

Téma 16: ADT - Fronta

Fronta i zásobník jsou lineární spojové seznamy. Oba dva mají strukturu tvořenou **elementy**, které na sebe ukazují, proto musíme nejdříve vytvořit element, který je pro oba lineární seznamy stejný.

Fronta je lineární spojový seznam typu FIFO – First In First Out, což znamená, že první prvek, který se do fronty vloží se z ní i jako první odebere. Je to prostě stejné jako fronta na obědě nebo tak něco.

Nejprve si musím určit, odkud budu z fronty odebírat a kam budu dávat nový element. Ideální stav je aby vršek fronty sloužil k odebírání prvků z fronty. Spodek fronty bude tedy sloužit k přidávání prvků do pole.

Pokud chci zjistit zda je fronta prázdná, jednoduše se kouknu na top a pokud je roven null fronta je prázdná.

Pokud chci přidat do fronty nový prvek, zavolám metodu na přidávání. Nejprve vytvořím nový Element s požadovanou hodnotou. Poté musím zkontrolovat, zda je již ve frontě nějaký prvek nebo zda je prázdná. Pokud je fronta prázdná, nastavím nejprve nově vytvořený element jako top, poté i jako down. Pokud jsou ve frontě již nějaké prvky, pak aktuálnímu elementu na spodku fronty nastavím ukazatel na nový prvek. Poté nastavím nový prvek jako down.

Pokud chci prvek z fronty odebrat, zavolám metodu na odebírání prvků. Nejprve musím zkontrolovat, zda je fronta prázdná, pokud ano pak vrátím null. Pokud není prázdná, mohu odebírat prvky z fronty. Nejdříve do proměnné vložím hodnotu aktuálního topu. Poté do topu vložím element, na který ukazuje aktuální top. Poté zkontroluji, zda je v topu null, protože když tam bude null tak fronta je prázdná a musím nastavit down na null. Poté jen vrátím hodnotu, kterou jsem si předtím vyzvedl z topu.

```
public class Queue {
    private Element top = null;
    private Element down = null;

    public void add(int value) {
        Element e = new Element(value);
        if(!isEmpty()) {
            down.setElement(e);
        } else {
            top = e;
        }
        down = e;
    }

    public boolean isEmpty() {
        if(top == null) {
            return true;
        } else {
            return false;
        }
    }
}
```

```
    }  
}  
  
public Integer remove() {  
    if(isEmpty()) {  
        return null;  
    } else {  
        int value = top.getValue();  
        top = top.getElement();  
        if(top == null) {  
            down = null;  
        } return value;  
    }  
}  
}
```