



**AKADEMIA GÓRNICZO-HUTNICZA**

Dokumentacja do projektu

# **Biblioteka obsługi bufora kołowego**

z przedmiotu

## **Języki Programowania Obiektowego**

Elektronika i Telekomunikacja, 3 Rok

*Szymon Smoła*

Piątek 11:30

prowadzący: mgr inż. Jakub Zimnol

07.01.2025

# 1. Opis projektu

Tematem projektu jest biblioteka w języku C++ obsługująca bufor cykliczny. Bufor cykliczny to struktura danych o stałej pojemności, która pozwala na przechowywanie elementów w sposób cykliczny, umożliwiając nadpisywanie najstarszych danych w przypadku przekroczenia maksymalnej pojemności. Implementacja wspiera operacje takie jak dodawanie, usuwanie oraz wyświetlanie elementów, a także porównywanie zawartości dwóch buforów.

Bufory cykliczne znajdują zastosowanie w:

- Zarządzaniu danymi w systemach o ograniczonej pamięci.
- W systemach komunikacyjnych (np. FIFO w protokołach sieciowych).
- Przechowywaniu ostatnich wyników w systemach analizy danych.

## 2. Struktura

Projekt składa się z dwóch głównych plików:

- CircularBuffer.hpp: Definicja szablonowej klasy CircularBuffer, zawierająca implementację bufora cyklicznego.
- main.cpp: Plik testowy prezentujący działanie klasy CircularBuffer.

## 3. Funkcjonalności

- Dodawanie elementu (push): Wstawia nowy element do bufora. W przypadku, kiedy bufor jest pełny nadpisywany zostaje najstarszy element.
- Usuwanie elementu (pop): Usuwa najstarszy element z bufora. Wyświetla komunikat w przypadku próby usunięcia z pustego bufora.
- Wyświetlanie zawartości (display): Prezentuje aktualne elementy w buforze w kolejności ich przechowywania.
- Przeciążenie operatorów:
  - `=` : Przypisanie jednego bufora do drugiego.
  - `==` : Porównanie dwóch buforów.
  - `!=` : Sprawdzenie różnic pomiędzy buforami.
- Metody pomocnicze:
  - `is_full()`: Sprawdza, czy bufor jest pełny.
  - `is_empty()`: Sprawdza, czy bufor jest pusty.

## 4. Kompilacja i uruchomienie

Aplikację można skompilować za pomocą kompilatora g++:

```
g++ main.cpp -o main
```

Po pomyślnej kompilacji program uruchamia się poleceniem:

```
./main
```

## 5. Podsumowanie

Projekt stanowi przykład efektywnej implementacji bufora cyklicznego w języku C++ z wykorzystaniem programowania obiektowego oraz szablonów. Implementacja charakteryzuje się elastycznością, pozwalając na przechowywanie elementów różnych typów, a także na łatwe rozszerzenie funkcjonalności.