Oppgave 4: Planlegging av turer fra hytte til hytte (40 poeng)

Denne oppgaven består av en rekke deloppgaver der du skal implementere komponenter av et større system. Om du hopper over en deloppgave er det likevel viktig at du leser hele denne teksten (den ligger samlet som en pdf-fil vedlagt hver deloppgave). I hver av deloppgavene kan du bruke klassene og metodene fra tidligere deloppgaver, også om du ikke har programmert dem.

Du skal implementere deler av et system som organiserer en rekke turer. Hver tur går over flere dager med overnatting på en ny hytte hver dag. Den delen av systemet du trenger å kjenne til består av klassene **Hytte**, **Tur** og **Turplanlegger**. Det anbefales å lage en tegning av strukturen for eget bruk.

Klassen **Turplanlegger** holder rede på alle hytter i systemet ved hjelp av en ordbok (Dictionary) med hyttenavnet som nøkkel og referanse til hytta som verdi. Klassen **Turplanlegger** har også en liste med referanser til turer.

I klassen **Tur** finnes det en tekst (som kan inneholde blanke, men ikke linjeskift) som beskriver turen, og en liste med referanser til de hyttene som besøkes i løpet av turen (som alle ligger i **Turplanleggers** ordbok).

Hvert objekt av klassen **Hytte** har et unikt navn, et antall sengeplasser og prisen for overnatting per seng i denne hytta (alltid lik pris for hver seng i en hytte).

Oppgave a) (8 poeng)

Skriv klassen **Hytte.** Konstruktøren skal ha parametere for instansvariablene navn, antall senger og pris for overnatting. Videre skal klassen ha følgende metoder:

hentNavn som returnerer hyttenavnet.

totPris som returnerer prisen for et antall personer (antallet er parameter til metoden) for én natt.

sjekkPlass som returnerer **True** om hytta har nok senger til et antall personer (parameter til metoden), ellers **False**.

__str__ som returnerer en bruker-vennlig streng med hyttas navn, antall senger og pris per seng.

__eq__ som sammenligner objektet den kalles på med et annet objekt (referansen er parameter til metoden). To **Hytte**-objekter er like hvis de har samme navn, da returneres **True**, ellers **False**.

Oppgave b) 10 poeng

Skriv klassen **Tur** med en konstruktør som tar imot en liste med referanser til hytter, og en (en-linjes) tekst som beskriver turen. Videre skal klassen ha følgende metoder:

skrivTur som skriver ut på terminal beskrivelsen av turen, og deretter informasjon om hver hytte turen går innom.

sjekkPrisPlass som går gjennom alle hyttene på turen og sjekker om det er nok senger for et antall personer på hver av hyttene, og om totalprisen for turen (alle personer på alle hyttene) er under et maksbeløp. Antall personer og maksbeløp er parametere til metoden. Metoden returnerer **True** om det er nok plasser og kostnadene blir under maksbeløpet, ellers **False**.

Oppgave c) 10 poeng

Du skal skrive klassen **Turplanlegger** med konstruktør og metodene **_hytterFraFil** og **_turerFraFil**. Metodene kalles fra konstruktøren og leser inn data til henholdsvis ordboken med hytter, og listen med turer. Filnavnene er parametere til konstruktøren og til hver sin metode.

Metoden **_hytterFraFil** åpner og leser fra filnavnet gitt i parameter til metoden. Filen har et ukjent antall linjer på følgende format (to hytter vist)

```
Hyttenavn antSenger pris
Hyttenavn antSenger pris
:
```

der første streng skal leses som en tekst og de to siste strengene som heltall og flyttall. Hver hytte representeres i et objekt av klassen Hytte, som lagres i en ordbok i med hyttenavnene som nøkler og referanse til objektet som verdier. Denne ordboken returneres fra metoden når filen er ferdig lest.

Metoden **_turerFraFil** åpner og leser all informasjon om alle turene i systemet fra filen med navn oppgitt i parameteren, og oppretter og returnerer en liste over **Tur**-objekter. Denne filen inneholder to linjer per tur. Første linje inneholder turbeskrivelsen (typisk flere ord), andre linje inneholder navnene på alle hytter på turen. Ingen hyttenavn inneholder mellomrom (space). Samme hytte kan inngå i flere turer. Filen har følgende format:

```
En tekst som beskriver en tur
Hyttenavn1 Hyttenavn2 Hyttenavn3
En tekst som beskriver en annen tur
Hyttenavn1 Hyttenavn4
:
```

Oppgave d) 6 poeng

Skriv en metode **finnTurer** i klassen **Turplanlegger** som kan brukes til å identifisere turer som har plass til et gitt antall personer på alle hyttene, der totalkostnad for alle personer alle dager er under et maksimalbeløp, og som ikke varer mer enn et maks antall dager (turene går alltid til en ny hytte hver dag). Metoden skal gå gjennom alle turer, og skrive ut på terminalen all informasjon om de turene som tilfredsstiller kravene. Antall personer, maksbeløp og maksimalt antall dager er parametere til metoden. Du kan bruke metoder fra tidligere oppgaver uansett om du har skrevet dem eller ikke.

Oppgave e) 6 poeng

Skriv et testprogram for klassen **Turplanlegger** som gjør følgende:

- Oppretter et objekt av klassen Turplanlegger som leser inn data fra filene *hytter.txt* og *turer.txt*
- Skriver ut all informasjon om alle turer som tilfredsstiller følgende krav:
    - ikke varer mer enn 5 dager
    - har overnattingsplass på alle hyttene for et følge på 7 personer, ..
    - ..med et overnattingsbudsjett på til sammen 7000,- for alle 7, hele turen.