

## Kolokwium nr 2 AiSD

### Cześć A: Zadania sprawdzające znajomość implementacji konkretnych struktur danych

1. **Implementacja tablicy dynamicznej** (dst – tablica liczb całkowitych, db – tablica wartości opakowanych np. klasami Integer, Double, Float lub obiektów typu String, bdb – tablica złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.)
2. **Implementacja uporządkowanej tablicy dynamicznej** (dst – tablica liczb całkowitych, db – tablica wartości opakowanych np. klasami Integer, Double, Float lub obiektów typu String, bdb – tablica złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.)
3. **Implementacja listy powiązanej liczb całkowitych** (dst – lista powiązana jednostronna, db – lista powiązana dwustronna, bdb – lista powiązana dwukierunkowa)
4. **Implementacja listy powiązanej uporządkowanej liczb całkowitych** (dst – lista powiązana jednostronna, db – lista powiązana dwustronna, bdb – lista powiązana dwukierunkowa)
5. **Implementacja drzewa BST liczb całkowitych** (dst – jedna funkcjonalność, db – dwie funkcjonalności, bdb – trzy funkcjonalności)
6. **Implementacja tablicy mieszającej metodą łańcuchową** (dst – jedna funkcjonalność, db – dwie funkcjonalności, bdb – trzy funkcjonalności)

### Cześć B: Zadania sprawdzające umiejętność implementowania abstrakcyjnych struktur danych przy wykorzystaniu standardowych konkretnych struktur danych

*Przy implementacji można stosować metody: przez osadzanie lub przez dziedziczenie.*

1. **Implementacja stosu za pomocą klasy ArrayList** (dst – stos liczb całkowitych, db – stos wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – stos złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
  - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów A, B, C, D, E. Przebieg eksperymentu: tworzymy strukturę, wstawiamy do struktury A, B, C; usuwamy dwa elementy; wstawiamy D i E; usuwamy wszystkie elementy wypisując je na ekran.
2. **Implementacja stosu za pomocą klasy LinkedList** (dst – stos liczb całkowitych, db – stos wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – stos złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
  - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów A, B, C, D, E. Przebieg eksperymentu: tworzymy strukturę, wstawiamy do struktury A, B, C; usuwamy dwa elementy; wstawiamy D i E; usuwamy wszystkie elementy wypisując je na ekran.
3. **Implementacja kolejki za pomocą klasy ArrayList** (dst – kolejka liczb całkowitych, db – kolejka wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – kolejka złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
  - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów A, B, C, D, E. Przebieg eksperymentu: tworzymy strukturę, wstawiamy do struktury A, B, C; usuwamy dwa elementy; wstawiamy D i E; usuwamy wszystkie elementy wypisując je na ekran.
4. **Implementacja kolejki za pomocą klasy LinkedList** (dst – kolejka liczb całkowitych, db – kolejka wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – kolejka złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
  - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów A, B, C, D, E. Przebieg eksperymentu: tworzymy strukturę, wstawiamy do struktury A, B, C; usuwamy dwa elementy; wstawiamy D i E; usuwamy wszystkie elementy wypisując je na ekran.
5. **Implementacja listy z bezpośrednim dostępem do każdego elementu poprzez podanie jego numeru za pomocą klasy ArrayList** (dst – lista liczb całkowitych, db – lista wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – lista złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).

- Wykonać następujący eksperyment ze strukturą w której bierze udział 6 różnych elementów A, B, C, D, E, F (). Przebieg eksperymentu: tworzymy strukturę, wstawiamy do listy A, B, C, D; wypisujemy elementy listy; usuwamy element drugi i trzeci; wypisujemy elementy listy; wstawiamy E i F; wypisujemy elementy listy.
- Implementacja listy dwustronnej za pomocą klasy** LinkedList (dst – lista liczb całkowitych, db – lista wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – lista złożonych obiektów zdefiniowanych przez użytkownika, np. tablica osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
    - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów A, B, C, D, E. Przebieg eksperymentu: tworzymy listę L, wstawiamy do listy L elementy A, B i C od lewej strony; wypisujemy elementy listy; wstawiamy do listy L elementy D i E od prawej strony; wypisujemy elementy listy; usuwamy element z lewej strony i element z prawej strony; wypisujemy elementy listy; sprawdzamy, czy na liście jest element B i wypisujemy wynik testu; sprawdzamy, czy na liście jest element E i wypisujemy wynik testu.
  - Implementacja zbioru za pomocą klasy** ArrayList (dst – zbiór liczb całkowitych, db – zbiór wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – zbiór złożonych obiektów zdefiniowanych przez użytkownika, np. zbiór osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
    - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów a, b, c, d, e. Przebieg eksperymentu: tworzymy zbiory A i B, wstawiamy do A elementy a, b, c, d oraz do B elementy c, d, e; wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B; usuwamy ze zbioru A element c i ze zbioru B element e; ponownie wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B.
  - Implementacja zbioru za pomocą klasy** TreeSet (dst – zbiór liczb całkowitych, db – zbiór wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – zbiór złożonych obiektów zdefiniowanych przez użytkownika, np. zbiór osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
    - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów a, b, c, d, e. Przebieg eksperymentu: tworzymy zbiory A i B, wstawiamy do A elementy a, b, c, d oraz do B elementy c, d, e; wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B; usuwamy ze zbioru A element c i ze zbioru B element e; ponownie wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B.
  - Implementacja zbioru za pomocą klasy** HashSet (dst – zbiór liczb całkowitych, db – zbiór wartości opakowanych np. klasami Double, Float lub obiektów typu String, bdb – zbiór złożonych obiektów zdefiniowanych przez użytkownika, np. zbiór osób [imię nazwisko, wiek], książek [tytuł, wydawnictwo, rok wydania] itp.).
    - Wykonać następujący eksperyment ze strukturą w której bierze udział 5 różnych elementów a, b, c, d, e. Przebieg eksperymentu: tworzymy zbiory A i B, wstawiamy do A elementy a, b, c, d oraz do B elementy c, d, e; wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B; usuwamy ze zbioru A element c i ze zbioru B element e; ponownie wyliczamy i wypisujemy zbiory: iloczyn A i B, suma A i B, różnica A\B.

## Część C: Zadania sprawdzające umiejętność implementowania metod sortowania

- Implementacja metody sortowania bąbelkowego (bubble sort)** dla tablicy liczb całkowitych.
  - Wykonać eksperyment polegający na sortowaniu liczb: 5, 6, 3, 2, 8, -4, -9, 9, 0, 1
- Implementacja metody sortowania przez wybór (selection sort)** dla tablicy dynamicznej (Array List) liczb całkowitych.
  - Wykonać eksperyment polegający na sortowaniu liczb: 5, 6, 3, 2, 8, -4, -9, 9, 0, 1
- Implementacja metody sortowania przez wstawianie (insertion sort)** dla tablicy dynamicznej (Array List) liczb całkowitych.
  - Wykonać eksperyment polegający na sortowaniu liczb: 5, 6, 3, 2, 8, -4, -9, 9, 0, 1
- Implementacja metody sortowania szybkiego (quick sort)** dla tablicy dynamicznej (Array List) liczb całkowitych.
  - Wykonać eksperyment polegający na sortowaniu liczb: 5, 6, 3, 2, 8, -4, -9, 9, 0, 1
- Implementacja metody sortowania przez scalanie (merge sort)** dla tablicy dynamicznej (Array List) liczb całkowitych.
  - Wykonać eksperyment polegający na sortowaniu liczb: 5, 6, 3, 2, 8, -4, -9, 9, 0, 1