Skrivarkö

Inlämningsuppgift på momentet datastrukturer

Du ska skapa en skrivarkö som hanterar jobb i form av textsträngar som skrivs ut på skärmen. Din kö ska bygga på en länkad lista. Du ska dessutom skriva ett litet testprogram som visar att din lösning fungerar.

Länkad lista

Du ska börja med att skapa en implementation av en länkad lista. Listan ska innehålla strängar men det går bra att skapa en generell länkad lista för alla datatyper. Du väljer själv hur stort API du vill implementera men det viktigaste är att du kan använda din lista som en kö och en stack, d.v.s. lägga till och ta bort data i början och i slutet av listan. Den LinkedList som finns inbyggd i Java har en onödigt stor funktionalitet. Du behöver bara addFirst, addLast, removeFirst, removeLast, size och isEmpty. Ett litet tips är att hantera listan bara som en uppsättning noder, dvs det finns bara en nod first och en nod last i den länkade listan, noden kan vara en inre klass till listan. Många exempel på nätet visar mer komplexa konstruktioner, till exempel i Ecks's javanotes.

Skrivarkö

När du skapat din länkade lista ska du bygga en kö av den. Kön ska innehålla följande metoder: boolean isEmpty(), int size(), void enqueue(String data) och String dequeue().

Testprogram

Din kö ska sedan användas i ett program som simulerar en skrivarkö. En enkel simulering lägger ett antal jobb på kön och skriver sedan ut dem. En mer avancerad lösning är att köra simuleringen som två processer där den ena skapar jobb och den andra skriver ut dem.

Inlämning

Lämna in din lösning i form av en länkad lista, en kö och en simulering via ett github-repo i classroom. Ditt repo ska innehålla källkod (välskriven och kommenterad) och en rapport med beskrivning över din lösning och exempel på testkörningar.