

# **Zadanie nr 1 - Generacja sygnału i szumu**

## **Cyfrowe Przetwarzanie Sygnałów**

Arkadiusz Kowalczyk, 204207

Damian Omiotek, 204222

Paweł Znamiec, 204247

26 maja 2018

# 1 Cel zadania

Celem zadania było utworzenie aplikacji graficznej umożliwiającej generowanie sygnałów cyfrowych, takich jak:

- szum o rozkładzie jednostajnym,
- sygnał sinusoidalny,
- sygnał sinusoidalny wyprostowany jednopółkowo
- sygnał sinusoidalny wyprostowany dwupółkowo
- sygnał prostokątny
- sygnał prostokątny symetryczny
- sygnał trójkątny
- skok jednostkowy
- impuls jednostkowy
- szum impulsowy

Należało również umożliwić wykonywanie prostych operacji arytmetycznych na wymienionych powyżej sygnałach. Do operacji należy:

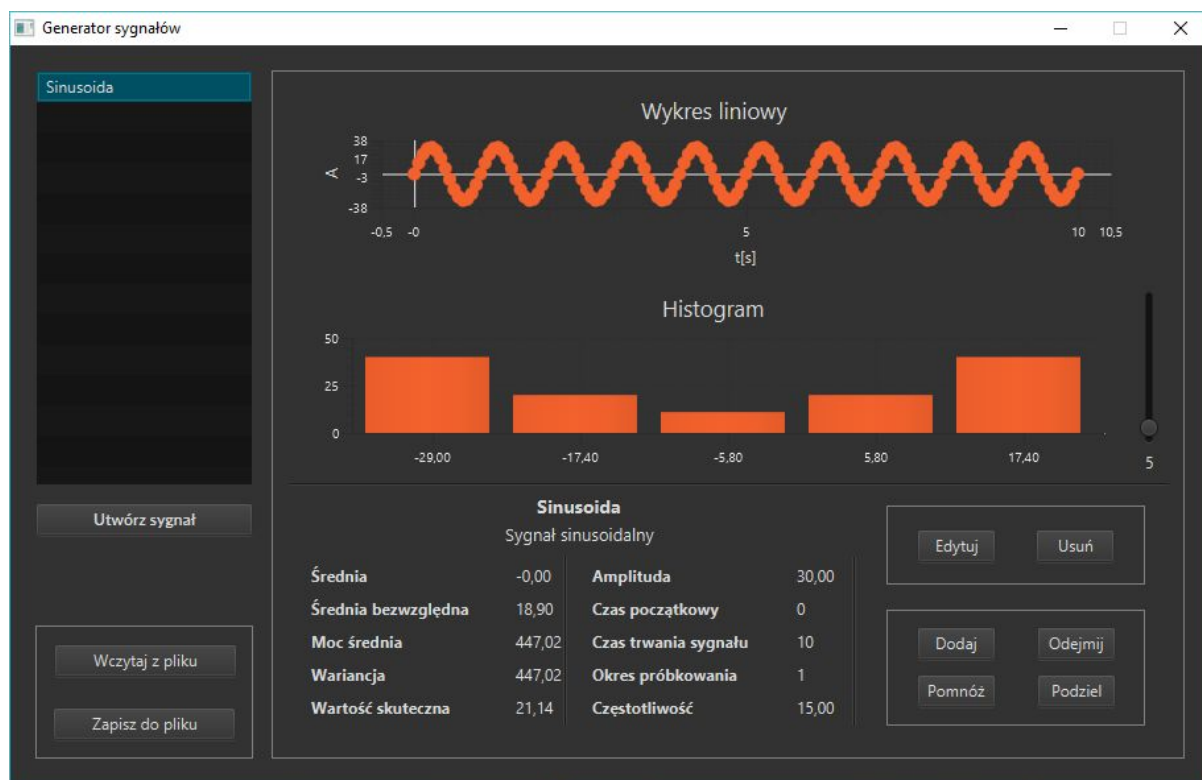
- dodawanie
- odejmowanie
- mnożenie
- dzielenie

Sygnały miały zostać przedstawione w formie graficznej - na wykresie liniowym i histogramie, oraz miała zostać zaimplementowana możliwość zapisywania sygnałów do pliku oraz ich odczyt.

