

# Zásobník

Strategie: LIFO

Implementace seznamem:

```
public class Node {
    private Object element;
    private Node next;
    public Node() { element=null; next=null; }
    public Node(Object element, Node next){
        this.element=element;
        this.next=next;
    }
    public Object getElement(){ return (element); }
    public Node getNext() { return(next); }
    public void setElement(Object e){ element=e; }
    public void setNext(Node n){ next=n; }
}

public class StackL {
    private Node sp; // ukazatel na vrchol zasobniku
    private int sz; // pocet objektu v zasobniku
    public StackL() {
        sp=null;
        sz=0;
    }
    public boolean isEmpty() {
        if (sp==null)
            return true;
        else return false;
    }
    public int size(){ return(sz);} // vraci pocet objektu v
    zasobniku
    public void push(Object o){
        sp=new Node(o,sp);
        sz++;}
}
```

Polem:

```
public class StackA {
    private final int CAPACITY=100;
    private Object[] s; // pole pro ulozeni prvku zasobniku
    private int sz;
    private int sp; //ukazatel zasobniku
    public StackA(){
        s=new Object[CAPACITY];
        sp=0;
        sz=0;}
    public boolean isEmpty(){
        if (sp==0)
            return(true);
        else return(false);
    }
    public int size(){
        return(sz);
    }public Object top()throws StackEmptyException {
        if (sp==0)
            throw new StackEmptyException("Stack is empty");
        else return(s[sp-1]);
    }
    public Object pop() throws StackEmptyException {
        if (sp==0)
            throw new StackEmptyException("Stack is empty");
        else return(s[--sp]);
    }
    public void push(Object e) throws StackFullException{
        if (sp>=CAPACITY)
            throw new StackFullException("Stack is full");
        else s[sp++]=e;
    }
}
```

## ADT Fronta

Strategie: FIFO

Metody:

- Enqueue(o) – vkládá object o na konec fronty.
- Dequeue() – Odstraňuje object z čela fronty. Pokud je fronta prázdná – chyba
- Size() – Vrací počet prvků ve frontě
- isEmpty()
- front – Vrací objekt z čela fronty bez jeho odstranění.

## ADT Oboustraná fronta

Stejná jako fronta, ale umožňuje vkládání a výběr z obou konců fronty

## ADT Vektor

Je to lineární posloupnost prvků V, která obsahuje n prvků. Každý prvek vektoru V je přístupný prostřednictvím indexu r (rank) rozsahu [0,] n-1. Vektor připomíná datový typ pole, ale není to pole!

```
public class ArrayVector {
    private Object[] a; // Array storing the elements of the vector
    private int capacity = 16; // Length of array a
    private int size = 0; // Number of elements stored in the vector
    // Constructor
    public ArrayVector() { a = new Object[capacity]; } // O(1) time
    // Accessor methods
    public Object elemAtRank(int r) { return a[r]; } // O(1) time
    public int size() { return size; }
    public boolean isEmpty() { return size() == 0; } // O(1) time
    // Modifier methods
    public Object replaceAtRank(int r, Object e) { // O(1) time
        Object temp = a[r];
        a[r] = e;
        return temp;
    }
    public Object removeAtRank(int r) { // O(n) time
        Object temp = a[r];
        for (int i=r; i<size-1; i++) // Shift elements down
            a[i] = a[i+1];
        size--;
        return temp;
    }
    public void insertAtRank(int r, Object e) { // O(n) time
        if (size == capacity) { // An overflow
            capacity *= 2;
            Object[] b = new Object[capacity];
            for (int i=0; i<size; i++)
                b[i] = a[i];
            a = b;
        }
        for (int i=size-1; i>=r; i--) // Shift elements up
            a[i+1] = a[i];
        a[r] = e;
        size++;
    }
}
```

## ADT Seznam

Seznam je lineární posloupnost prvků, které jsou propojeny ukazateli. Prvek se do seznamu vkládá na určitou pozici.

Metody pro práci se seznamem:

- first() – odkaz na první prvek seznamu
- last()
- atd. ...

## Kolekce v javě

- jsou to objekty tříd z balíku java.util
- slouží k uchování většího množství objekt

ArrayList – pole proměnné délky, někdy nazývané seznam.

Set – rozhraní k množinám

HashSet – množina. Třída implementující metody Set

HashMap – mapa. Třída implementující metody rozhraní Map