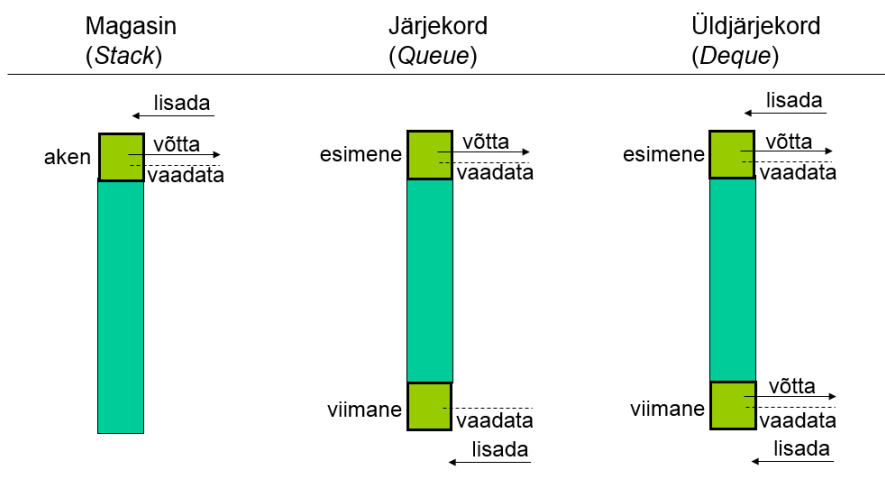


# Magasin ja järjekord

Ülevaade käesolevas käsitletavate andmestruktuuride (magasin, järjekord ja üldjärjekord) operatsioonivarudest on esitatud joonisel 1. Vastavad Java-vahendid on koondatud tabelitesse joonistel 2 ja 3. Nii järjekord kui ka üldjärjekord on soovitatav realiseerida topelt-seotud ahelana, Javas klassi *LinkedList* isendina. Sel puhul on kõik käitlemismeetodid konstantse ajalise keerukusega ( $O(1)$ ), samuti nagu seda on Java klassina *Stack* realiseeritud magasin operatsioonid.



Joonis 1: Kolm andmestruktuuri.

Operatsioon	Java
	<code>import java.util.Stack; // magasin</code>
Teha uus, tühi	<code>Stack&lt;T&gt; mag = new Stack&lt;T&gt;(); // T on elemendi tüüp</code>
Lisada element x (aknale)	<code>mag.push(x);</code>
Aknal oleva väärtus (vaadata)	<code>mag.peek();</code>
Võtta aknalt ära Võtta aknalt ja omistada y-le	<code>mag.pop(); y = mag.pop();</code>
Kas on tühi?	<code>if (mag.empty())</code>

Joonis 2: Magasin ja valik selle käitlemise meetodeid.

Operatsioon	Java
	<pre>import java.util.LinkedList; // järjekord või üldjärjekord</pre>
Teha uus, tühi	<pre>LinkedList&lt;T&gt; jrk = new LinkedList&lt;T&gt;(); // T on elemendi tüüp</pre>
Lisada element x jrk lõppu (viimaseks)	<pre>jrk.addLast(x);</pre>
Võtta esimene ära	<pre>jrk.pollFirst();</pre>
Võtta esimene ära ja omistada muutujale y	<pre>y = jrk.pollFirst();</pre>
Esimese elemendi väärtus (vaadata)	<pre>jrk.peekFirst();</pre>
Viimase elemendi väärtus (vaadata)	<pre>jrk.peekLast();</pre>
Kas on tühi?	<pre>if (jrk.isEmpty())</pre>
Lisada element x jrk algusesse (esimeseks)	<pre>jrk.addFirst(x); // lihtjärjekorrale mitte</pre>
Võtta viimane ära	<pre>jrk.pollLast(); // lihtjärjekorrast mitte</pre>
Võtta viimane ära ja omistada muutujale y	<pre>y = jrk.pollLast(); // -- " --</pre>

Joonis 3: Järjekorra / üldjärjekorra käitlemise meetodeid.