## 2 Obsługa programu

### 2.1 Panel główny

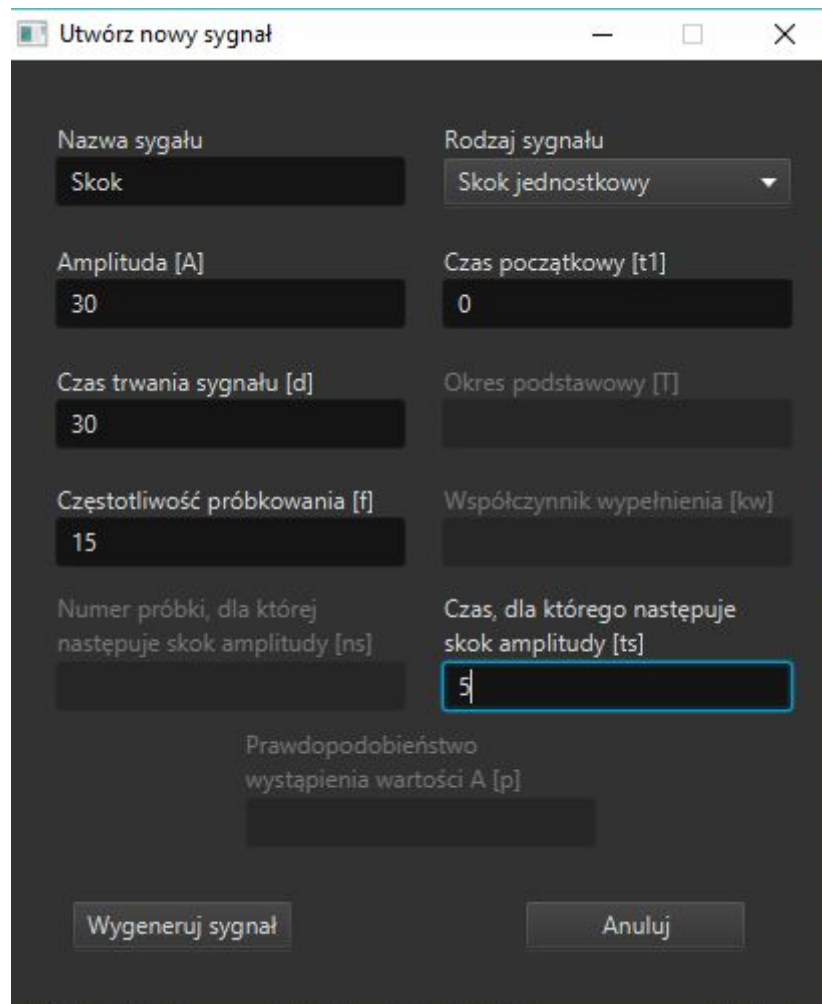
Program składa z okna głównego, w którym jest możliwość podglądu listy sygnałów, ich tworzenia, edycji, usuwania, zapisu i odczytu do / z pliku, operacje arytmetyczne oraz podgląd na podstawowe informacje o sygnale, jego statystyki oraz histogram i wykres liniowy.



Panel główny

## 2.2 Panel tworzenia sygnału

Panel tworzenia sygnału pozwala na wybór jednego z rodzaju sygnału oraz wpisanie odpowiednich danych, które są niezbędne do jego wygenerowania. Aktywność pól jest uzależniona od wybranego typu sygnału. Po wpisaniu znaku w jakiekolwiek pole, uruchamia się dla niego walidacja wskazująca, czy użytkownik wpisuje poprawne dane. Również po kliknięciu przycisku “Wygeneruj sygnał” sprawdzane są wszystkie wymagane pola pod względem ich poprawności oraz tego, czy są wypełnione.



Utwórz nowy sygnał

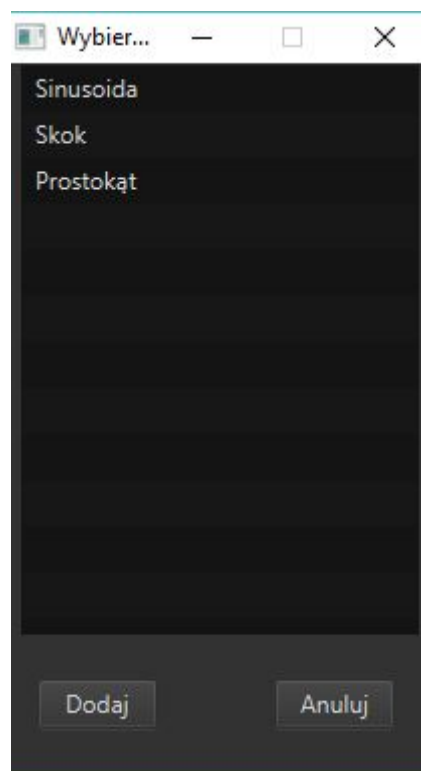
Nazwa sygnału	Rodzaj sygnału
Skok	Skok jednostkowy
Amplituda [A]	Czas początkowy [t1]
30	0
Czas trwania sygnału [d]	Okres podstawowy [T]
30	
Częstotliwość próbkowania [f]	Współczynnik wypełnienia [kw]
15	
Numer próbki, dla której następuje skok amplitudy [ns]	Czas, dla którego następuje skok amplitudy [ts]
	5
Prawdopodobieństwo wystąpienia wartości A [p]	

