

15. ADT - Zásobník (princip, realizace, základní operace)

Element

Jak fronta, tak zásobník jsou architektury, které lze realizovat pomocí lineárních spojových seznamů. Oba mají strukturu tvořenou **elementy**, jenž na sebe ukazují, proto se nejdříve vytvoří element, který je pro oba lineární seznamy stejný.

```
public class Element {  
    private int value;  
    private Element next;  
  
    public Element(int value) {  
        this.value = value;  
        this.next = null;  
    }  
    public int getValue() {  
        return this.value;  
    }  
    public void setValue(int value) {  
        this.value = value;  
    }  
    public Element getNext() {  
        return this.next;  
    }  
    public void setNext(Element next) {  
        this.next = next;  
    }  
}
```

Element udržuje hodnotu daného elementu (int **value**), která se nastaví při vytvoření elementu. Proměnná typu Element (Element **next**) je **ukazatel** na další Element.

Zásobník

LIFO (Last In First Out). Prvek, který byl do fronty přidán jako poslední, je z fronty odebrán jako první.
Základní metody pro práci se zásobníkem:

Konstruktor

Vytvoření nového zásobníku

isEmpty()

Ověření prázdnosti zásobníku

top()

Ukazatel na element na vrcholu.

push()

Přidání elementu na začátek zásobníku. Elementu, který se přidává, se nastaví další prvek na top. Současný top se nastaví na přidávaný element.

pop()

Odebrání elementu ze začátku zásobníku. Odebírání nelze provádět, pokud je zásobník prázdný!! Vytvoří se nový dočasný element, do kterého se uloží aktuální top. Do aktuálního topu se uloží další element (tmp.getNext()). Poté se vrátí dočasný element.

Výhody

Nekonečný (oproti poli, které má pevně definovanou velikost).

Nevýhody

Zabere delší čas na vyhledávání, oproti poli, které má konstantní rychlost.

Využití

DFS

Zásobník v poli

Vytvoří se nový zásobník s parametry:

private int array[]; – pole, které slouží jako zásobník

private int top; – počítadlo prvků

private int max; – maximální velikost zásobníku

Při přidání hodnoty, se přidá hodnota na **array[this.top]** a zvýší se top o 1.

Při odebírání se sníží top o 1, a vrátí se **hodnota array[this.top]**.

Poté se hodí metody pro zvětšení a zmenšení zásobníku.

```
public class Stack {  
    private Element top;  
  
    public Stack() {  
        this.top = null;  
    }  
    public boolean isEmpty() {  
        if (this.top == null) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
    public Element top() {  
        return this.top;  
    }  
    public void push(Element e) {  
        e.setNext(this.top);  
        this.top = e;  
    }  
    public Element pop() {  
        if (this.top != null) {  
            Element tmp;  
            tmp = this.top;  
            this.top = tmp.getNext();  
            return tmp;  
        } else {  
            return null;  
        }  
    }  
}
```