

- I. Stos jest strukturą danych, w której kolejne elementy są odkładane na wierzch stosu i z wierzchołka stosu są pobierane w kolejności odwrotnej niż były wkładane (LIFO).

Utwórz własną implementację `MyStack`, który będzie:

- przyjmował jako argument konstruktora wielkość stosu (ilość elementów jakie można w nim umieścić);
- zawierał metodę `void push(...)` pozwalającą na dodanie do stosu elementu;
- zawierał metodę `... pop()` zwracającą element pobrany ze stosu.

Ponadto przygotuj klasę `Student` definiującą prywatne pola:

- `int sNumber`,
- `String name`.

Klasa będzie również definiować:

- dwuargumentowy konstruktor, inicjujący pola klasy;
- metodę `show()` wyświetlającą w konsoli stan wszystkich pól obiektu;

Następnie utwórz program sprawdzający poprawność implementacji dla przypadków:

- dodawania i pobierania określonej liczby elementów;
- umieszczenia większej liczby elementów na stosie, niż to jest dozwolone;
- próby pobrania elementu z stosu, gdy stos jest pusty.

Zrealizuj to zadanie na dwa sposoby, najpierw wykorzystując tablicę, a następnie listę jednokierunkową.

- II. Kolejka jest strukturą danych, w której nowe dane dopisywane są na końcu kolejki, a pobierane są z początku kolejki.

Utwórz własną implementację `MyQueue`, która będzie:

- zawierała metodę `void put(...)` pozwalającą na dodanie do kolejki elementu;
- zawierała metodę `... get()` zwracającą element pobrany z kolejki.

Dane są dwa koszyki (oznaczone jako A i B) zawierające obiekty klasy `String`. W koszyku A znajdują się ciągi znaków: PSG, Atletico Madryt, Sporting CP, Inter, Benfica, Villarreal, RB Salzburg, Chelsea, natomiast w koszyku B znajdują się: Manchester City, Liverpool, Ajax Amsterdam, Real Madryt, Bayern Monachium, Manchester United, Lille, Juventus. Utwórz dwie struktury reprezentujące koszyki i naprzemiennie wyciągaj z nich elementy, które zostaną umieszczone w kolejce. Gdy koszyki będą już puste, wówczas rozpocznij wyciąganie elementów z kolejki. Wyciągaj sekwencyjnie po dwa elementy i tak powstałą parę wyświetl na ekranie. Powtarzaj sekwencję aż do opróżnienia kolejki.