



# Politechnika Wrocławska

---

Platformy programistyczne .Net i Java  
**Dokumentacja programu wyświetlającego pogodę z API**

Prowadzący: Mgr. inż. Michał Jaroszczuk  
Grupa Środa, 18:55-20:40

Jan Klisowski 263485

27.03.2024

# Spis treści

<b>1</b>	<b>Opis Projektu</b>	<b>2</b>
<b>2</b>	<b>Opis Klas i Metod</b>	<b>2</b>
2.1	Klasa Program . . . . .	2
2.2	Klasa Form1 . . . . .	2
2.2.1	Metoda <code>getWeather</code> . . . . .	2
2.2.2	Metoda <code>convertDateTime</code> . . . . .	2
2.3	Klasa <code>WeatherInfo</code> . . . . .	2
<b>3</b>	<b>Repozytorium</b>	<b>2</b>
<b>4</b>	<b>Zdjęcia</b>	<b>3</b>
4.1	Interfejs Graficzny Aplikacji . . . . .	3
<b>5</b>	<b>Zdjęcia</b>	<b>3</b>
5.1	Drzewo Projektu . . . . .	3
5.2	Kluczowy Fragment Programu . . . . .	4

# 1. Opis Projektu

Projekt dotyczy implementacji aplikacji pogodowej, która umożliwi użytkownikowi sprawdzenie aktualnej pogody na podstawie danych z zewnętrznego API. Aplikacja została napisana w języku C# z wykorzystaniem platformy .NET Framework.

## 2. Opis Klas i Metod

### 2.1. Klasa Program

Klasa `Program` zawiera główny punkt wejścia dla aplikacji, czyli metodę `Main`. Metoda ta inicjuje interfejs graficzny aplikacji.

### 2.2. Klasa Form1

Klasa `Form1` reprezentuje główne okno aplikacji. Umożliwia użytkownikowi wprowadzenie nazwy miasta i uzyskanie informacji o pogodzie w tym miejscu. Wyświetla pobrane dane, takie jak warunki pogodowe, temperatura, wschód i zachód słońca, ciśnienie oraz prędkość wiatru.

#### 2.2.1. Metoda `getWeather`

Metoda `getWeather` jest odpowiedzialna za pobranie danych o pogodzie z zewnętrznego API na podstawie nazwy miasta wprowadzonej przez użytkownika. Następnie deserializuje otrzymaną odpowiedź JSON i wyświetla informacje w interfejsie użytkownika.

#### 2.2.2. Metoda `convertDateTime`

Metoda `convertDateTime` służy do konwersji czasu z formatu UNIX na lokalny czas użytkownika.

### 2.3. Klasa `WeatherInfo`

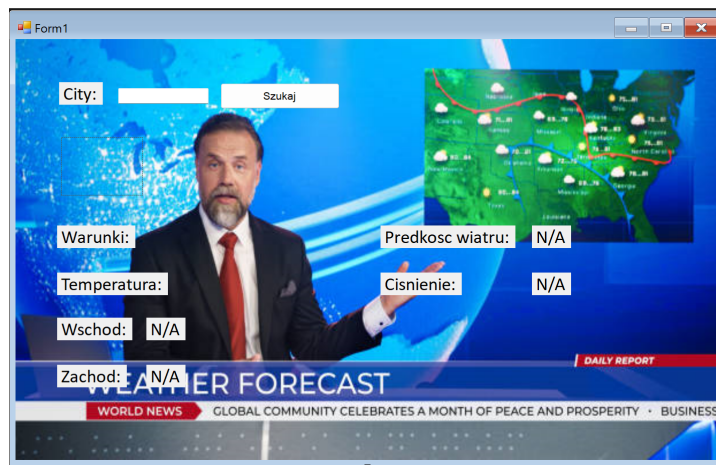
Klasa `WeatherInfo` zawiera definicję struktury danych używanej do deserializacji odpowiedzi z API pogodowego. Zawiera informacje o warunkach pogodowych, temperaturze, wschodzie i zachodzie słońca, prędkości wiatru i ciśnieniu.

