

INF1001 - Obligatorisk innlevering 7 - Hangman

Frist: 7.11.16 12:00

1 Introduksjon

Du skal i denne innleveringen skrive et objektorientert Hangman-spill. Spillet skal ha støtte for å lese inn en ordliste fra en fil og å velge et tilfeldig ord fra denne ordlisten. Man skal kunne spille flere spill etter hverandre og få en poengsum. Du skal kunne ta vare på denne poengsummen og vise en high-score-liste.

Når programmet starter skal man få opp en meny, her skal du kunne velge mellom å starte et spill, vise high-scores (høyeste poengsummer) eller å avslutte. Når du så starter et spill skal du få lov til å forsøke å løse en Hangman-oppgave. Hvis du klarer oppgaven skal du få lov til å spille videre helt til du ikke klarer en oppgave.

1.1 Spilleets regler

I ett spill (merk at det er forskjell på ett spill og en spillsekvens!) med Hangman skal det gis spilleren ti forsøk på å komme frem til riktig ord, hvor spilleren skal gjette en og en bokstav. Man mister kun et forsøk hvis bokstaven ikke er riktig, hvis man forsøker å skrive inn en bokstav som allerede er forsøkt skal dette heller ikke telle som et forsøk. Hvis en bokstavgjetning er riktig så skal alle forekomster av denne bokstaven bli fylt ut. Det skal ikke gjøres forskjell på store og små bokstaver. Et spill er ferdig enten når alle forsøk er oppbrukt og spilleren har tapt eller når spilleren har klart å få på plass samtlige bokstaver i ordet.

2 Struktur

2.1 main

Dette er en struktur som vi bruker for å starte programmet vårt. Dette vil være en egen Python-fil som du kan kjøre, f.eks. ved navn *hangman.py*. Det er her du skal lage objektene av klassene Ordliste og HighScore. Du skal her skrive ut en meny til spillet hvor du lar brukeren velge om spilleren skal starte en sekvens av flere spill, vise high score-listen eller avslutte.

2.2 Klasser

Oppgaven skal ha en forholdsvis enkel objektorientert struktur. Du skal ha med de følgende klassene:

- *Ordliste*: Denne klassen skal ha støtte for å lese inn en ordliste fra en fil. Den skal inneholde to lister: én liste med brukte ord og én liste med ubrukte ord. Den skal ha støtte for å hente ut et tilfeldig Spillord fra listen over ubrukte Spillord. Hvis listen over ubrukte Spillord er tom skal den selv sørge for at alle de nå brukte ordene kommer tilbake i listen med ubrukte Spillord. Ordlisten skal kun leses inn fra fil en gang i løpet av programmet. Filen er gitt [her](#) og inneholder ett ord per linje.
- *Spillord*: Denne klassen representerer et ord brukt i spillet hangman. Den skal inneholde ordet og den skal ha en metode for å spille med dette ordet.
- *Spillsekvens*: Denne klassen skal inneholde data fra en sekvens av flere spill. Vi skal ha instansvariabler som lagrer en referanse til et Ordliste-objekt, poengsummen, antall spill spilt og navnet til spilleren. Fra en spillsekvens skal vi kunne hente ut poengsummen og navnet til spilleren. Vi må også implementere to metoder `__lt__` og `__gt__` for å kunne sammenligne to *Spillsekvens*-objekter og avgjøre hvilket som har høyest poengsum. Konstruktøren skal ta i mot en referanse til et Ordliste-objekt.
- *HighScore*: Denne klassen skal liste opp spillsekvenser i en synkende rekkefølge for å vise hvilke spillsekvenser som har høyest poengsum. Den skal kunne ta i mot en spillsekvens og putte spillsekvensen inn i en liste over spillsekvenser. Den skal kunne skrive ut listen i sortert rekkefølge.

Se diagrammet under for klassestruktur. Merk at dette diagrammet ikke gir samtlige variabler eller metoder, du må derfor definere flere variabler og metoder selv.

2.3 Metoder

Du skal implementere metodene under. Se her for beskrivelse av hva de skal gjøre. Merk at du må finne ut av eventuelle parametere selv.

- *Ordliste.hent_tilfeldig*: Henter et tilfeldig ord fra listen over ubrukte ord og flytter ordet over i listen av brukte ord. Hvis listen over ubrukte ord er tom skal du sette listen med brukte ord til å være ubrukte ord og omvendt. Returnerer det tilfeldige ordet.

Hint: for å hente ut et tilfeldig valgt ord er det nyttig å bruke funksjonen "choice" fra random-biblioteket. Importer funksjonen først med "from random import choice", da kan du bruke "choice(liste)" hvor "liste" er listen du vil hente et tilfeldig element fra.