Wygeneruj sygnał      Anuluj

Panel tworzenia sygnału

## 2.3 Panel wyboru sygnału w operacji arytmetycznej

Panel uruchamia się po naciśnięciu któregoś z przycisków: Dodaj, Odejmij, Pomnóż, Podziel. Pozwala na wybór drugiego sygnału, z którym zostanie wykonana operacja arytmetyczna. Lista zawiera wyłącznie sygnały, które mają taką samą częstotliwość próbkowania, jak pierwotny sygnał. Nowy sygnał będzie miał nazwę w formacie *Sygnał pierwszy + Sygnał drugi*



Panel dodawania sygnałów

## 2.4 Zapis i odczyt z pliku

Sygnaly są zapisywane do pliku w formacie JSON, który pozwala na wygodną edycję danych z poziomu pliku. Przy odczycie do programu dane są odpowiednio deserializowane do obiektu.

Przykładowy plik JSON z danymi o sygnale wygląda następująco:

```
{
  "signalType": "Sygnał sinusoidalny",
  "name": "Sinusoida",
  "amplitude": 30.0,
  "startTime": 0,
  "duration": 5,
  "frequencySampling": 5.0,
  "baseInterval": 1,
  "signal": {
    "0.0": 0.0,
    "0.2": 28.531695488854606,
    "0.4": 17.633557568774197,
    "0.6": -17.63355756877419,
    "0.8": -28.53169548885461,
    "1.0": -7.34788079488412E-15,
    "1.2": 28.531695488854606,
    "1.4": 17.6335575687742,
    "1.6": -17.633557568774183,
    "1.8": -28.531695488854613,
    "2.0": -1.469576158976824E-14,
    "2.2": 28.53169548885462,
    "2.4": 17.633557568774208,
    "2.6": -17.633557568774176,
    "2.8": -28.531695488854613,
    "3.0": -2.2043642384652355E-14,
    "3.2": 28.5316954888546,
    "3.4": 17.633557568774215,
    "3.6": -17.633557568774172,
    "3.8": -28.531695488854616,
    "4.0": -2.939152317953648E-14,
    "4.2": 28.531695488854595,
    "4.4": 17.633557568774133,
    "4.6": -17.63355756877408,
    "4.8": -28.53169548885462
  },
  "avg": 1.4210854715202005E-15,
  "absoluteAvg": 18.466101223051517,
  "avgSignalPower": 449.9999999999998,
  "variance": 449.9999999999998,
  "rms": 21.21320343559642
}
```



Wykres dla danych z pliku JSON

### 3 Opis implementacji

Program został napisany przy użyciu języka obiektowego *Java*, a GUI zostało stworzone przy pomocy oficjalnego frameworka graficznego *JavaFX*. Framework ten zawierał wszelkie niezbędne narzędzia do prezentacji danych użytych w programie, łącznie z wykresami. Do serializacji oraz deserializacji danych do / z pliku w formacie JSON zastosowano bibliotekę o nazwie *gson*.

W klasie *Main* aplikacji tworzony jest kontroler (*MainAppController*) oraz ładowany widok (*MainApp.fxml*) głównego okna. Po kliknięciu przycisku "Utwórz sygnał" ładowany jest kontroler z oddzielnym widokiem (*SignalDetailsWindow*), który odpowiada za tworzenie oraz walidację nowego sygnału. Stamtąd tworzony jest obiekt wybranego sygnału. Implementacja generatora każdego sygnału znajduje się w oddzielnych klasach, które naturalnie dziedziczą z klasy *Signal*, a następnie od pochodnych, np. *RectangularSymmetric* od *Rectangular*. W projekcie znajduje się również klasa *Utils*, w której znajdują się zmienne statyczne dla projektu, generacja wykresów oraz statyczne operacje arytmetyczne dla sygnałów.