Plenum: Tråder 2

Ahmed Waseem Saeed

April 2024

Introduksjon

Velkommen til ny plenumstime! Tråder er i fokus denne uka også, og vi skal videreføre konsepter fra forrige uke, samt bygge på med nye. Dagens eksempel er mer sammensatt enn forrige uke og vi skal modellere pakker som sendes via postterminaler og utleveringssteder. Videre skal vi lage fabrikker som produserer pakker, og postbud som både henter varer og produserer pakker som leveres til mottakere (kunder). Dette innebærer å ha to monitorer, der en klasse også både produserer og konsumerer en ressurs. Vi skal se på en mer detaljert beskrivelse i de neste seksjonene. Les hele oppgaveteksten før du begynner.

Vare

Skriv klassen Vare som har et navn, en kategori og fabrikken som laget varen. Den skal også ha en variabel bestiltAv (String) som markerer hvem bestilte varen. Overskriv også toString-metoden, slik at den har med relevant informasjon for en vare.

Postterminal

Klassen Postterminal er den første av to monitorer. Monitoren har en liste over varer. Klassen får inn varer av fabrikker (produsenter), som blir gjort om til pakker av postbud (konsumenter). Du kan anse klassen som en terminal som får inn varer og gjør de klare til utsending. Klassen må ha en lås som brukes når vi deler varelista. Vi har en maks-grense på 20 varer inne i terminalen til enhver tid. Vi trenger altså to Conditions, ikkeTom og ikkeFull som håndterer ressursene våre.

```java
/*
 * Metoden skal produsere varer ved å
 * legge til varen i varelista.
*/
public void leverVare(Vare vare);


/*
 * Metoden skal la et postbud hente en
 * pakke. Fjern et element
 * i varelista og returner et Pakke-objekt.
*/
public Pakke hentPakke(Postbud postbud);
```

Pakke

En pakke er ganske lik en Vare, den har en referanse til et Vare-objekt i tillegg til en Postbud-referanse som er postbudet som er ansvarlig for pakken.

Overskriv toString()-metoden slik at du får frem informasjonen til en Pakke på en ryddig måte.

Fabrikk

Klassen Fabrikk har en referanse til en Postterminal, et navn, samt en liste over varer som produseres av fabrikken. Konstruktøren skal ta dette inn og sette disse verdiene, og lese inn varer fra en fil, og legge de til i listen. Filnavnet tas også inn i konstruktøren. Metoden lesFraFil() skal kalles i konstruktøren.

Fabrikk-klassene skal opptre som tråder som produserer varer til Postterminalen vår. run-metoden skal altså levere varer fra lista (en og en) til postterminalen. For hver vare som leveres skal tråden sove i 1000 ms (tilsvarer ett sekund).

Når Fabrikkene er ferdig med å lage varer, må vi varsle postterminalen om dette. Her står du fritt frem for å håndtere dette. Et tips er å se tilnærmingen fra forrige ukes plenumstime. Husk at når fabrikkene er ferdig med å lage varer, så kan det fortsatt eksistere pakker i terminalen som må hentes, og dette må håndteres.

Innlesing fra fil

Den private metoden lesFraFil(String filnavn) skal lese inn varer fra en fil, der hver linje er en vare. Dataen er separert med komma og er på formatet <navn,kategori,bestiltAv>. Opprett Vare-objekter og sett de inn i lista over varer. Se eksempelfiler på plenumssiden.

Postbud

Et Postbud har et navn og et selskap vedkommende jobber for (String), i tillegg til en referanse til et Postterminal-objekt og et Utleveringssted-objekt (opprett en tom klasse inntil videre). Klassen implementerer Runnable og run()-metoden skal hente pakker så lenge det finnes pakker i Postterminalen (du må håndtere situasjonen når Postterminalen er tom). Legg hver pakke i en egen liste i Postbudklassen. Skriv også metoden *skrivPakkeInnhold()* som skriver ut innholdet i lista.

Postbudet må hvile litt (sove i 100 ms) etter hver pakke, før neste pakke håndteres.

Testing på nåværende punkt

På nåværende stadie har vi muligheten til å teste om vi klarer å legge inn varer og hente de ut fra postterminalen vår. Opprett et hovedprogram der du lager tre Fabrikk-tråder med vilkårlige navn som tar inn ulike filnavn (en fil per Fabrikk), der du bruker filene fra plenumssiden.

Deretter lager du fire Postbud-tråder og setter disse i gang også. Lag også et en barriere ved hjelp av en CountDownLatch som initialiseres med antall tråder vi skal lage. Modifiser Fabrikk- og Postbud-klassen slik at de tar inn en barriere hver og teller ned når de er ferdig.

Vent til alle trådene er ferdig, og skriv ut en melding som indikerer at fabrikkene er ferdig, og kall på skrivPakkeInnhold() til hvert postbud.

Utleveringssted

Vi skal nå skrive klassen Utleveringssted. Klassen har et navn samt en liste over pakkene levert hos seg, og opptrer som en monitor. Her vi trenger vi kun en lås. Vi skal sørge for at alle postbud har levert varer, før noen kunder begynner å hente dem (altså at produsentene gjør seg ferdig før konsumentene konsumerer).

Metoden *leverPakke(Pakke pakke)* skal levere en pakke og sette den inn i lista over pakker. *hentPakke(Kunde kunde)* skal gå gjennom lista med pakker og hente den første pakken i lista som matcher med kundens navn. Husk å fjerne pakken fra lista. *Hvordan fjerne noe fra en liste når vi itererer over den?*

Endre run()-metoden i Postbud

Modifiser run()-metoden slik at budet også leverer en gitt pakke til utleveringsstedet.

Kunde

Klassen Kunde har et navn. Klassen opererer som tråder som konsumerer pakker fra et utleveringssted. Vi trenger derfor også en referanse til et Utleveringssted-objekt. Klassen skal fortsette å konsumere pakker til det ikke er flere pakker å konsumere. For hver pakke skal en kunde slappe av i 1000 ms (Thread.sleep()). Skriv ut hver pakke en kunde henter.

Utvide hovedprogrammet

Opprett fem kunder med følgende navn: Ahmed, Andreas, Annika, Bushra, og Samuel. Før du starter alle kundetrådene, så sørger du for at alle postbudtrådene er ferdig og at alle pakker er levert.

Deretter kall .join() på hver kundetråd og skriv til slutt ut at "Nå er programmet ferdig, kundene har fått alle varene sine".