

# Testy dokumentacji

Stan na dzień: 12.06.2020r

Kierownik grupy testowej: Mateusz Kacprzyk

Testerzy: Mateusz Gajewski, Paulina Gnas, Adam Kowalczyk, Dawid Skolimowski,  
Karolina Sobolewska, Zhenya Suharevich, Daniel Szot, Mateusz Żelaszczyk

## Spis treści

Baza_danych.....	2
BazaDanychPanel.....	3
Complex.....	4
DTFT.....	5
F0Panel.....	6
FFT.....	7
funkcje.....	8
LittleBigEndian.....	9
MainWindow.....	10
NagłówekPlikuWave.....	11
Nagranie.....	12
Nagrywarka.....	13
PlikWave.....	14
Rysowanie.java – PanelOkna.....	15
Rysowanie.java – Ramka.....	16
Rysowanie.java – Rysowanie.....	17
widmo.....	18
WidmoPanel.....	19
Wykres.....	20
WykresWidma.....	21

# Baza\_danych

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa zapisuje listę nagrań do pliku nagraniaData.dat

- ♦ **Opis metod**

- x *Baza\_danych()* - Konstruktor bazy danych jest poprawnie opisany.
- x *PobierzListe()* - Metoda jest poprawnie opisana.
- x *SetNagrania()* - Metoda jest poprawnie opisana. Dzięki komentarzom wiemy co metoda pobiera oraz co zwraca.
- x *DodajNagranie()* - Metoda jest poprawnie opisana.
- x *Nagrania()* - Metoda jest poprawnie opisana.

- ♦ **Uwagi**

Część komentarzy jest w języku polskim, a część w języku angielskim, co nie wygląda estetycznie. Przed klasą powinien być także krótki opis do czego ona służy.

# BazaDanychPanel

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa nie jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa tworzy tabelę z historią wszystkich nagrań. Tabela zawiera takie informacje jak: nazwa, imię, nazwisko, wiek, płeć osoby poddanej analizie.

- ♦ **Opis metod**

- × *BazaDanychPanel()*- Metoda nie posiada żadnego opisu. Nie wiemy do czego służą kolejne podawane argumenty oraz co ma zwracać metoda.

- ♦ **Uwagi**

Brak komentarzy.

# Complex

## ♦ Czy klasa jest opisana w kodzie?

Klasa jest opisana w kodzie.

## ♦ Zasada działania

Po analizie wszystkich metod można wnioskować, że klasa służy do obliczania fazy. Jednak przez brak komentarzy nie znamy szczegółów.

## ♦ Opis metod

- x *Complex()* - Brak komentarza do konstruktora klasy.
- x *setReal()* - Metoda jest poprawnie opisana.
- x *getReal()* - Metoda jest poprawnie opisana.
- x *setImaginary()* - Metoda jest poprawnie opisana.
- x *getImaginary()* - Metoda jest poprawnie opisana.
- x *Complex plus()* - Metoda jest poprawnie opisana.
- x *Complex minus()* - Metoda jest poprawnie opisana.
- x *Complex times()* - Metoda jest poprawnie opisana.
- x *Complex scale()* - Metoda jest poprawnie opisana.
- x *Complex conjugate()* - Metoda jest poprawnie opisana.
- x *absolute()* - Z komentarzy wiemy co zwraca metoda, ale nie znamy jej celu.
- x *ObliczFazę()* - Nie ma komentarzy do metody. Na samym końcu dowiadujemy się tylko o zakończeniu warunku if, co jest zbędne.
- x *getModuł()* - Brak opisu działania metody.
- x *getFazę()* - Brak opisu działania metody.
- x *toString()* - Brak opisu działania metody.

## ♦ Uwagi:

Bez analizy poszczególnych metod nie wiemy do czego służy klasa. Przed klasą powinien być krótki opis wyjaśniający do czego została stworzona.

# DTFT

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa oblicza pulsację, próbkę widma oraz widmo.

- ♦ **Opis metod**

- ✗ *DTFT()* - Konstruktor bazy danych jest poprawnie opisany.
- ✗ *Pulsacja()* - Metoda jest poprawnie opisana.
- ✗ *ObliczPróbkeWidma()* - Metoda jest poprawnie opisana.
- ✗ *Oblicz DTFT()* - Metoda jest poprawnie opisana.

