UNIVERSITETET I TROMSØ UiT
FAKULTET FOR NATURVITENSKAP OG TEKNOLOGI

EKSAMENSOPPGAVE

Eksamen i	:	INF-1400 Objektorientert programmering
Dato	:	Mandag 27. mai 2013
Tid	:	0900–1300
Sted	:	Åsgårdvegen 9
Tillatte hjelpemidler	:	Ingen

Oppgavesettet er på 5 sider inklusiv forside

Kontaktperson under eksamen: Anders Andersen, 951 80675

Del A

Ola og Kari er begge opptatt av kart og turer, og de ønsker å ta vare på informasjon om hvor og når de har vært på tur. For å gjøre dette enklere så har de kjøpt en GPS-klokke som kan spore turene deres. Det vil si at mens de er på tur så vil klokken med jevne mellomrom lagre deres nåværende *posisjon*. Når de er ferdig så kan de laste over disse posisjonene til en PC. Samlet gir alle posisjonene på en tur *et spor* (en liste av posisjoner). En posisjon inneholder *lengdegrad*, *breddegrad*, *meter over havet* og *tidspunkt* (dato og klokkeslett). Ola og Kari bestemmer seg for å lage et program for å håndtere disse dataene. De starter med å identifisere følgende klasser:

- `Posisjon` – En gitt posisjon på en tur med attributtene `lengdegrad`, `breddegrad`, `moh` (meter over havet) og `tidspunkt`.
- `Spor` – Et spor for en tur som inneholder følgende attributter:
  - Liste over posisjoner (objekter av klassen Posisjon)
  - starttidspunkt for dette sporet (tidspunkt for første posisjon)
  - stopptidspunkt for dette sporet (tidspunkt for siste posisjon)
- `Tur` – En tur med attributtene `spor` (et Spor objekt), `navn` og `beskrivelse`.

De bestemmer seg også for å spesialisere klassen tur for de turene som er *sightseeing*. Denne klassen `SightseeTur` skiller seg fra klassen `Tur` med at den også har en liste over spesielt interessante posisjoner (POI – point of interest) i løpet av en tur. Hvert enkelt element i denne listen referer til et element i sporet til denne turen (se attributtet `spor` i klassen `Tur`).

Oppgave 1 – 20%

Vi skal foreløpig konsentrere oss om attributtene (og ikke metodene) i de klassene vi jobber med. Vi kommer tilbake til metodene senere.

a) Klassen `SightseeTur` kan både realiseres med arv (inheritance) og komposisjon (composition). Forklar forskjellen på arv og komposisjon og hva vi mener med en *is-a* (er-en) og en *has-a* (har-en) relasjon. Beskriv og begrunn også ditt valg av arv eller komposisjon i klassen `SightseeTur`.

b) Tegn opp klassediagrammene for de fire klassene `Posisjon`, `Spor`, `Tur` og `SightseeTur`.

c) Tegn opp datastrukturene for *en* sightsee tur med et kort spor (noen få posisjoner) og to spesielt interessante posisjoner.

Oppgave 2 – 25%

Implementer klassene vi har spesisfisert til nå. Du trenger ikke å ta med andre metoder enn `__init__()` siden vi skal jobbe videre med klassene senere. Argumenetene til `__init__()` metoden til de ulike klassene er som følger:

- `Posisjon`: `lengdegrad`, `breddegrad`, `moh`, `tidspunkt`
- `Spor`: en liste med *dictionaries*, hvor hver dictionary har følgende nøkler med verdier:
  - `"lengdegrad"`
  - `"breddegrad"`
  - `"moh"`
  - `"tidspunkt"`

  Eksempel på en slik Python liste med *et* element (tidspunkt er antall sekunder siden Epoch, 1. januar 1970 klokken 00.00.00 UTC):

  ```
  [{"lengdegrad": 18.96778529, "breddegrad": 69.68299052,
    "moh": 64.4, "tidspunkt": 1369651934}]
  ```

