

Prog1 prosjektrapport

Ole-Jørgen Andersen HiØ

Prosjektet mitt ble å lage et program i python, som henter trafikkdata fra flysimulator nettverket vatsim og viser avganger og ankomster ved en gitt flyplass, da jeg ville jobbe med ekstern data.

Planen ved starten var å strukturere koden på en slik måte at jeg kunne gjenbruke store deler av koden (DRY), samtidig som den var oversiktlig og lett leselig. Dette gjorde jeg delvis ved å strukturere store deler av koden i klasser. F.eks, for å lage de to tabellene for avganger og ankomster, lagde jeg en generell klasse **Table**, som kan brukes til å lage slike tabeller og skrive de ut til konsollen. Et annet eksempel er **Style** klassen, som blant annet holder på metoder for å endre fargen på teksten i konsollen. Disse metodene brukes flere ganger i prosjektkoden.

En annen faktor som gjør koden mer oversiktlig er, at jeg valgte å dele opp koden i moduler. F.eks. modulen "vatsim" holder på **Vatsim** klassen osv. Disse modulene blir så importert i "main" filen, som er hovedskriptet og holder på koden for konsoll dialogen, her kommer koden fra alle modulene sammen og blir brukt. Men dette er ikke det eneste stedet man bruker kode fra en annen modul. Det fine med moduler er at de kan gjenbrukes overalt. F.eks. bruker jeg Style modulen i Table modulen for å farge tekst i tabellen.

Et aspekt som gjorde prosjektet vanskelig var, at jeg ikke begynte å kommentere koden med en gang. Dette medførte til at jeg ofte mistet oversikt over hva koden gjorde, i tillegg til at jeg brukte langt lengre tid på å kode. Så min anbefaling til andre studenter er, kommenter med en gang!

Et annet aspekt som gjorde prosjektet vanskelig var, at jeg ville skrive gjenbrukbar kode, da dette kan kreve litt ekstra planlegging på forhånd. F.eks. planla jeg på forhånd å lage en felles metode for å farge konsoll teksten, fordi jeg kom til å få bruk for det flere ganger i prosjektkoden. Det å planlegge slik på forhånd var litt krevende, så det jeg kan anbefale er å bare begynne å skrive også eller slå sammen liknende kode biter til felles metoder senere.

En annen faktor som gjorde spesielt planleggingsfasen vanskelig var, at jeg ville holde antall forespørsler om vatsim data til et minimum, da disse serverne er svært trafikkert. Dette

krevde en del ekstra planlegging, men var vel verdt det når jeg fikk det til! Jeg landet på en dobbel løsning, hvor dataene var lagret i klassen mens programmet kjørte, i tillegg til at det var lagret i en json fil. Jeg valgte å lagre i en json fil i tillegg i tilfelle brukeren skulle restarte programmet mens dataene fortsatt var ferske, og delevis siden det var et krav å lese og skrive til filer. Dette var en fin løsning, da jeg vil tørre å påstå at det er mer effektivt å hente data som er lagret i en variabel, enn lese data fra en fil hver gang.

Mens programmet kjører, er dataene kun sjekket og eventuelt hentet fra vatsim, hvis en `get` metode i `Vatsim` klassen er kalt, f.eks. når brukeren skriver inn icao koden til flyplassen han/hun ønsker å se trafikk ved. Nye data blir forespurt kun hvis dataene er eldre enn oppdaterings intervallet på vatsim serverne.

Viktig info om programmet:

- **Bruk Python 3.8 eller nyere.**
- **Bruk en konsoll som støtter ANSI fargekoder.**
- **Programmet bruker kun Python Standard Library.**
- `Vatsim` klassen har en metode `__fail_safe()`, som henter en lokal kopi av vatsim dataene hvis data serverne skulle være utilgjengelige når sensor retter prosjektet. Metoden kjører automatisk hvis en HTTP / URL exception i `__fetch_new_data()` blir reist.
- `Vatsim` klassen har parametere for å aktivere `local_mode`, som henter data fra den lokale kopien av vatsim dataene, og `debug_mode`, som aktiverer utskrift av debug meldinger i konsollene. Disse parameterne er laget for å gjøre utviklingen av prosjektet lettere.
- Error meldinger blir alltid skrevet ut, uavhengig av `debug_mode`.