

Høsten 2022 – Tentamen IT 2 del 2

Oppgavene på denne delen løses med programmering. Kommunikasjon er ikke tillatt. Skriv oversiktlig og ryddig kode. Bruk kommentarer der det egner seg, blant annet til å oppgi kilder. Løs oppgaven på en måte som gjør at du får vist din kompetanse. Dersom du ikke får til å løse oppgaven på måten det står beskrevet, kan alternativ måte gi noe uttelling.

Besvarelsen leveres på Teams i form av en .zip fil. Pass på at alle filer som trengs for å kjøre koden leveres.

Oppgave 1

Sr-89 er en radioaktiv isotop av grunnstoffet strontium. Den har en halveringstid $H = 51$ dager. Dette er hvor lang tid det tar før halvparten av mengden Sr-89 er omdannet.

Hvor mye av isotopen som gjenstår er gitt ved følgende funksjon:

$$N(t) = N_0 \cdot \left(\frac{1}{2}\right)^{t/H}$$

Der $N(t)$ er mengden Sr-89 som er igjen i gram etter tida t i dager. N_0 er mengden av stoffet ved start ($t = 0$). La $N_0 = 50$ g.

- Lag to arrays **t_array** og **N_array** av lengde 1001. **t_array** skal være fylt med tidsverdier i intervallet $[0, 3H]$, altså fra og med 0 til og med 3 ganger halveringstiden. **N_array** skal være fylt med korresponderende $N(t)$ verdier.
- Plott grafen til halveringskurven. Husk aksetitler.

Bonusoppgave (gjør denne til slutt hvis du har tid)

Et primtall er et naturlig tall større enn 1, som bare er delelig med seg selv og 1.

Summen av primtallene lavere enn 10 er $2 + 3 + 5 + 7 = 17$.

Lag et program som finner summen av primtallene lavere enn 100 000.

Oppgave 2

- a) Lag en klasse «Person». En person skal ha et fornavn, etternavn og fødselsdato.
- b) Opprett et Person-objekt med et valgfritt fornavn, etternavn og fødselsdato.
- c) Lag en metode i Person-klassen som regner ut alderen til personen, gitt dagens dato. Test denne metoden med person-objekt du lagde i b).
- d) Legg til en metode i Person-klassen som presenterer informasjon i person-objektet i form av en utskrift.
- e) Lag en klasse Elev som arver fra klassen Person. En elev skal i tillegg til fornavn, etternavn og fødselsdato ha en liste med fag man har dette skoleåret. Opprett et Elev-objekt med ditt eget fornavn, etternavn og fødselsdato, i tillegg til dine egne fag du har dette skoleåret.
- f) Utvid metoden fra d) til å oppgi fagene til eleven i utskriften. Test den ved å bruke ditt eget Elev-objekt.
- g) Lag en funksjon som sammenligner alderen til to personer. Programmet skal oppgi hvem som er eldst, evt. om personene er like gamle. Lag passende utskrift(er). Test denne funksjonen med to selvvalgte Person-objekter.

Vi ønsker å lage mange nye person-objekter. Til dette skal du bruke API-en randomuser (<https://randomuser.me/>). Dette er en API som lager tilfeldig brukerdata. Man kan hente informasjon om en tilfeldig person ved å gjøre en spørring til følgende URL:

<https://randomuser.me/api/>

- h) Skriv kode som henter fornavn, etternavn og fødselsdato fra API-en. Fornavn og etternavn ser dere i «name», og fødselsdato i «dob» (date of birth). Lag et person-objekt ut fra disse opplysningene. Test programmet ved å kalle metoden du lagde i deloppgave d).
- i) Lag et program som henter 50 tilfeldige genererte personer fra API-en. Det skal lages Person-objekter for hver av disse, og objektene skal legges i en liste. Tips: Det kan ta tid å gjøre 50 individuelle API-spørringer. I dokumentasjonen står det forklart hvordan man kan hente flere tilfeldige brukere med én spørring.
- j) Skriv kode som lager en csv eller txt fil. Hver person i listen fra deloppgave i) skal skrives til denne fila; en person for hver rad. Første linje i fila skal være en passende overskrift, slik det er tydelig hva hver kolonne betyr.