

Arbeidskrav 2: PGR100 høst 2017

Frist:

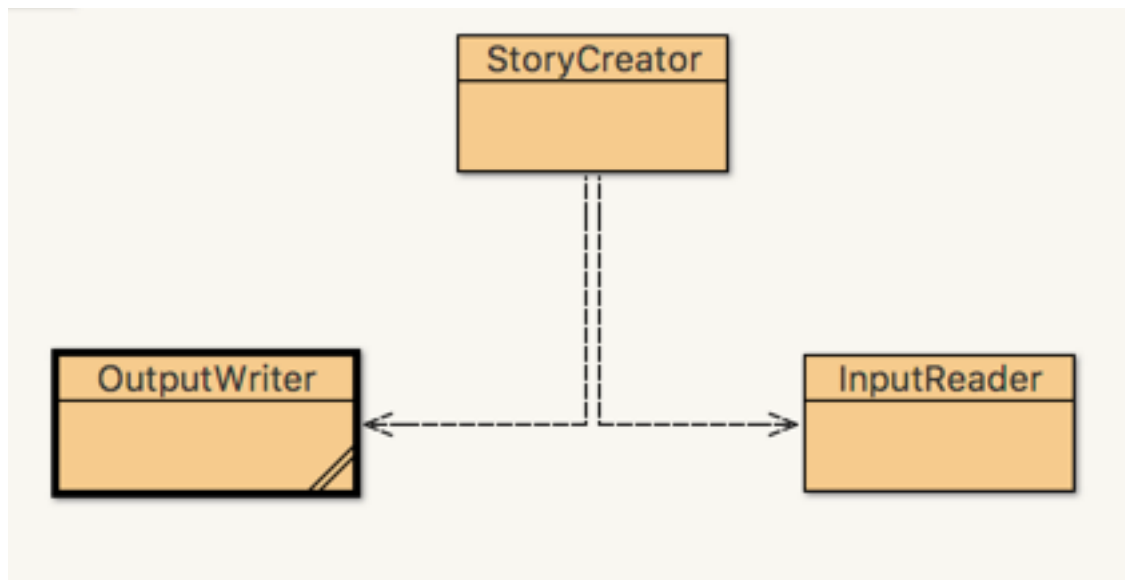
Søndag 5.11 innen 23:59. Leveres i It's Learning. IKKE LEVER I SISTE LITEN!

Mål:

Lage et program som kan produsere en [adjektivhistorie](#). En adjektivhistorie består av et historie-utgangspunkt der vi fyller inn forskjellige adjektiver der adjektiver kan passe inn. Dette kan gjøres med penn og papir, men det går jo mye raskere med et dataprogram!

Utgangspunkt:

Når du åpner prosjektet vil du se klassene i figuren nedenfor.



Gjør deg kjent med de eksisterende klassene, og hva de gjør. Disse vil du få bruk for! Du vil se at i noen av oppgavene nedenfor er det presisert hvilke metoder som skal implementeres i hvilke klasser. I noen oppgaver kan du gjøre egne valg. Det kan være lurt å lage noen hjelpemetoder for å kunne holde metodelengdene (antall kodelinjer) nede, men dette vil ikke påvirke vurderingen.

StoryCreator

Denne klassen representerer logikken for å lage adjektivhistoriene og det er her det meste av koden må skrives.

OutputWriter

Denne representerer skriving til fil.

InputReader

Denne klassen leser inn fra fil.

Oppgaver:

Denne innleveringen krever at du gjør følgende:

1. Implementer metoden

```
public void createAdjectiveStory(String storyFilename, String
adjectivesFilename, String outputFilename) i klassen StoryCreator.
```

Metoden skal benytte innholdet i en fil med filnavn `storyFilename` som grunnlag for historien. Historien inneholder noen forekomster av ordet 'ADJEKTIV'. Disse skal du erstatte med *tilfeldige* adjektiver fra filen med filnavn `adjectivesFilename`. Du finner eksempler på to historier og en adjektivfil i prosjektmappen (aftenpostenhistorie.txt, adjektivhistorie.txt og adjektiv.txt). Du kan også velge å lage egne filer hvis du ønsker det, men de må da følge samme regler for oppsett. Når du har lagt adjektiv inn i historien skal du skrive innholdet til en fil med filnavn `outputFilename`. Du trenger ikke å bekymre deg over manglende linjeskift eller problemer med encoding (rare æøå etc). Du trenger heller ikke bekymre deg over adjektiver som ikke blir helt grammatisk riktige, for eksempel: «Jeg spiste en gang et grønn eple...» Men du må passe på å ikke skifte ut deler av ord som for eksempel «adjektivhistorie».

