



UiT Norges arktiske universitet

Kort hjemmeeksamen i:	INF-1400 Objektorientert Programmering
Dato:	2021-09-27
Tidspunkt:	09:00 – 13:15
Kursansvarlig:	John Markus Bjørndalen Edvard Pedersen Henrik Hillestad Løvold
Antall sider:	3
Support:	Du kan ringe 776 20 880 for support på eksamensdagen.
Vekting av spørsmål, eller annen informasjon:	
Viktig informasjon om sitering og plagiering:	<ol style="list-style-type: none">1. Dette er en individuell eksamen som skal besvares uten samarbeid med andre.2. Alle hjelpemidler er tillatt (egne notater, pdf'er fra forelesningene, lærebok, internett etc).3. Alle eksamener som leveres i WISEflow blir automatisk sjekket for plagiat. Det er ikke tillatt å kopiere medstudenter, netressurser, kilder, eller litteratur uten referanser.

Praktisk informasjon

- **Dere kan levere pseudokode.** Det er ikke behov for å lage en kjørende løsning. Vi er ute etter å se mønster og tenking. Prøver dere å lage en kjørende løsning kommer dere raskt til å bruke mer tid enn det vi krever.
- Ikke overdriv kompleksiteten i løsningen (f.eks. modellere ting som ikke er beskrevet i oppgaven). Vi skal ikke lage en komplett løsning på 3 timer. Det betyr at noen ting kan være litt urealistiske. Fokuser på hvordan du skal jobbe med koden og designet på programmet.
- Vi vil være tilgjengelige på telefon og Discord hvis det er spørsmål under eksamen.
- Det er lov å stille spørsmål hvis det er noe dere ikke forstår i den utleverte koden.

Bakgrunn

Endelig skal Nordnorgebanen settes i drift! Som et tiltak for å spare penger, blir det énsپoreт jernbane hele veien, utenom ved noen stasjoner underveis med to spor der tog fra nord og tog fra sør kan krysse.

På banen vil det gå to typer tog; passasjertog og godstog. Banen er inndelt i seksjoner, hvor hver seksjon separeres med en stasjon der det er mulig å krysse. Når et tog kommer til en stasjon må det vente dersom det er motgående tog på neste seksjon. Flere tog kan bevege seg innen samme seksjon om de beveger seg i samme retning.

Dersom to tog ønsker å kjøre i samme retning når seksjonen fremfor dem blir ledig, skal alltid passasjertog ha prioritet og få lov til å kjøre først.

I denne oppgaven skal vi bygge opp en simulering av følgende jernbane:

- Det er fem seksjoner, separert av fire stasjoner med dobbeltspor.
- Ved starten av programmet, starter et tilfeldig antall (mellom 2 og 8) tog av tilfeldig type fra de ulike stasjonene. Togene beveger seg i tilfeldig retning (nord/sør).
- Programmet er ferdig når alle tog har nådd sin endestasjon.

Oppgave 1

Lag et klassediagram for de klassene du tenker er naturlige å ta med, inkludert relasjoner mellom klassene. Løsningen skal demonstrere arv og polymorfi¹.

Oppgave 2

Beskriv hvordan oppførselene kan implementeres og grove trekk for hvordan du endrer den eksisterende koden. Beskriv antagelsene du gjør. Forklar hvordan arv og polymorfi brukes i løsningen.

Oppgave 3

Implementer det du har skissert (pseudokode holder, bruk i alle fall ikke tiden din på å debugge småfeil).

¹ Hvis du mener dette gir en klønete løsning kan du gjerne kommentere dette og hvordan du ellers ville gjort det som et tillegg. Målet med oppgaven er å gi deg en mulighet til å demonstrere arv og polymorfi.

Oppgave 4

Vi ønsker å legge til vedlikeholdstog som kjører på banen og inspiserer skinnegangene. Vi ønsker at denne inspeksjonen ikke skal forstyrre den vanlige trafikken i så, slik at vi slipper buss for tog. Dette vedlikeholdstoget må passere alle deler av skinnegangen, inkludert begge sider av stasjonene hvor det kan passeres.

- 1) Vil du plassere denne i klassehierarkiet, og i så fall hvor?
- 2) Hvilke nye attributter og metoder behøver denne klassen, sammenliknet med tog-klasser i hierarkiet ditt?
- 3) Implementer klassen (pseudokode holder her også).