



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Informatyki, Elektroniki i Telekomunikacji

Dokumentacja do projektu:
digital-filters-lib

z przedmiotu:
Języki Programowania Obiektowego
Elektronika i Telekomunikacja, 3. rok

Kinga Olszewska
środa 8:00

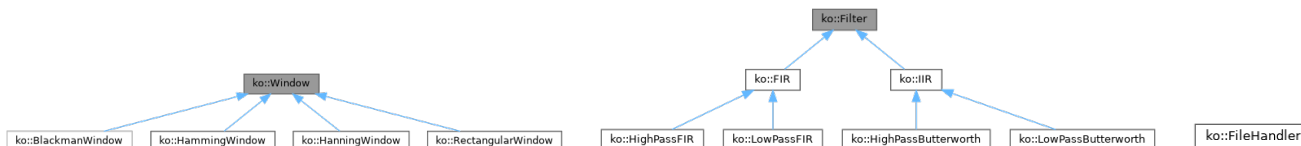
prowadzący: mgr inż. Jakub Zimnol

10.01.2026

1 Wprowadzenie

Celem projektu jest implementacja biblioteki do cyfrowego przetwarzania sygnałów, umożliwiającej realizację filtrów cyfrowych oraz funkcji okiennych. Projekt stanowi bibliotekę napisaną w języku C++, pozwalającą na projektowanie i stosowanie filtrów FIR oraz IIR, a także na wykorzystanie funkcji okien sygnałowych w analizie widmowej. Biblioteka może znaleźć zastosowanie m.in. w analizie sygnałów, przetwarzaniu sygnałów pomiarowych.

2 Opis zaimplementowanych klas



Rysunek 1: Hierarchia klas

2.1 Window

Klasa bazowa definiująca wspólny interfejs dla okien sygnałowych.

2.1.1 BlackmanWindow

Klasa dziedzicząca po klasie `Window`, reprezentująca okno Blackmana, charakteryzujące się bardzo dobrym tłumieniem listków bocznych kosztem szerszego listka głównego.

2.1.2 HammingWindow

Klasa dziedzicząca po klasie `Window`, reprezentująca okno Hamminga, stanowiące kompromis pomiędzy tłumieniem listków bocznych a rozdzielczością częstotliwościową.

2.1.3 HanningWindow

Klasa dziedzicząca po klasie `Window`, reprezentująca okno Hanninga, zmniejszające przeciek widmowy poprzez wygładzenie sygnału na jego krańcach.

2.1.4 RectangularWindow

Klasa dziedzicząca po klasie `Window`, reprezentująca okno prostokątne, będące najprostszą formą okna sygnałowego.

2.2 Filter

Klasa bazowa definiująca wspólny interfejs dla filtrów cyfrowych.

2.2.1 FIR

Klasa dziedzicząca po klasie `Filter`, reprezentująca filtry o skończonej odpowiedzi impulsowej.

2.2.2 HighPassFIR

Klasa dziedzicząca po klasie **FIR**, implementująca górnoprzepustowy filtr o skończonej odpowiedzi impulsowej na bazie wybranego okna.

2.2.3 LowPassFIR

Klasa dziedzicząca po klasie **FIR**, implementująca dolnoprzepustowy filtr o skończonej odpowiedzi impulsowej na bazie wybranego okna.

2.2.4 IIR

Klasa dziedzicząca po klasie **Filter**, reprezentująca filtry o nieskończonej odpowiedzi impulsowej.

2.2.5 HighPassButterworth

Klasa dziedzicząca po klasie **IIR**, implementująca górnoprzepustowy filtr typu Butterworth o nieskończonej odpowiedzi impulsowej.

2.2.6 LowPassButterworth

Klasa dziedzicząca po klasie **IIR**, implementująca dolnoprzepustowy filtr typu Butterworth o nieskończonej odpowiedzi impulsowej.

2.3 FileHandler

Klasa odpowiedzialna za obsługę danych wejściowych oraz wyjściowych, w tym za wczytywanie i zapisywanie danych do pliku.

3 Opis uruchomienia

Opis uruchomienia biblioteki wraz z przykładowym plikiem znajduje się również w repozytorium projektu na platformie GitHub: <https://github.com/kolszewska16/digital-filters-lib>.

3.1 Wymagania systemowe

Projekt był rozwijany i testowany w środowisku systemu operacyjnego Linux. Do poprawnej kompilacji wymagane są:

- CMake w wersji co najmniej 3.10
- kompilator języka C++ z obsługą standardu C++17

3.2 Uruchomienie i kompilacja biblioteki

Projekt wykorzystuje system budowania **CMake**. W celu zbudowania biblioteki należy wykonać następujące kroki z poziomu katalogu głównego projektu:

```
mkdir build
cd build
cmake ..
make
```

Po poprawnym zakończeniu procesu kompilacji biblioteka zostanie zbudowana w katalogu **build**.

3.3 Uruchomienie testów jednostkowych

Testy jednostkowe można uruchomić z poziomu katalogu `build` przy pomocy poleceń:

```
./filehandler-test  
./FIRfilter-test  
./IIRfilter-test  
./window-test
```

3.4 Uruchomienie przykładu

Przykładowy program przedstawiający działanie biblioteki można uruchomić z poziomu katalogu `build` przy pomocy poniższych poleceń:

```
./generate-signal  
./main
```

Literatura

- [1] Tomasz Zieliński. *Cyfrowe przetwarzanie sygnałów: od teorii do zastosowań*. Wydawnictwa Komunikacji i Łączności, 2007.
- [2] Piotr Szawdyński / cpp0x.pl. Obsługa plików. <https://cpp0x.pl/kursy/Kurs-C++/Dodatkowe-materialy/Obsluga-plikow/305>. Dostęp: 2025-12-23.
- [3] Tom / The Code Hound. Butterworth Filter Design in C++. <https://thecodehound.com/butterworth-filter-design-in-c/>, 2023. Dostęp: 2026-01-05.
- [4] Kinga Olszewska. Digital-Signal-Processing - materiały z zajęć. <https://github.com/kolszewska16/Digital-Signal-Processing>, 2025. Dostęp: 2025-12-23.