- ♦ **Uwagi:**

Część komentarzy jest w języku polskim, a część w języku angielskim. Nazwy zmiennych zawierają polskie znaki co nie jest dobrą praktyką.

# F0Panel

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Po analizie wszystkich metod można wnioskować, że klasa służy do zaprogramowania panelu/okna w którym uruchamiane są funkcje obliczające F0. Jednak przez brak komentarzy nie znamy szczegółów.

- ♦ **Opis metod**

✗ *F0Panel()* - Brak komentarzy do konstruktora klasy.

- ♦ **Uwagi:**

Do czego służy klasa można domyśleć się tylko po nazwach zmiennych. Przed klasą powinien być krótki opis wyjaśniający do czego została stworzona. Nazwy zmiennych są w języku polskim i angielskim jednocześnie.

# FFT

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Brak opisu klasy.

- ♦ **Opis metod**

- ✗ *fft(Complex[] x)* -

- ✗ *ifft(Complex[] x)* -

- ✗ *cconvolve(Complex[] x, Complex[] y)* -

- ✗ *convolve(Complex[] x, Complex[] y)* -

- ✗ *dft(Complex[] x)* -

- ♦ **Uwagi:**

Brak konsekwencji biorąc pod uwagę cały projekt, część komentarzy po ang część po polsku. Szczątkowe opisy, brak opisów co jest zwracane, czym są przesyłane argumenty itp.

# funkcje

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa służąca do generowania spróbkowanych typowych funkcji. Dzięki klasie mamy możliwość zwrócenia spróbkowanych typowych funkcji takich jak: sinus, prostokąt, spadek wykładniczy, okres prostokąt. Każda z funkcji zwraca tablice typu double z danymi wyjściowymi funkcji.

- ♦ **Opis metod**

-

- ♦ **Uwagi:**

Brak komentarza do funkcji wewnętrznych klasy funkcje: public double[] sinus, public double[] prostokat, public double[] SpadekWykladniczy, public double[] OkresowyProstokat.



# LittleBigEndian

## ♦ Czy klasa jest opisana w kodzie?

Klasa jest opisana w kodzie.

## ♦ Zasada działania

Klasa służąca do zamiany Small Endian na BigEndian i odwrotnie.

## ♦ Opis metod

- ✕ *ZamieńSignedIntNaTablicęLittleEndian(int x) -*
- ✕ *ZamieńUnsignedIntNaTablicęLittleEndian(long x) -*
- ✕ *ZamieńShortNaTablicęLittleEndian(short x) -*
- ✕ *ZamieńUnsignedShortNaTablicęLittleEndian(int x) -*
- ✕ *UtwórzUnsignedShortZLittleEndian(byte[] UnsignedShort) -*
- ✕ *UtwórzUnsignedIntZLittleEndian(byte[] UnsignedInt) -*
- ✕ *UtwórzSignedShortZLittleEndian(byte[] SignedShort) -*
- ✕ *UtwórzSignedIntZLittleEndian(byte[] SignedInt) -*
- ✕ *UtwórzSignedInt24bitZLittleEndian(byte[] bit24) -*

## ♦ Uwagi:

Brak komentarzy dla metody wewnętrznej:

`public static int UtwórzSignedInt24bitZLittleEndian.`

# MainWindow

- ♦ Czy klasa jest opisana w kodzie?

-

- ♦ Zasada działania

-

- ♦ Opis metod

-

- ♦ Uwagi:

-

# NagłówekPlikuWave

- ♦ Czy klasa jest opisana w kodzie?

-

- ♦ Zasada działania

-

- ♦ Opis metod

-

- ♦ Uwagi:

-

# Nagranie

- ♦ Czy klasa jest opisana w kodzie?

-

- ♦ Zasada działania

-

- ♦ Opis metod

-

- ♦ Uwagi:

-

# Nagrywarka

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

-

- ♦ **Opis metod**

- x Nagraj()* - Metoda jest poprawnie opisana.
- x Start()* - Metoda jest poprawnie opisana.
- x Stop()* - Metoda jest poprawnie opisana.
- x setPath()* - Metoda jest poprawnie opisana.
- x getPath()* - Metoda jest poprawnie opisana.

