



AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do obsługi bufora kołowego

z przedmiotu

Języki Programowania Obiektowego

Elektronika i Telekomunikacja 3 rok

Sebastian Broda

grupa piątek 11:30

prowadzący: mgr inż. Jakub Zimnol

07.01.2025

CEL PROJEKTU

Stworzenie biblioteki do obsługi bufora kołowego w języku C++, czyli struktury przechowującej dane w sposób cykliczny. Znajduje ona zastosowanie m. in. w systemach o ograniczonych zasobach, gdyż najstarszy wprowadzony element po wypełnieniu bufora jest nadpisywany przez nowy element.

OPIS WYKONANIA PROJEKTU

1. Stworzenie pliku *buffer.hpp* zawierającego klasę szablونową *CircularBuffer*.

Pola zaimplementowane w klasie:

m_buffer – wskaźnik na dynamicznie alokowaną tablicę, przechowywane są w nim elementy bufora,
m_head – wskazuje na indeks tablicy, gdzie zapisany jest najstarszy element (pierwszy do odczytu),
m_tail – wskazuje na indeks tablicy, w którym może zostać zapisany kolejny podany element,
m_capacity – określa maksymalną pojemność bufora
m_size – określa bieżącą liczbę elementów w buforze.

Konstruktor inicjalizuje pole **m_capacity** na wartość podaną przez użytkownika, pozostałe pola na wartość 0.

Metody zaimplementowane w klasie:

push() – pozwala na dodanie podanego elementu do tablicy **m_buffer**, do indeksu **m_tail**, inkrementuje wartość **m_tail** i **m_size**, a jeśli bufor jest pełny, inkrementuje wartość indeksu **m_head**,
pop() – pozwala na usunięcie z bufora najstarszego podanego elementu, następnie indeks **m_head** zwiększa się o 1 i wskazuje kolejny w kolejności starszeństwa element,
front() – pozwala wyświetlić najstarszy w kolejności element, bez usuwania go,
isEmpty(), isFull() – zwracają logiczną 1, jeśli bufor jest odpowiednio pusty lub pełny,
size() – zwraca wartość pola **m_size**
print() – wyświetla wszystkie liczby znajdujące się w buforze.

2. Stworzenie pliku *main.cpp*, reprezentującego działanie bufora.

Po uruchomieniu programu, użytkownik proszony jest o podanie pojemności bufora, następnie ma możliwość wybrania jednej z pięciu dostępnych opcji:

```
Please enter buffer capacity: 5
-----
1. Add element to buffer
2. Remove element from buffer
3. Display buffer
4. View last element
0. End
Please choose option: |
```

Prezentacja działania programu:

```
-----  
buffer: 12 73 34 54 66  
Is buffer full?: Yes  
Current size: 5 elements  
-----
```

```
Please choose option: 1  
-----  
99  
-----  
1. Add element to buffer  
2. Remove element from buffer  
3. Display buffer  
4. View last element  
0. End  
Please choose option: 3  
-----  
buffer: 73 34 54 66 99  
Is buffer full?: Yes  
Current size: 5 elements  
-----
```

```
Please choose option: 2  
-----  
Removed element: 73  
-----  
1. Add element to buffer  
2. Remove element from buffer  
3. Display buffer  
4. View last element  
0. End  
Please choose option: 3  
-----  
buffer: 34 54 66 99  
Is buffer full?: No  
Current size: 4 elements  
-----
```

```
Please choose option: 4  
-----  
Last element: 34  
-----
```