- `Tur`: `spor`, `navn`, `beskrivelse` (`beskrivelse` kan være valgfri)
- `SightseeTur`: Samme som `Tur`

Oppgave 3 – 20%

Vi skal nå lage to metoder. Den første metoden returnerer en referanse eller indeks til den posisjonen i et spor som er nærmest tidspunktet angitt som argument:

```
finnPosisjon(self, tidspunkt):
    ...
```

Den andre metoden legger til en ny interessant posisjon i forbindelse med sightseeing. Også her er tidspunktet angitt som argument. Tidspunktet er det tidspunktet man var ved eller på den interessante posisjonen.

```
registrerPOI(self, tidspunkt):
    ...
```

Angi hvilke klasser du vil plassere disse metodene i og skriv koden for metodene.

(Hint: forsøk å utnytte funksjonalitet du allerede har skrevet kode for når det er mulig.)

Del B

Kari og Ola ønsker nå å få plottet turene sine på kart. De har fått tak i kode for dette, men denne koden er ikke helt ferdig og den må tilpasses deres klasser og objekter. De har startet med å lage klassen `KartTur`, men de har ikke gjort ferdig metoden `draw()`. I koden nedenfor er `spor1` en instans av klassen `Spor` for dagens tur i Tromsdalen.

```
class KartTur(Tur):
    def __init__(self, spor, navn, beskrivelse=''):
        Tur.__init__(self, spor, navn, beskrivelse)
    def draw(self):
        pass

# Last kart-bakgrunn fra bilde og skaler ned
scaleFactor = 7
bgimage = pygame.image.load('map1.jpg')
xsize, ysize = bgimage.get_size()
res_x = xsize/scaleFactor
res_y = ysize/scaleFactor
bgimage = pygame.transform.scale(bgimage, (res_x, res_y))

# Tegn kart i bakgrunnen
pygame.init()
screen = pygame.display.set_mode((res_x, res_y), 0, 32)
background = bgimage.convert()
screen.blit(background, (0,0))

# Tegn inn tur
turPaKart = KartTur(spor1, 'Tromsdalen 2013-05-27')
turPaKart.draw()

# Vis kart med tur på skjermen
pygame.display.update()
```

For å få tegnet turen inn på kartet må hver enkelt posisjon transformeres til en (x,y) koordinat på skjermen (`screen`). For å få til det har vi klassen `KartTrans` med metoden `transformer()`. Nedenfor ser vi hvordan den kan brukes for å transformere lengdegrad og breddegrad til posisjon på skjermen i eksemplet:

```
transformMap = KartTrans(res_x, res_y, 18.99021854, 69.68469448, 2142)
(x, y) = transformMap.transformer(18.96778529, 69.68299052)
```

De to første argumentene, `res_x` og `res_y`, angir størrelsen (i pixler) på området vi kan tegne i. De to neste argumentene angir posisjonen i lengdegrad og breddegrad for hjørnet av skjermen, altså for posisjon (0,0). Det siste argumentet angir målestokken til kartet. Det er utenfor oppgaven å lage koden for `KartTrans`, Dere vil kunne bruke denne instansen av klassen for å transformere lengdegrad og breddegrad i posisjoner til (x,y) koordinater på skjermen.

Oppgave 4 – 25%

a) Skriv ferdig koden for metoden `draw()` i klassen `KartTur`. Metoden `draw()` må tegne en strek som følger sporet for den gitte turen. For å tegne en linje mellom 2 punkter kan du bruke `pygame.draw.line()`:

```
pygame.draw.line(screen, color, (x1, y1), (x2, y2))
```

b) I klassen `KartTur` kan vi fjerne koden for metoden `__init__()`. Forklar hvorfor det er mulig og hva som da skjer når vi lager en instans av denne klassen (se linje 27 i koden på forrige side).

Oppgave 5– 10%

Gi en beskrivelse av 2 av de følgende uttrykk/konsepter:

a) Objekt (object)

b) Klasse (class)

c) Mutable/immutable

d) Instans (instance)

e) Metode