- ♦ **Uwagi**

Klasa zawiera wymagane informacje z wyjątkiem opisu klasy.

# PlikWave

## ♦ Czy klasa jest opisana w kodzie?

Klasa jest opisana w kodzie.

## ♦ Zasada działania

-

## ♦ Opis metod

- x *OpisStatusu()* - Metoda jest poprawnie opisana.
- x *getStatusPlikuWave()* - Metoda jest poprawnie opisana.
- x *getStatusNagłówkaPlikuWave()* - Metoda jest poprawnie opisana.
- x *OtwórzIstniejącyPlik()* - Metoda jest poprawnie opisana.
- x *ZamknijPlik()* - Metoda jest poprawnie opisana.
- x *PodajPozycjęData()* - Metoda jest poprawnie opisana.
- x *getWielkośćPliku()* - Metoda jest poprawnie opisana.
- x *getLiczbaKanałów()* - Metoda jest poprawnie opisana.
- x *getLiczbaBitów()* - Metoda jest poprawnie opisana.
- x *getLiczbęBajtówNaCałąPróbkę()* - Metoda jest poprawnie opisana.
- x *getCzęstotliwośćBajtowa()* - Metoda jest poprawnie opisana.
- x *getWielkośćNagłówka()* - Metoda jest poprawnie opisana.
- x *getWielkośćBlokuDanych()* - Metoda jest poprawnie opisana.
- x *getDodatkoweParametry()* - Metoda jest poprawnie opisana.
- x *getCzęstotliwośćPróbkowania()* - Metoda jest poprawnie opisana.
- x *getKodowanieAudio()* - Metoda jest poprawnie opisana.
- x *getCałąPróbkę()* - Metoda jest poprawnie opisana.
- x *getLiczbęPróbek()* - Metoda jest poprawnie opisana.
- x *getOdczytajWszystkiePróbki()* - Metoda jest poprawnie opisana.
- x *getPobierzKilkaPróbek()* - *Brak opisu zwracanych wartości.*
- x *UtwórzNowyPlik()* - Metoda jest poprawnie opisana.
- x *DopiszDaneDoPliku()* - Metoda jest poprawnie opisana.
- x *OdczytajDanePozaDźwiękiem()* - Metoda jest poprawnie opisana.

## ♦ Uwagi

Klasa nie zawiera jedynie opisu zwracanych wartości dla metody **PobierzKilkaPróbek**

# Rysowanie.java – PanelOkna

## ♦ Czy klasa jest opisana w kodzie?

Klasa jest opisana w kodzie.

## ♦ Zasada działania

Klasa na podstawie podanych wartości takich jak: liczba podziałek X oraz Y, największa oraz najmniejsza wartość widma i najmniejsza oraz największa częstotliwość tworzy skalę skale do wykresu, a następnie sam wykres widma. Podczas tworzenia wykresu widma program przelicza podane parametry, na kolejne współrzędne na wykresie, między którymi tworzy linie. Wykres jest tworzony do połowy obszaru ze skalą. Dalsza część wykresu jest odbiciem lustrzanym pierwszej połowy wykresu. Kolejne metody tej klasy odpowiadają za znalezienie największej i najmniejszej wartości amplitudy.

## ♦ Opis metod

- x *PanelOkna()* - Metoda jest poprawnie opisana.
- x *paintComponent()* - Metoda jest poprawnie opisana.
- x *RysujSkalę()* - Brak objaśnień, za co odpowiadają poszczególne składowe.
- x *RysujWidmo()* - Brak objaśnień, za co odpowiadają poszczególne składowe.
- x *PrzeliczWspółrzędne()* - Metoda jest poprawnie opisana.
- x *Lustro()* - Nie wiadomo za co tak naprawdę odpowiada metoda.
- x *ZnajdźMaksAmplitudę()* - Brak objaśnień, za co odpowiadają poszczególne składowe.
- x *ZnajdźMinAmplitudę()* - Brak objaśnień, za co odpowiadają poszczególne składowe.