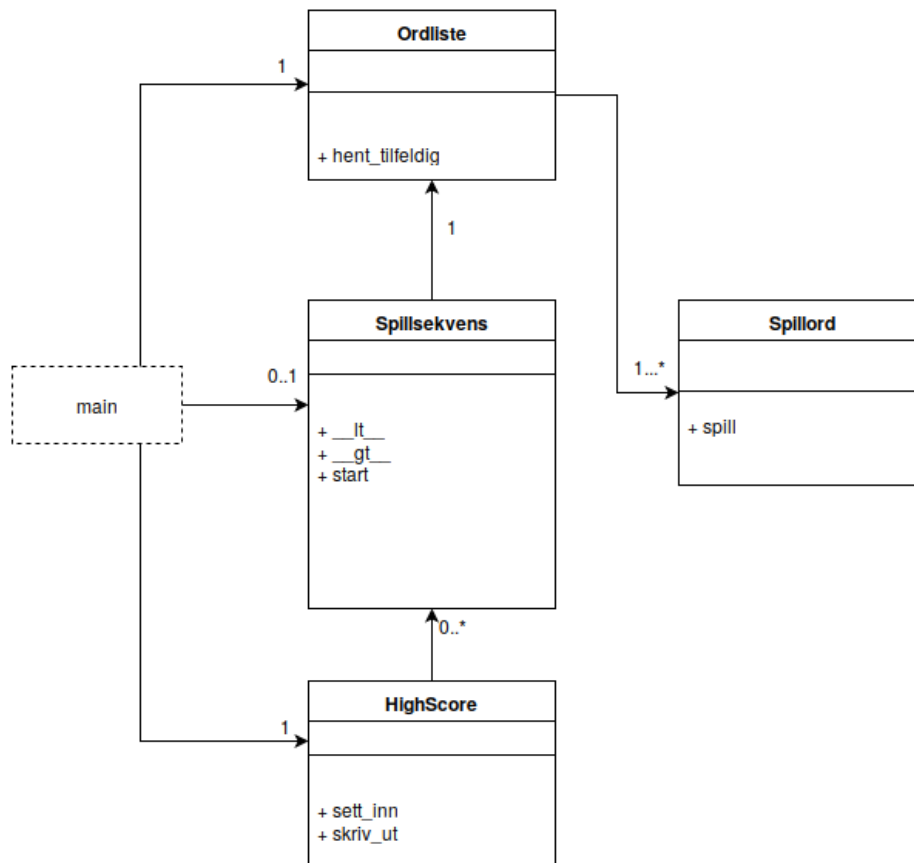


Figure 1: Diagram over klassestruktur

- *Spillord.spill*: Spiller et enkelt spill med hangman. Returnerer poengsummen spilleren har fått. Poengsummen skal være lik antall tegn (bokstaver) i ordet. Hvis spilleren ikke har klart å løse oppgaven skal antall poeng som returneres være 0. Se spillreglene for mer informasjon om et spill.

Hint: Bruk String sin lower() metode for å unngå å gjøre forskjell på store og små bokstaver som blir skrevet inn, lag en liste over brukte tegn og lag en liste som du fyller ut med like mange understrek tegn som det er tegn i ordet. Du kan da fylle ut "riktige" tegn i denne listen når brukeren klarer å skrive inn dette.

- *Spillsekvens.lt*: Tar i mot et annet *Spillsekvens*-objekt som parameter. Skal returnere True hvis objektet vi er i er "mindre enn" objektet vi mottar som parameter. "Mindre enn" betyr at poengsummen er lavere. Skal ellers returnere False.

- *Spillsekvens.__gt__*: Tar i mot et annet *Spillsekvens*-objekt som parameter. Skal returnere True hvis objektet vi er i er ”større enn” objektet vi mottar som parameter. ”Større enn” betyr at poengsummen er høyere. Skal ellers returnere False.
- *Spillsekvens.start*: Starter en spillsekvens. Den må registrere navnet til spilleren for så å kjøre et spill. Vi skal her ha en løkke hvor vi spiller spillene. Den skal hente et ord fra *Ordliste*-objektet Hvis spilleren klarer spillet, dvs. får en poengsum større enn 0 skal spilleren få mulighet til å spille på nytt. Den skal lagre total poengsum over alle spill i denne sekvensen med spill og antall spill spilleren har spilt i sekvensen.
- *HighScore.sett_inn*: Setter inn et *Spillsekvens*-objekt i listen over spillsekvenser.
- *HighScore.skriv_ut*: Skriver ut spillsekvensene i sortert rekkefølge. For å sortere en liste kan du bruke *liste.sort(reverse=True)* hvor *liste* er listen du vil sortere. For å kunne gjøre dette må du ha implementert *__lt__* og *__gt__* i klassen til objektene du skal sortere. Spillet med høyest poengsum skal komme først. Metoden skal skrive ut plassen sekvensen er på, navnet på spilleren og antall poeng og antall spill spilleren har spilt i spillsekvensen.

