

# Ukesoppgaver uke 8

## **Oppgave: Om å lete etter og behandle mønstre i immunreseptorer.**

I obligatorisk oppgave 5 skal du lage et program som analyserer blodprøver . Programmet skal prøve å finne mønstre som indikerer infeksjon av et spesielt virus. Programmet skal lete i såkalte immunreseptorer, der hver reseptor er representert av en streng av (store) bokstaver. En slik streng er typisk på 15 – 30 bokstaver. En blodprøve fra en person gir mange slike strenger, og hver streng lagres som en linje i en fil, en fil for hver person.

Vi ønsker å finne mønstre i disse reseptorene. Disse mønstrene består av korte substrenger eller *subsekvenser*. En subsekvens er en String av en gitt lengde, typisk 3 eller 4 (en konstant i programmet ditt). Når vi analyserer blodet til en person vil vi finne mange slike subsekvenser, og disse vil bli lagret i én HashMap. Vi ønsker å analysere blodet til mange personer, og vi trenger derfor en beholder som tar vare på mange HashMap-er.

I denne oppgaven kan du lage noen verktøy som du kan bruke til å behandle slike subsekvenser og HashMap-er som tar vare på dem. Om du synes noe er uklart i denne oppgaven, så gjør dine egne fornuftige forutsetninger. Når du bruker de programdelene du lager her i oblig 5, må du sikkert uansett gjøre noen justeringer.

### **Oppgave 1.**

Lag en klasse, kall den gjerne SubSekvens, som kan ta vare på en subsekvens (en String) og et antall (en int som angir antall forekomster av denne subsekvensen). La alle instansvariable være private, og lag metoder for å manipulere dem (settere og gettere). Lag noen metoder nå som kan brukes når du tester hele denne oppgaven. Når du senere skal bruke denne klassen i oblig 5 kan du gjerne legge inn flere metoder. (Hvis du finner det nyttig å legge til noen flere metoder i denne klassen underveis for å løse de senere oppgavene er det lov)

### **Oppgave 2.**

Lag en beholder som kan ta vare på mange HasMap-er (En liste (arraylist/linklist etc) med HashMaps) , der hver HashMap lagrer objekter av klassen SubSekvens. Nøkkelen i HasMap-ene er subsekvensene i objektene. Det du skal kunne gjøre med denne beholderen er:

- Sette inn en slik HashMap
- Ta ut en slik HashMap

- Spørre hvor mange HashMap-er den inneholder

Det er ingen krav til rekkefølgen du tar ut i.

### Oppgave 3.

I beholder-klassen du skrev i oppgave 2 skal du lage en statisk metode som slår sammen to HashMap-er som tar vare på SubSekvens-er. Parametrene til metoden er de to HashMap-ene som skal slås sammen, mens returverdien skal være en HashMap som er en sammenslåing (en fletting) av de to parametrene. Det flettede resultatet består av alle objektene i de to parametrene, men der subsekvensen /nøkkelen er den samme skal det bare lagres ett objekt i resultatet, men antallet i dette objektet skal være summen av antallene i de to objektene med samme nøkkel i parametrene.

### Oppgave 4.

Tilslutt skal du skrive et testprogram for å teste metodene i de foregående oppgavene. Bruk filene som er gitt (metadata.txt, fil1.txt, fil2.txt etc). metadata.txt inneholder navnen på de andre datafilene. Programmet ditt skal lese en og en datafil. Hver datafil representerer en person og hver linje en immunreseptor.

For hver person (datafil) skal du lage en HashMap med String og SubSekvens. Disse HashMapene skal så legges inn i Beholderen du lagde i oppgave 2. En subsekvens er en del av immunreseptoren, du bestemmer selv hvor stor en subsekvens skal være (om den skal være 3,4,5,6 etc) men det kan være lurt å lagre lengden i en konstant. Hvis vi har sekvensen ABABA og vi sier at subsekvensenes lengde skal være 3 får vi sekvensen ABA (de første 3 bokstavene), BAB(de tre midterste), og ABA(de tre siste). Som vi ser her dukker ABA opp 2 ganger. Da ønsker vi ikke å legge til ABA sin subsekvens i HashMapene en gang til, ~~men heller øke antall med én.~~

Når du er ferdig med å lese inn alle filene skal du ta ut to og to av HashMapene og flette dem sammen for å så putte den flettede HashMappen inn igjen i beholderen. Dette skal gjøres til det kun er 1 HashMap igjen (sagt med andre ord at alle hashMapene har blitt flettet sammen).

Tilslutt skal du skrive ut de subsekvensene som har f.eks. flere forekomster enn 5 (dette tallet kan du bestemme helt selv).