## ♦ Uwagi

**W metodzie RysujSkalę** brakuje opisów w komentarzach, za co odpowiadają kolejne argumenty, właściwości, czy też odwołania do innych metod.

W metodzie **RysujWidmo** podobnie jak wyżej brakuje opisów w komentarzach. Głównie chodzi o komentarze, które objaśniałyby czym są kolejne argumenty używane w metodzie.

Jeśli chodzi o metodę **Lustro** to w zasadzie brak jakiegokolwiek wyjaśnienia jej działania. Jedynie po nazwie możemy się domyślać, że prawdopodobnie chodzi o jakieś odbicie lustrzane.

W metodach **ZnajdźMaksAmplitudę** i **ZnajdźMinAmplitudę** brakuje opisu w komentarzach dla czynności wykonywanych warunkowo... w klamrach „if”.

# Rysowanie.java – Ramka

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa *Ramka* odpowiada za utworzenie ramki o odpowiednim rozmiarze oraz ustawienie tytułu dla tworzonego wykresu. Dzięki zastosowaniu odpowiednich narzędzi, klasa ta powoduje umieszczenie okna z rysunkiem w centralnym miejscu ekranu naszego komputera.

- ♦ **Opis metod**

- × *Ramka()* - Niedokładnie opisana. Wewnątrz konstruktora praktycznie nie ma komentarzy objaśniających.
- × *UstawMnieNaŚrodku()* - Metoda jest poprawnie opisana.

- ♦ **Uwagi**

W konstruktorze tej klasy brakuje komentarzy, które pomogłyby w zrozumieniu działania kodu na poszczególnych etapach.



# Rysowanie.java – Rysowanie

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa jest opisana w kodzie.

- ♦ **Zasada działania**

Klasa ta jest odpowiedzialna za wykonanie się, skompilowanie całego rysunku, czyli panelu pod wykres widma jak i samego widma. W klasie tej są przekazywane parametry, które później wykorzystują klasy Ramka oraz PanelOkna. Znajduje się tutaj również metoda main, dzięki której program uruchamia się i możemy zobaczyć wygenerowany obraz wraz z wykresem widma.

- ♦ **Opis metod**

- × *main()* - Metoda jest poprawnie opisana.
- × *WprowadźWidmo()* - Brak opisu działania metody, nie ma dostatecznej informacji co się dzieje wewnątrz metody.

- ♦ **Uwagi**

Metoda main() niedokładnie opisana.

# widmo

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa nie jest opisana w kodzie.

- ♦ **Zasada działania**

-

- ♦ **Opis metod**

- ✗ `getCzęstotliwość()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `getWidmoAmplitudowe()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `getWidmoFazowe()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `get_dB()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `toString()` - Brak opisu metody oraz zwracanych wartości.

- ♦ **Uwagi**

Klasa nie zawiera żadnego z wymaganych opisów.

# WidmoPanel

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa nie jest opisana w kodzie.

- ♦ **Zasada działania**

-

- ♦ **Opis metod**

- ✗ WidmoPanel() - Brak opisu metody oraz zwracanych wartości.
- ✗ wyświetlWidmo() - Brak opisu metody oraz zwracanych wartości.

- ♦ **Uwagi**

Klasa nie zawiera żadnego z wymaganych opisów.

# Wykres

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa nie jest opisana w kodzie.

- ♦ **Zasada działania**

-

- ♦ **Opis metod**

- ✗ `initUI()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `createDataset()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `createChart()` - Brak opisu metody oraz zwracanych wartości.

- ♦ **Uwagi**

Klasa nie zawiera żadnego z wymaganych opisów.

# WykresWidma

- ♦ **Czy klasa jest opisana w kodzie?**

Klasa nie jest opisana w kodzie.

- ♦ **Zasada działania**

-

- ♦ **Opis metod**

- ✗ `initUI()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `createDataset()` - Brak opisu metody oraz zwracanych wartości.
- ✗ `createChart()` - Brak opisu metody oraz zwracanych wartości.

- ♦ **Uwagi**

Klasa nie zawiera żadnego z wymaganych opisów.