IN1010 V18, Obligatorisk oppgave 6

Innleveringsfrist: Onsdag 02.05. kl 10:00

Versjon 1.1. Sist modifisert av Silje Merethe Dahl og Kristine Jevne Berge.

Innledning

I denne oppgaven skal du lage et program med tråder som virker som produsenter og konsumenter. For å gjøre dette litt mer konkret, kan vi tenke oss at vi simulerer en etterretningssentral fra 1900-tallet som overvåker linjer hvor det sendes krypterte meldinger. Du skal bruke tråder og monitorer for å simulere forskjellige arbeidere.

Før du begynner med implementasjonen, burde du se grundig gjennom modellen og dokumentasjonen for prekoden. Forstår man dette, vil oppgaven gå mye enklere!

Modell

Etterretningssentralen har tre typer arbeidere: én operasjonsleder, og et bestemt antall telegrafister og kryptografer.

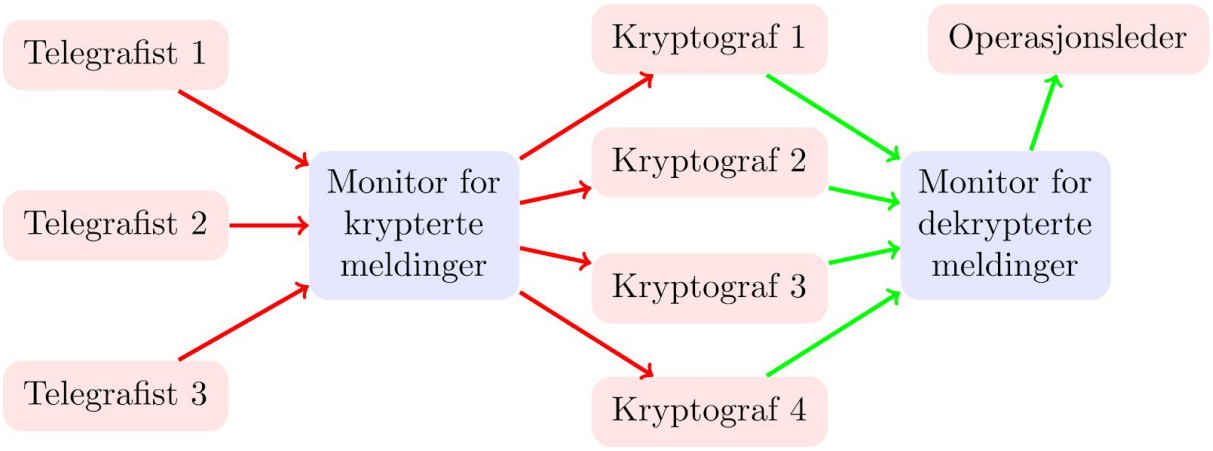
Hver telegrafist lytter til sin kanal som det kommer krypterte meldinger på. Når han mottar en melding, gir han den til monitoren. Han gjør dette til det ikke kommer flere meldinger (f.eks. når han mottar en melding som signaliserer SLUTT). Meldingene i monitoren er krypterte og må dekrypteres av kryptografene.

Hver kryptograf henter en kryptert melding fra monitoren, dekrypterer den, og sender den dekrypterte meldingen til en ny monitor. Han gjør dette til det ikke er flere krypterte meldinger, og det ikke kan komme flere (f.eks., når alle telegrafister er ferdige, så gir de en spesiell melding til monitoren som sier INGEN FLERE MELDINGER).

Operasjonslederen henter dekrypterte meldinger fra den andre monitoren. Han samler alle meldingene og sorterer dem basert på kanal og sekvensnummer. Når han vet at det ikke kan komme nye meldinger på denne monitoren (kryptografene sier fra på samme måte som telegrafistene ovenfor), så skriver han ut meldingene for hver kanal i sortert rekkefølge.

Hver person — telegrafistene, kryptografene og operasjonslederen — er en tråd. Alle trådene skal kjøre parallelt. Det gjør at flere kanaler kan lyttes til samtidig og flere meldinger kan dekrypteres samtidig, og programmet kjører fortere. Begge monitorene (én for krypterte, og én for dekrypterte meldinger) som må beskyttes med en lås for å unngå at flere tråder prøver å modifisere dem samtidig.

Figur 1: Flyten av meldinger i programmet. De røde og grønne pilene representerer henholdsvis overføringer av krypterte og dekrypterte meldinger. De røde boksene representerer tråd-objekter, og de blå boksene representerer monitor-objekter. NB: Dette er **ikke** en datastrukturtegning!



Prekode

Prekoden lastes ned som zip-fil. Denne må dere pakke ut, og plassere filene i samme mappe som du lagrer de andre klassene dine i. I zip-filen finner dere klassene "Kanal", "Kryptografi" og "Operasjonssentral". Full dokumentasjon av klassene finner dere her. I dokumentasjonen kan du se hvilke metoder som er tilgjengelig og hvordan konstruktøren for hver klasse ser ut. En liten hjelp på veien finner dere under.
NB: dere skal ikke gjøre endringer på prekoden!

Hovedprogram

I hovedprogrammet ditt vil du trenge å opprette en ny operasjonssentral og hente ut kanalene, dette gjør du slik:

```
Operasjonssentral ops = new Operasjonssentral(antallTelegrafister);
Kanal[] kanaler = ops.hentKanalArray();
```

Lytting på kanaler

Telegrafistene kan lytte på sin kanal ved å kalle på `kanal.lytt()`.

