

Projekt Laws of Nature

Grafisk vejrstation



Skrevet og programmeret af Mathias Michailidis Skarving

Indholdsfortegnelse

Problemformulering.....	1
Krav til program.....	1
Teori for programmet.....	1
Læs data i JSON filer.....	1
Brug af API-key	1
Omdannelse af tidskoder	1
Tidsplan.....	2
Github	2
Flowchart	2
UML-Diagram	3
Test af programmet.....	4
Konklusion.....	4

Problemformulering

Hvordan programmerer man en grafisk vejrstation, som kan vise live vejrdato over alle verdens byer?

Krav til program

Programmet skal kunne fremstille vejrdato for alle byer i hele verden på en visuel måde. Vejrdatoen indeholder bl.a. temperatur, solopgang, solnedgang og en vejrbeskrivelse.

Teori for programmet

Læs data i JSON filer

Programmet skal kunne læse data fra JSON filer, hvilket kræver to processing libraries:

```
import processing.data.JSONArray;  
import processing.data.JSONObject;
```

Dette gøres ved at loade JSON filen fra en URL, og derefter hente den krævede data fra filen.

```
myCity = new Cityselector(cityname);  
myCity.cityselect(loadJSONArray("worldcities.json"));
```

Brug af API-key

API-keyen bruges til at hente live vejrdato over byerne. Dette gøres ved at modificere en url med koordinaterne til den ønskede by. Det sted, vi henter data fra, er kaldet Openweathermap.

```
url = String.format(format:"https://api.openweathermap.org/data/2.5/weather?lat=%s&lon=%s&units=metric&appid=89a2601e9cd2d8c8c5f113cba2ba204e",  
lat, lng);
```

Koordinaterne finder programmet på en lokal JSON-fil, der indeholder en liste af over størstedelen af alle byer i hele verden. Listen af byer er hentet fra dette GitHub repository:

<https://github.com/dr5hn/countries-states-cities-database>

Omdannelse af tidskoder

Tidskoderne for hhv. solopgang og solnedgang bliver hentet fra via vores API-key. Denne tid er beskrevet i Unix-tid, og er defineret ved det antal sekunder, der er gået siden 00:00:00 UTC den 1. januar 1970.

Unix-tid omdannes i koden gennem disse funktioner:

```
Date sunrisetime = new java.util.Date((long) myData.getsunrise() * 1000);  
Date sunsettime = new java.util.Date((long) myData.getsunset() * 1000);  
sunsetdate = df.format(sunsettime);  
sunrisedate = df.format(sunrisetime);
```

Her hentes tidszonen fra computeren, hvilket sikrer at vi får tiden i den rigtige tidszone.

Tidsplan

For at overholde en tidsplan skal projektet foregå sådan:

Opgave	Afleveringsdato	Dato for færdiggørelse
Projektbeskrivelse	23/11/2023	23/11/2023
Programmet	16/01/2024	16/01/2024

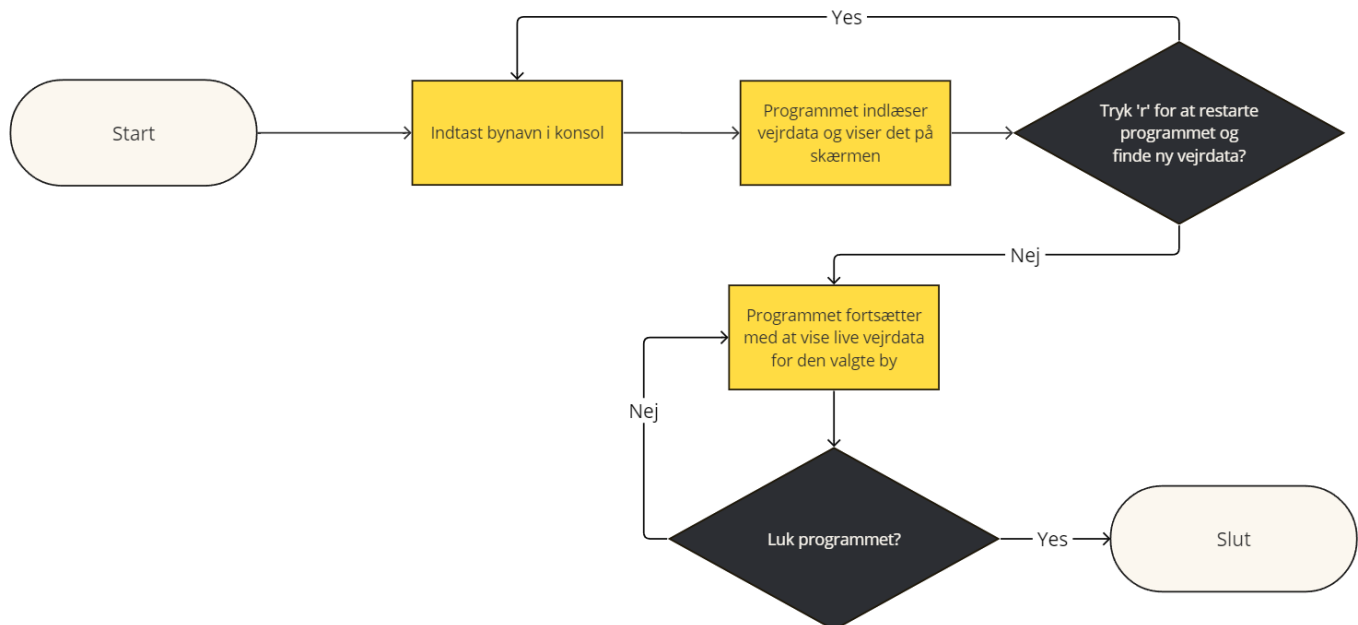
Github

Koden er delt til offentlig tilgængelighed på min Github konto, så den er nem at tilgå:

<https://github.com/MathiasSkarving/Projekt-Natures-Love>

Flowchart

Herunder ses et diagram for hele programmets forløb, når det startes til det lukkes ned. Dette gøres for at give et nemt overblik over, hvordan programmet fungerer og forløber.



Figur 1: Flowchart

UML-Diagram

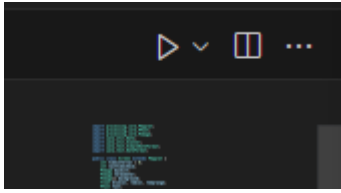
Programmet er konstrueret fra fire klasser, hvor superklassen er PApplet, processing's klasse. UML-Diagrammets formål er at give et overskueligt overblik over programmets variable og metoder. Diagrammet ses til nedenunder:



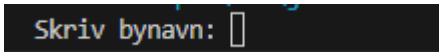
Figur 2: UML-Diagram

Test af programmet

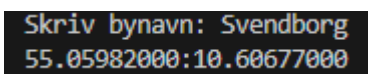
I dette afsnit tester vi programmets funktionsdygtighed, for at få at se om det virker efter kravene.



Trin 1: Programmet køres



Trin 2: Man bliver bedt om at skrive et bynavn

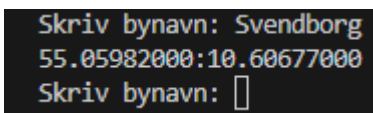


Trin 3: Bynavn skrives i konsollen, og programmet finder med det samme byens koordinater



Trin 4: Der åbnes et vindue, som grafisk repræsenterer vejret i byen

Nu trykkes der på 'r' for at starte og finde en ny by.



Trin 5: Man kan nu skrive et nyt bynavn

Konklusion

Vi kan ud fra testen se, at programmet opfylder alle kravene, og at programmet virker fejlfrit.