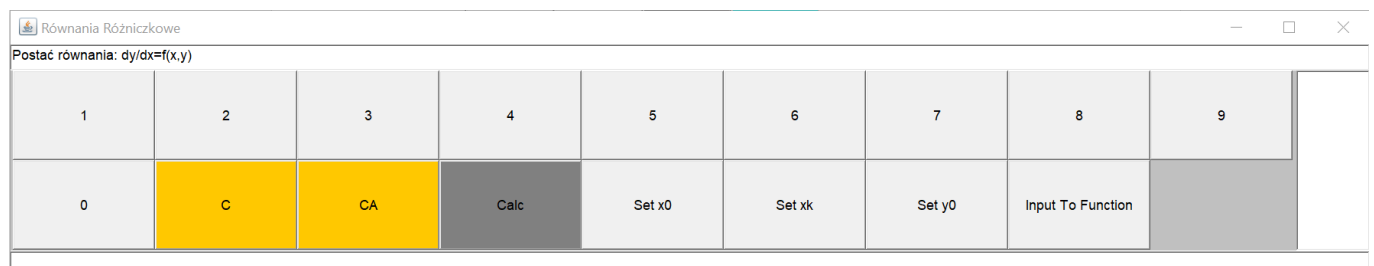


Kalkulator Graficzny Napisany w Języku Java – Raport

Projekt jest realizowany poprzez dwie klasy, z czego klasa **Kalkulator**, zawierająca metodę *main* odpowiada za całą funkcjonalność standardowego kalkulatora naukowego. Druga natomiast klasa **Differentials** odpowiada za funkcjonowanie osobnego okienka odpowiadającego za rozwiązywanie równań różniczkowych.

Postać rozwiniętej aplikacji okienkowej:



Okienko do rozwiązywania równań różniczkowych wywoływane jest za pomocą przycisku **d/dx**, którego funkcjonowanie w kodzie zdefiniowane jest następująco:

```
if(e.getSource()==Bdif) {  
    Differentials Differentials = new Differentials();  
}
```

Elementy interfejsu użytkownika zostały wykonane za pomocą biblioteki AWT, natomiast jego interaktywność poprzez bibliotekę Swing.

Poszczególne pola tekstowe w kalkulatorze przechowują zmienne typu String, co pozwala na dynamiczne łączenie poszczególnych inputów użytkownika, a następnie zamienianie ich na wartości liczbowe następująco:

```
n=Float.parseFloat(s1) + Float.parseFloat(s2);  
poletekst.setText(String.valueOf(df.format(n)));
```

Działanie odpowiednich przycisków w kalkulatorze naukowym jest oczywiste względem ich wizualnego opisu, gdzie warto mieć na uwadze iż pierwszy input (czyli dokonany przed naciśnięciem przycisku pożądanej operacji) jest oznaczony jako Y, a drugi input (dokonany po naciśnięciu przycisku operacji) jest oznaczony jako X.

Istotnym jest także wspomnieć, że w wyniku braków w dostępnych bibliotekach operacja logarytmu nie funkcjonuje w zadawalającym zakresie zaokrąglenia i daje wyniki niemiarodajne dla logarytmów o bazie innej niż 10.

Aby zaimplementować operację logarytmu o dowolnej podstawie skorzystałem z zależności matematycznej opisanej w kodzie następująco:

```
n=(Math.log10(Double.parseDouble(s1)/Math.log10(Double.parseDouble(s2))));  
poletekst.setText(String.valueOf(df.format(n)));
```

W wyniku zaokrągleń jakie występują podczas tej operacji metoda staje się niemiarodajna.

Podczas użytkowania kalkulatora do równań różniczkowych należy wprowadzać input do pola tekstowego z prawej strony menu.

Po wprowadzeniu pożądanego wejścia należy ustawić początkowe wartości $x(0)$, $y(0)$ oraz x_k , ostatnie z których jest wartością x , dla której pragniemy znaleźć wartość y z podanego równania. Aby to zrobić używamy odpowiednio oznaczonych przycisków.

W dolnym polu tekstowym wyświetlane są po kolei części definiowanej funkcji wedle oznaczenia z górnego pola tekstowego, dodawane za pomocą wpisania ich do prawego pola tekstowego i wciśnięcia przycisku *Input To Function*.

Przycisk **C** zeruje prawe pole tekstowe służące do wprowadzania danych, natomiast przycisk **CA** czyści także pole wyświetlające zadaną funkcję.

Właściwe rozwiązywanie równania różniczkowego dokonywane jest za pomocą podstawowych metod numerycznych, takich jak metoda Eulera:

```
dt=(xk-x[0])/N;  
  
for (int i = 0; i < N; i++)  
{  
    y[i+1]=y[i]+f[i]*dt;  
    yk=y[i+1];  
}
```

gdzie dt jest naszym krokiem obliczeniowym uzależnionym od stałej wartości N , którą na potrzeby szybkości kalkulacji przyjąłem roboczo jako 100.

Możemy również zaimplementować bardziej wydajną metodę Rungego-Kutty IV, która przyjmuje następującą postać (pseudokod):

```
for (int i = 0; i < N; i++)  
{  
    k1=dt*f[xi,yi];  
    k2=dt*f[xi+dt/2,yi+k1/2];  
    k3=dt*f[xi+dt/2,yi+k2/2];  
    k4=dt*f[xi+dt,yi+k3];  
    y[i+1]=y[i]+(1/6)*(k1+2*k2+2*k3+k4)*dt;  
}
```