Dekryptering av meldinger

En `String kryptertMelding` kan dekrypteres slik:

```
String dekryptertMld = Kryptografi.dekrypter(kryptertMelding);
```

Legg merke til at metoden `dekrypter(String s)` er en statisk metode, så du trenger ikke å opprette en instans av Kryptografi for å kalle på metoden.

Del A: Monitorer og tråder

Melding

I tillegg til String-objektet som utgjør innholdet i meldingen, må hver melding ha et sekvensnummer og ID-en til kanalen meldingen kom fra. Denne informasjonen er nødvendig for å kunne skille meldingene etter kanal og sortere dem i riktig rekkefølge slik at meldingene fra hver kanal kan skrives ut sammen.

Telegrafister

Hver telegrafist(-tråd) har sin egen kanal. Telegrafistens oppgave er å lytte etter beskjeder (husk at `.lytt()` returnerer en `String`) på kanalen sin. Når en telegrafist får en beskjed skal den opprette en `Melding` og så sende meldingen videre til monitoren for krypterte meldinger, for deretter å gå tilbake til å lytte etter nye meldinger. Når det ikke er flere meldinger å hente, returnerer `.lytt()`-metoden "`null`".

Kryptografer

Kryptografтrådene sin oppgave er å motta meldinger fra monitoren for krypterte meldinger. Kryptografen skal så dekryptere meldingen og så sende den ferdig dekrypterte meldingen videre til monitoren for dekrypterte meldinger. Merk at kryptografene ikke trenger å ta hensyn til sekvensnummer eller kanal-ID når de henter ut meldingene.

Monitorer

Vi skal ha to monitorer i dette systemet, én monitor som mottar krypterte meldinger og sender meldingene videre til kryptografer, og én monitor som tar i mot dekrypterte meldinger og sender meldingene videre til operasjonsleder når alle meldinger er ferdig dekryptert. For å holde på meldingene i monitoren bør du bruke en passende beholder, for eksempel ArrayList eller LinkedList. Alternativt kan du benytte en beholder fra oblig 3 (vi vil oppfordre til å benytte ArrayList eller LinkedList, da det kan være nyttig å bli bedre kjent med Java-biblioteket). Pass på at ingen meldinger blir liggende for lenge i monitoren før de blir hentet av en kryptograf.

Husk å synkronisere trådene dine, med andre ord: når en tråd har fått tilgang til en monitor, må alle andre trådene vente til den første tråden er ferdig.

Avslutning av trådene

Når alle telegrafistene er ferdig, og det ikke er flere meldinger å hente, må de på en eller annen måte signalisere til monitoren at de ikke kommer til å sende mer. Hvis monitoren ikke lenger får inn noen nye meldinger, skal kryptografene avslutte når monitoren er tom.

Relevante Trix-oppgaver: Alle oppgaver om tråder, men spesielt 11.04.

Del B: Skrive til fil

Operasjonsleder

Når alle telegrafistene og kryptografene er ferdige, skal operasjonslederen skrive meldingene til fil, med en fil for hver kanal. For å kunne sortere meldingene kan det være nyttig å legge til noe i Melding-klassen. Det kan også være lurt at operasjonslederen har en beholder(e) for å holde styr på og sortere meldingene. Når meldingene skal skrives til fil, skal hver melding skal være adskilt av to linjeskift. Pass på at meldingene kommer i riktig rekkefølge!

Relevant Trix-oppgave: 9.01

Hovedprogram

I hovedprogrammet ditt skal du opprette en Operasjonssentral, monitorene, telegrafister, kryptografer og operasjonslederen. Deretter må telegrafistene, kryptografene og operasjonslederen settes i arbeid. Hvor mange telegrafister og kryptografer du har, spiller ingen rolle, så lenge det er et positivt tall, men husk at det skal være like mange telegrafister som det er kanaler (hver telegrafist har sin egen kanal).

Tips

Det kan være lurt å løse denne oppgaven etappevis.

1. Telegrafist (lytting)
2. Telegrafist (levere kryptert melding) og Kryptograf (hente og dekryptere melding)
3. Kryptograf (levere dekryptert melding) og Operasjonsleder (hente og organisere dekrypterte meldinger)

Merk

Meldingene du mottar er kodet i UTF-8, så bruk UTF-8 når du skriver meldingene til fil. Du kan gjøre som følger for å opprette en `PrintWriter` som skriver til `File utfil` med UTF-8-koding:

```
new PrintWriter(utfil, "utf-8");
```

Du vil se om programmet ditt virker som det skal ved at tekstene som skrives til fil er leselige og rekkefølgen på avsnittene virker rimelig. Du bør også sjekke at meldingene er ordnet med sekvensnumrene i stigende rekkefølge.
Hvis du synes programmet går sakte, kan du øke antallet kryptografer. Med 3 telegrafister og 5 kryptografer tar det ca. 40 sekunder. Med 20 kryptografer tar det ca. 15 sekunder.

Oppsummering

Du skal levere alle klasser som skal til for at hovedprogrammet skal fungere. Dere trenger ikke å levere inn prekoden.

Alle delene av programmet må kompilere og kjøre på Ifi-maskiner for å kunne få oppgaven godkjent. Unngå bruk av packages (spesielt relevant ved bruk av IDE-er som IntelliJ). *Ikke lever zip-filer!* Det går an å laste opp flere filer samtidig i Devilry.

Merk: Resten av oppgaven er frivillig (men anbefalt).

Del C: Valgfri del

Endre monitoren din slik at du benytter samme monitor-klasse til å både motta krypterte meldinger fra telegrafister (første monitoren), og til å ta imot dekrypterte meldinger fra kryptografene.

Lykke til!