3 Eksempelkjøring

```
> python3 hangman.py
Meny:
1. Spill
2. Vis high score
0. Avslutt
Valg: 1
Skriv inn navn: Yrjan
Du har 10 forsøk igjen, ordet ser slik ut: _____
Skriv inn en bokstav: e
Du har 10 forsøk igjen, ordet ser slik ut: _____e__e
Skriv inn en bokstav: r
Du har 10 forsøk igjen, ordet ser slik ut: ___r___e__e
Skriv inn en bokstav: s
Du har 10 forsøk igjen, ordet ser slik ut: ___r_s__e__e
Skriv inn en bokstav: n
Du har 10 forsøk igjen, ordet ser slik ut: ___r_s__en_e
Skriv inn en bokstav: f
Du har 9 forsøk igjen, ordet ser slik ut: ___r_s__en_e
Skriv inn en bokstav: l
Du har 9 forsøk igjen, ordet ser slik ut: _l_r_s__en_e
Skriv inn en bokstav: a
Du har 9 forsøk igjen, ordet ser slik ut: _lar_s__en_e
Skriv inn en bokstav: f
```

Du har allerede skrevet denne bokstaven.
 Du har 9 forsøk igjen, ordet ser slik ut: _lar_s__en_e
 Skriv inn en bokstav: t
 Du har 9 forsøk igjen, ordet ser slik ut: _larts__en_e
 Skriv inn en bokstav: k
 Du har 9 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: j
 Du har 8 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: i
 Du har 7 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: v
 Du har 6 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: m
 Du har 5 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: r
 Du har allerede skrevet denne bokstaven.
 Du har 5 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: p
 Du har 4 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: o
 Du har 3 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: i
 Du har allerede skrevet denne bokstaven.
 Du har 3 forsøk igjen, ordet ser slik ut: klartsk_en_e
 Skriv inn en bokstav: u
 Du har 3 forsøk igjen, ordet ser slik ut: klartskuen_e
 Skriv inn en bokstav: y
 Du har 2 forsøk igjen, ordet ser slik ut: klartskuen_e
 Skriv inn en bokstav: d
 Riktig, løsningen var "klartskuende"!
 Vil du gi deg nå? (ja/[nei]) nei
 Du har 10 forsøk igjen, ordet ser slik ut: _____
 Skriv inn en bokstav: e
 Du har 10 forsøk igjen, ordet ser slik ut: _____e__e__e
 Skriv inn en bokstav: r
 Du har 10 forsøk igjen, ordet ser slik ut: _r_____e___er_e
 Skriv inn en bokstav: t
 Du har 9 forsøk igjen, ordet ser slik ut: _r_____e___er_e
 Skriv inn en bokstav: a
 Du har 9 forsøk igjen, ordet ser slik ut: _r_____e_a_er_e
 Skriv inn en bokstav: v
 Du har 8 forsøk igjen, ordet ser slik ut: _r_____e_a_er_e
 Skriv inn en bokstav: g
 Du har 8 forsøk igjen, ordet ser slik ut: _r_g___e_a_er_e
 Skriv inn en bokstav: l
 Du har 7 forsøk igjen, ordet ser slik ut: _r_g___e_a_er_e

Skriv inn en bokstav: p
Du har 7 forsøk igjen, ordet ser slik ut: pr_g___e_a_er_e
Skriv inn en bokstav: r
Du har allerede skrevet denne bokstaven.
Du har 7 forsøk igjen, ordet ser slik ut: pr_g___e_a_er_e
Skriv inn en bokstav: o
Du har 7 forsøk igjen, ordet ser slik ut: prog_o_e_a_er_e
Skriv inn en bokstav: r
Du har allerede skrevet denne bokstaven.
Du har 7 forsøk igjen, ordet ser slik ut: prog_o_e_a_er_e
Skriv inn en bokstav: t
Du har allerede skrevet denne bokstaven.
Du har 7 forsøk igjen, ordet ser slik ut: prog_o_e_a_er_e
Skriv inn en bokstav: v
Du har allerede skrevet denne bokstaven.
Du har 7 forsøk igjen, ordet ser slik ut: prog_o_e_a_er_e
Skriv inn en bokstav: b
Du har 6 forsøk igjen, ordet ser slik ut: prog_o_e_a_er_e
Skriv inn en bokstav: n
Du har 6 forsøk igjen, ordet ser slik ut: progno_e_a_erne
Skriv inn en bokstav: s
Du har 6 forsøk igjen, ordet ser slik ut: prognose_a_erne
Skriv inn en bokstav: m
Du har 6 forsøk igjen, ordet ser slik ut: prognosema_erne
Skriv inn en bokstav: k
Riktig, løsningen var "prognosemakerne"!
Vil du gi deg nå? (ja/[nei]) ja
Du fikk totalt 27 poeng!
Meny:
1. Spill
2. Vis high score
0. Avslutt
Valg: 2
***** HIGH SCORE *****
1. Yrjan, 27 poeng, 2 spill
Meny:
1. Spill
2. Vis high score
0. Avslutt
Valg: