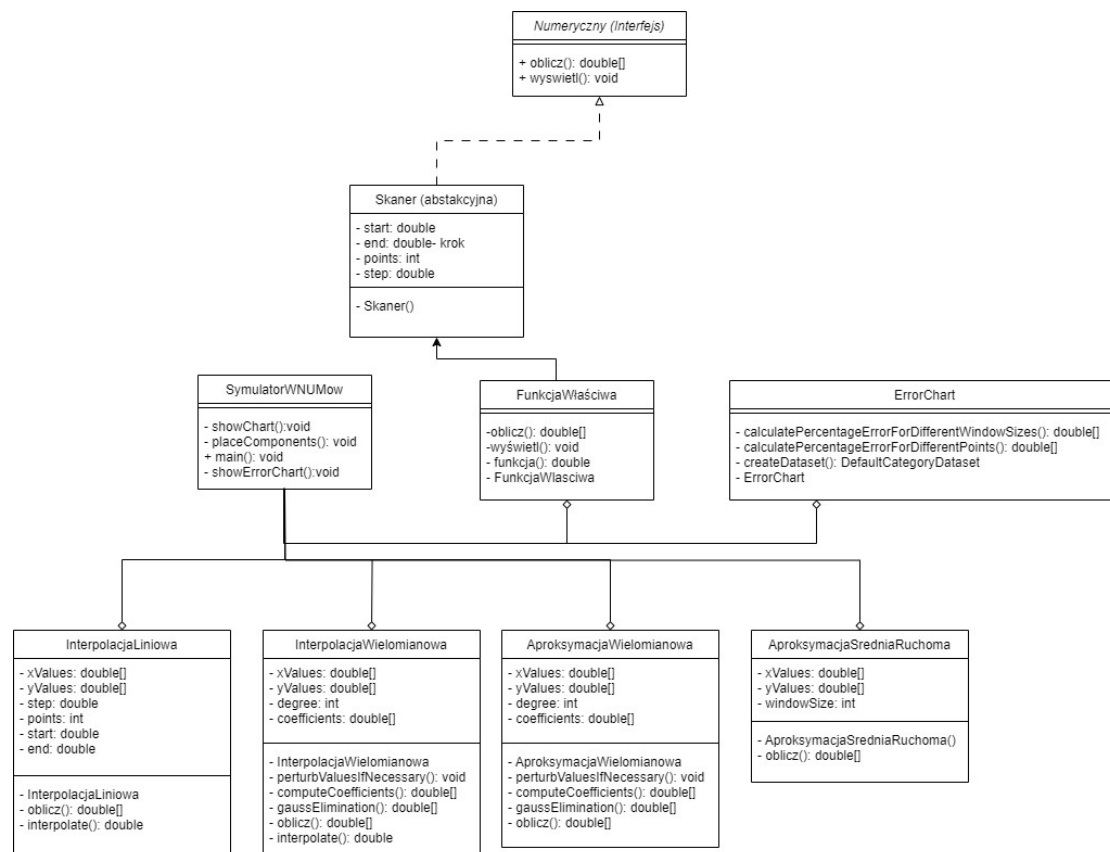
```
javac -cp "lib/commons-math3-3.6.1.jar:lib/jfreechart-1.5.3.jar:lib/jcommon-1.0.24.jar:lib/jep-java-4.0-trial.jar" src/main/java/com/example/*.java src/main/java/com/example/interpolacja/*.java src/main/java/com/example/aproksymacja/*.java
```

Natomiast uruchamiany jest, również z poziomu terminala, poleceniem:

```
java -cp "src/main/java:lib/commons-math3-3.6.1.jar:lib/jfreechart-1.5.3.jar:lib/jcommon-1.0.24.jar:lib/jep-java-4.0-trial.jar" com.example.SymulatorWNUMow
```

W celu dokonania interpolacji lub aproksymacji należy wpisać punkt początkowy, punkt końcowy oraz liczbę punktów (zostaną one rozmieszczone równoodległe), a także wpisać funkcję, która nas interesuje (format zapisu: np.  $e^x$  lub  $\sin(x)$ ), a także wpisać liczbę naturalną będącą stopniem wielomianu interpolującego/aproksymującego (dotyczy to metod wielomianowych). Po wpisaniu wszystkich danych należy kliknąć w przycisk „Pokaż Wykres” lub „Pokaż Wykres Błędu”.

### Diagram klas



## Opis poszczególnych klas

- **SymulatorWNUMow**

Główna klasa aplikacji odpowiedzialna za interakcję z użytkownikiem, wyświetlanie okna aplikacji i wywoływanie odpowiednich metod obliczeniowych i wizualizacyjnych.

Metody:

- `main()`: Uruchamia aplikację, tworząc okno główne.
- `placeComponents()`: Dodaje komponenty interfejsu użytkownika do panelu.
- `showChart()`: Tworzy i wyświetla wykresy na podstawie danych wejściowych.
- `showErrorChart()`: Tworzy i wyświetla wykresy błędów na podstawie danych wejściowych.

- **ErrorChart**

Klasa odpowiedzialna za tworzenie okna z wykresem błędów. Korzysta z biblioteki JFreeChart do tworzenia i wyświetlania wykresów.

Metody:

- `createDataset()`: Tworzy zbiór danych dla wykresu.
- `calculatePercentageErrorForDifferentPoints()`: Oblicza procentowy błąd dla danej liczby punktów.
- `calculatePercentageErrorForDifferentWindowSizes()`: Oblicza procentowy błąd dla różnych rozmiarów okna średniej ruchomej.

- **FunkcjaWlasciwa**

Klasa dziedzicząca po Skaner, implementująca konkretne obliczenia funkcji matematycznej.

Metody:

- oblicz(): Oblicza wartości funkcji w określonych punktach.
- funkcja(): Przykładowa funkcja matematyczna (sinus).
- wyswietl(): Metoda odpowiedzialna za wyświetlanie wyników (do implementacji).

- Numeryczny

Interfejs definiujący metody do obliczania i wyświetlania wyników.

Metody:

- oblicz(): Metoda do obliczania wartości.
- wyswietl(): Metoda do wyświetlania wyników.

- Skaner

Abstrakcyjna klasa implementująca interfejs Numeryczny, zawierająca podstawowe pola i konstruktor dla klas dziedziczących.

- AproksymacjaSredniaRuchoma

Klasa implementująca aproksymację średnią ruchomą.

Metody:

- oblicz(): Oblicza wartości aproksymacji średnią ruchomą.

- AproksymacjaWielomianowa

Klasa implementująca aproksymację wielomianową.

Metody:

- perturbValuesIfNecessary(): Perturbacja wartości x w celu uniknięcia problemów numerycznych.
- computeCoefficients(): Oblicza współczynniki wielomianu.
- gaussElimination(): Rozwiązuje układ równań metodą eliminacji Gaussa.
- oblicz(): Oblicza wartości aproksymacji dla podanych wartości x.

- InterpolacjaLiniowa

Klasa implementująca interpolację liniową.

Metody:

- oblicz(): Oblicza wartości interpolacji dla podanych punktów.
- interpolate(double x, double[] xs, double[] ys): Statyczna metoda wykonująca interpolację liniową.

- InterpolacjaWielomianowa

Klasa implementująca interpolację wielomianową.

Metody:

- perturbValuesIfNecessary(): Perturbacja wartości x w celu uniknięcia problemów numerycznych.
- computeCoefficients(): Oblicza współczynniki wielomianu.
- gaussElimination(): Rozwiązuje układ równań metodą eliminacji Gaussa.
- oblicz(): Oblicza wartości interpolacji dla podanych punktów.
- interpolate(): Oblicza wartość interpolacji dla pojedynczego punktu x.

**Lista funkcjonalności:**

- wprowadzanie przez użytkownika parametrów w postaci przedziału, liczby punktów, badanej funkcji
- możliwość wyboru metody interpolacji i aproksymacji
- możliwość wyświetlania w postaci wykresu wyniku interpolacji lub aproksymacji
- możliwość wyświetlenia wykresu błędu interpolacji

### Wygląd GUI

Symulator WNUMów@S10-ProLiant

Początek przedziału: 1

Koniec przedziału: 10

Liczba punktów: 10

Funkcja:  $e^x$

Stopień wielomianu: 2

☒ Funkcja Właściwa

☒ Interpolacja Lin...

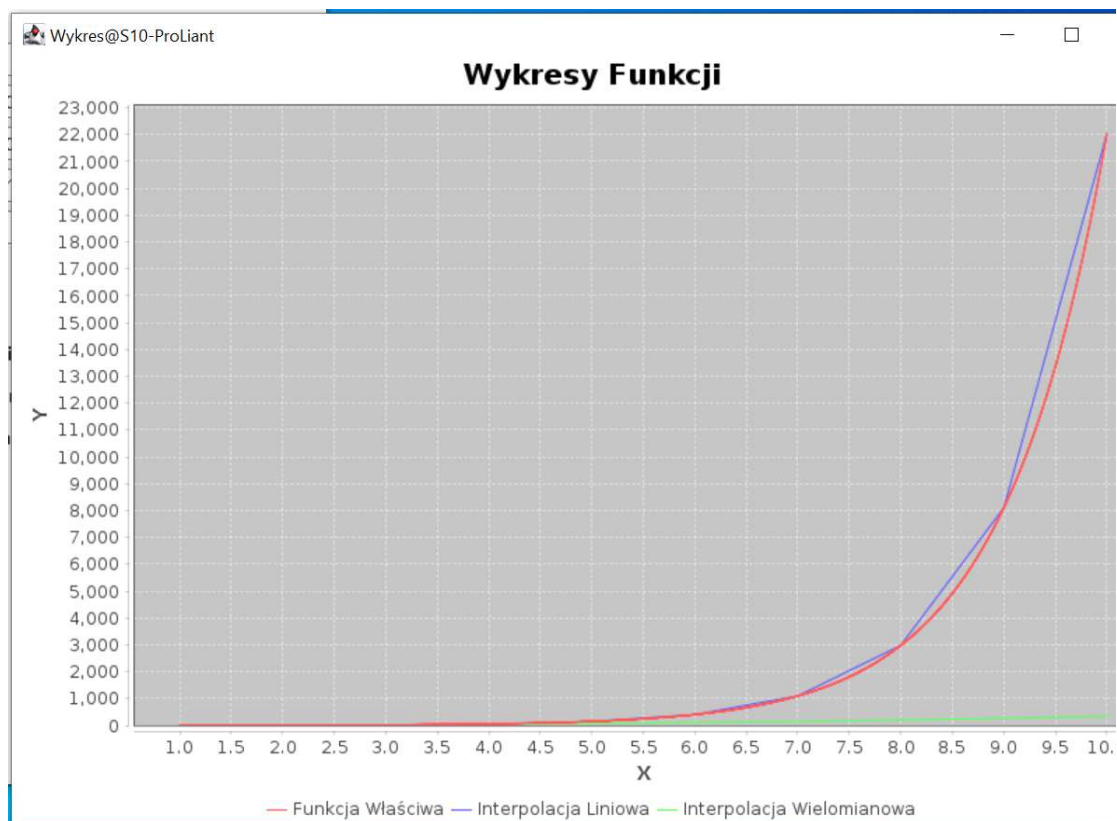
☒ Interpolacja Wielomia...

☐ Aproksymacja Wielomianowa

☐ Aproksymacja Średnią Ruchomą

Pokaż Wykres      Pokaż Wykres...

Okno wyboru parametrów i metod



Okno wykresów metod