2. Implementer metoden

```
public void createAdjectiveStory(String storyFilename, String
outputFilename) i klassen StoryCreator.
```

Her skal du også lage en adjektivhistorie, men denne gangen skal en bruker skrive inn adjektiv via terminalvinduet. Brukeren skal skriv inn adjektiv inntil han/hun er fornøyd og avbryter ved å skrive 'exit'. Da blir historien skrevet til fil (som i 1).

3. Implementer metoden

```
public void createAdjectiveStoryFromDictionary(String storyFilename,
String dictionaryFilename, String outputFilename) i klassen
StoryCreator.
```

Norges Scrabble-forbund har en ordliste (NSF-ordlisten) vi kan benytte for å sette inn tilfeldige adjektiver. Finn ordlisten (NSF-ordlisten.txt) i prosjektmappa og undersøk hvordan formatet på filen er. Fra filen skal du klare å trekke ut alle adjektivene og dermed bruke tilfeldige scrabble-godkjente adjektiver i adjektivhistorien din. Metoden fungerer ellers som i (1).

Tips: Husk å gjøre små endringer underveis og pass på at disse endringene kompilerer. Da er det mye enklere å rette opp bugs som dukker opp. Husk at det ikke er katastrofe om det er noe du ikke får til. Da leverer du det du har fått til.

Ekstraoppgaver (ikke nødvendig for å få godkjent):

To ekstraoppgaver:

- **Store/små bokstaver:** Håndter utfordringen at et punktum, utropstegn eller spørsmålstegn vil medføre stor bokstav i neste ord. Du skal altså sørge for at et adjektiv som blir plassert inn i en historie som første ord i en setning skal ha stor bokstav. Andre plasseringer skal medføre kun små bokstaver (selv hvis de i utgangspunktet er skrevet med store bokstaver).
- **Validering av antall substantiver:** Du skal sørge for at det er nok antall adjektiver til å bli satt inn i historien, og et adjektiv skal ikke benyttes flere ganger. I basisoppgavene kan vi fint benytte et adjektiv flere ganger. Men nå skal du altså kreve mange nok adjektiver for å fylle hele historien med unike adjektiver. I de tilfellene der adjektivene leses fra fil (basisoppg 1 og 3) skal det i output-filen stå: «Beklager. Det var ikke tilstrekkelig med adjektiver for å lage historien. Vennligst lag en fylligere adjektivliste og prøv igjen!» Når adjektivene kommer fra bruker (basisoppg 2) skal brukeren underveis få opplyst hvor mange adjektiver det gjenstår å skrive inn. Hvis input da avbrytes før vi har nok adjektiver, så skal brukeren (i terminalvinduet) få beskjed om at historien ikke kan bli laget da det ikke ble skrevet inn tilstrekkelig antall adjektiver.

Flere ekstraoppgaver (kun for moro skyld):

Legg inn en mulighet for å kunne publisere på HTML-format. Pimp siden slik at den ser fantastisk ut!

OBS! Dette skal ikke påvirke noen av de andre oppgavene beskrevet ovenfor. Dette er kun en separat utvidelse av programmet.

Vurdering:

Oppgaven kan godkjennes på 2 måter:

1: Oppgave er løst som beskrevet :

- Kildekoden blir vurdert til bestått.

Eller

2: *Dokumentasjon på innsats:*

Du klarte *ikke* å løse oppgaven, men dokumenterer::

- Hvordan du har jobbet med oppgaven
- Hvor mye tid du har brukt på oppgaven?
- Hvilke hjelpemidler du har brukt
- Hvilken hjelp du mener du trenger for å komme videre med din løsning

Det legges i vurderingen vekt på:

- at klasser / metoder er skrevet i tråd med beskrivelsen i oppgaven
- ryddig kode (luft, innrykk) og standard navngiving av klasser, metoder og variabler
- fornuftige datatyper
- klargjørende kommentering av kode der det er naturlig
- at klasser / metoder virker som tiltenkt
- at det ikke gjentas kode, men kaller på metoder der det er naturlig

Innleveringen teller ikke inn på endelig karakter, men må bestås for å kunne gå opp til eksamen.

Levering:

Oppgaven leveres som en zip fil på It's Learning.

Oppgaven er individuell.

Lykke til!