

## Dubbellänkad lista och kö

Den här uppgiften kommer att behandla det du har lärt dig i kursen så här långt samtidigt som du behöver nyttja kunskaper från tidigare java-kurser.

### Beskrivning

Du ska implementera en egen dubbellänkad lista och kunna nyttja den i en applikation som en kö. Hur du bygger upp dina klasser är upp till dig, men objekten som den länkade listan byggs av ska ligga i en egen klass (t.ex. klassen `ObjectNode`). Sedan ska den det finnas en annan klass som implementerar den dubbellänkade listan med hjälp av `ObjectNode` (t.ex. klassen `DoublyLinkedList`).

Utöver att implementera den dubbellänkade listan ska du dokumentera din Java-kod, samt testköra din implementering med ett egengjort testprogram (ett simpelt program med små testmetoder som testar din dubbellänkade lista som om den vore en kö). Skriv även några kommentarer i ditt testprogram där du förklarar dina testningar (det är viktigt att du kan motivera sina testningar).

Du ska testa att metoderna (se nedan) fungerar som de ska, utan att programmet kraschar.

T.ex. *Läggs objekten på rätt plats i listan? Returnerar `size()` korrekt storlek? Är listan korrekt länkad? Etc.*

Med hjälp av dokumentationen och testprogrammen ska en annan kursdeltagare granska och reflektera över din implementation när du är klar med inlämningen (både den dubbellänkade listan och testprogrammet).

### Krav för `ObjectNode`

De här metoderna ska finnas med:

Metod	Beskrivning
<code>getData() : E</code>	Returnera datan som finns i noden.
<code>getNext() : ObjectNode</code>	Hämta nästa nod.
<code>getPrevious() : ObjectNode</code>	Hämta föregående nod.
<code>setNext(ObjectNode node)</code>	Sätt referens till nästa nod.
<code>setPrevious(ObjectNode node)</code>	Sätt referens till föregående nod.

### Krav för den dubbellänkade listan:

De här metoderna ska finnas med:

Metod	Beskrivning
<code>size() : int</code>	Returnera antalet element i listan.
<code>add (int index, E data)</code>	Placera objektet <code>data</code> i positionen <code>index</code> . Senare element flyttas en position.
<code>addFirst (E data)</code>	Placerar objektet <code>data</code> i början av listan.
<code>addLast (E data)</code>	Placerar objektet <code>data</code> i slutet av listan.
<code>remove (int index) : E</code>	Tar bort objektet i position <code>index</code> och returnerar det borttagna objektet. Senare element flyttas en position.
<code>removeFirst() : E</code>	Tar bort första objektet i listan och returnerar det. Senare element flyttas en position.
<code>removeLast() : E</code>	Tar bort sista elementet i listan och returnerar det.
<code>get (int index) : E</code>	Returnerar objektet i position <code>index</code> .
<code>getFirst() : E</code>	Returnerar första objektet i listan (huvudet).
<code>getLast() : E</code>	Returnerar sista objektet i listan (svansen).

Ni får gärna lägga till flera metoder om ni anser att det behövs (t.ex. att stega i listan).

Tänk på att `E` ska kunna vara vilket objekt som helst.

Är listan tom ska huvudet och svansen vara `null`.

## Deadline

Sista redovisningstillfället är på labben den 18 mars. Men gör det så snart som möjligt och redovisa när du är klar med uppgiften (se under stycket redovisning).

## Paketering och inlämning

Allt ska ligga snyggt nerpackat i en .zip-fil och laddas upp till itslearning i samband med redovisning. ZIP-filen ska ha ett lämpligt namn när man skickar in den, t.ex. JonasP-inlämning1.zip

## Redovisning

Redovisning sker genom att man muntligt presenterar sin lösning för lärare under ett labbtillfälle någon gång innan kursslut (dock senast den 18 mars). Man ska även granska en annan students lösning. Det kommer att ske någon gång runt 18 mars.

Skulle ni inte hinna klart kommer det att dyka upp ett tillfälle efter kursen är slut, håll därför koll på itslearning efter kursslut om det skulle behövas.

## Granskning

Du kommer att få reda på vilket projekt du ska granska omkring den 18 mars.

Vid granskningen av en annan students arbete ska du reflektera över lösningen, kontrollera testningen och dokumentera ner det du kommer fram till i ett dokument som ska skickas in till itslearning. Tänk på att man ska vara konstruktiv när man skriver dokumentationen.

Det är endast läraren som kommer att läsa dokumentationen, det innebär alltså att den student som blir granskad inte kommer att läsa granskningsdokumentet för just sin lösning.

## Övrigt

Till er hjälp för inlämningsuppgiften har ni laborationerna samt föreläsningsanteckningarna att titta på.

Frågor kan ställas till mig när som helst via mejl [jonas.petersson@mah.se](mailto:jonas.petersson@mah.se) eller på ett labbtillfälle (som också kommer fungera som handledningstillfälle).

Lycka till!

/Jonas Petersson