# Projekt Laws of Nature

## Grafisk vejrstation



Skrevet og programmeret af Mathias Michailidis Skarving

Indholdsfortegnelse

[Problemformulering 1](#_Toc156317838)

[Krav til program 1](#_Toc156317839)

[Teori for programmet 1](#_Toc156317840)

[Læs data i JSON filer 1](#_Toc156317841)

[Brug af API-key 1](#_Toc156317842)

[Omdannelse af tidskoder 1](#_Toc156317843)

[Tidsplan 2](#_Toc156317844)

[Github 2](#_Toc156317845)

[Flowchart 2](#_Toc156317846)

[UML-Diagram 3](#_Toc156317847)

[Test af programmet 4](#_Toc156317848)

[Konklusion 4](#_Toc156317849)

# Problemformulering

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

# Krav til program

Programmet skal kunne fremstille vejrdata for alle byer i hele verden på en visuel måde. Vejrdataen 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 proccessing libraries:



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



## Brug af API-key

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



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: Et billede, der indeholder tekst, skærmbillede, Font/skrifttype

Automatisk genereret beskrivelse

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-Naturens-Love>

# Flowchart

A diagram of a diagram

Description automatically generatedHerunder 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 : Flowchart

# UML-Diagram

Et billede, der indeholder tekst, skærmbillede, Font/skrifttype, Tryk

Automatisk genereret beskrivelseProgrammet er konstrueret fra fire klasser, hvor superklassen er PApplet, proccesing’s klasse. UML-Diagrammets formål er at give et overskueligt overblik over programmets variable og metoder. Diagrammet ses til nedenunder:

Figur : UML-Diagram

# Test af programmet

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

Et billede, der indeholder skærmbillede, tekst, Font/skrifttype, design

Automatisk genereret beskrivelse

Trin : Programmet køres



Trin : Man bliver bedt om at skrive et bynavn



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

Et billede, der indeholder tekst, sky, skærmbillede, udendørs

Automatisk genereret beskrivelse

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

Nu trykkes der på ’r’ for at restarte og finde en ny by.

Et billede, der indeholder tekst, Font/skrifttype, skærmbillede, nummer/tal

Automatisk genereret beskrivelse

Trin : Man kan nu skrive et nyt bynavn

# Konklusion

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