

# Algorytmy i struktury danych - Stosy i kolejki

Marcin Żurowski

26 kwietnia 2024

# Stos - funkcje i procedury

```
STRUCT-STACK  
  data[1..5]  
  top = 0  
STACK-EMPTY(S)  
PUSH(S,k)  
POP(S)
```

## Kolejka - funkcje i procedury

```
STRUCT-QUEUE
```

```
  data[1..5]
```

```
  length = 5
```

```
  head = 1
```

```
  tail = 1
```

```
QUEUE-EMPTY(Q)
```

```
ENQUEUE(Q, k)
```

```
DEQUEUE(Q)
```

```
STRUCT-STACK S
INIT(S) /*inicjuje zmienne*/
PUSH(S,1)
PUSH(S,2)
WRITE(S) //1 2
PUSH(S,3)
WRITE(S) //1 2 3
WRITE(STACK-EMPTY(S)) //false
PUSH(S,4)
PUSH(S,5)
PUSH(S,6) //nadmiar
```

```
WRITE(POP(S)) //5  
WRITE(POP(S)) //4  
WRITE(POP(S)) //3  
WRITE(POP(S)) //2  
WRITE(S) //1  
WRITE(POP(S)) //1  
WRITE(POP(S)) //niedomiar  
WRITE(S) //  
WRITE(STACK-EMPTY(S)) //true
```

```
STRUCT-QUEUE Q
INIT(Q) /*inicjuje zmienne*/
ENQUEUE(Q,1)
ENQUEUE(Q,2)
WRITE(Q) //1 2
ENQUEUE(Q,3)
WRITE(Q) //1 2 3
WRITE(QUEUE-EMPTY(Q)) //false
WRITE(DEQUEUE(Q)) //1
WRITE(DEQUEUE(Q)) //2
ENQUEUE(Q,4)
ENQUEUE(Q,5)
ENQUEUE(Q,6)
ENQUEUE(Q,7)
ENQUEUE(Q,8) //nadmiar
```

```
WRITE(Q) //3 4 5 6 7
WRITE(DEQUEUE(Q)) //3
WRITE(DEQUEUE(Q)) //4
WRITE(DEQUEUE(Q)) //5
WRITE(DEQUEUE(Q)) //6
WRITE(DEQUEUE(Q)) //7
WRITE(DEQUEUE(Q)) //niedomiar
WRITE(Q) //
WRITE(QUEUE-EMPTY(Q)) //true
```