## 3. Repozytorium

Link do repozytorium projektu: <https://github.com/jachoofrachoo/netjavaKlis>

## 4. Zdjęcia

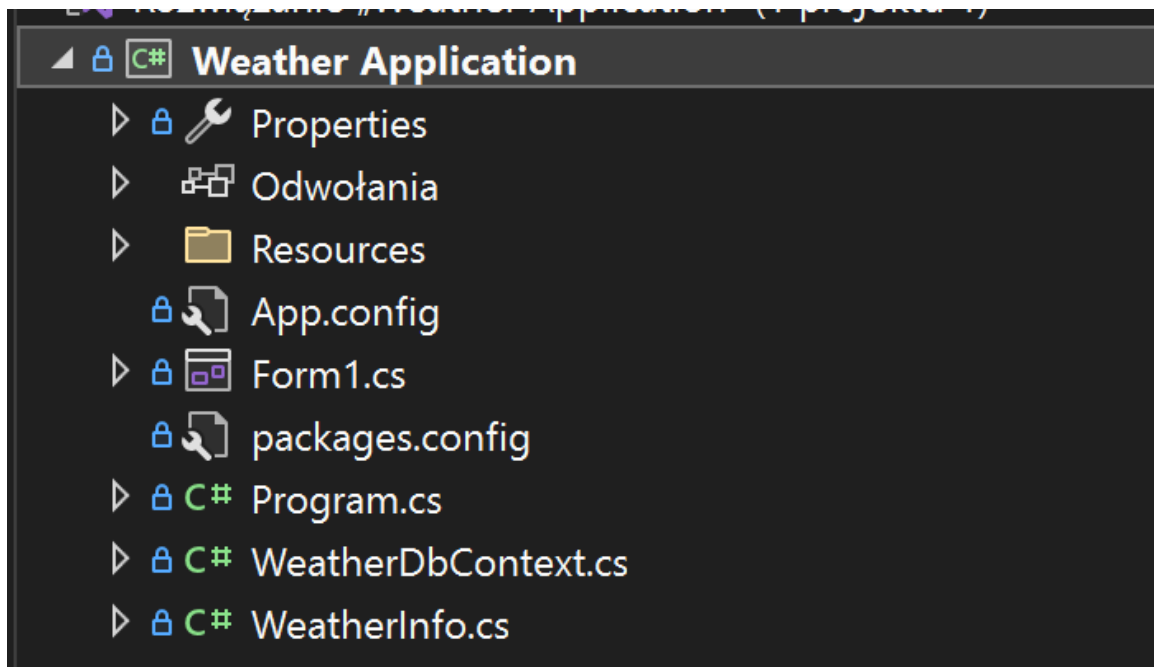
### 4.1. Interfejs Graficzny Aplikacji



Rys. 1: Interfejs Graficzny Aplikacji

## 5. Zdjęcia

### 5.1. Drzewo Projektu



Rys. 2: Drzewo Projektu

## 5.2. Kluczowy Fragment Programu

```
14 namespace Weather_Application
15 {
16     public partial class Form1 : Form
17     {
18         public Form1()
19         {
20             InitializeComponent();
21         }
22
23         string APIKey = "6ee9181e77a4792d55dd15b798c1c295";
24
25         private void labSunrise_Click(object sender, EventArgs e)
26         {
27
28         }
29
30         private void btnSearch_Click(object sender, EventArgs e)
31         {
32             getWeather();
33         }
34     }
35 }
```

Rys. 3: Kluczowy Fragment Programu

```
35 void getWeather()
36 {
37     using (WebClient web = new WebClient())
38     {
39         string url = string.Format("https://api.openweathermap.org/data/2.5/weather?q={0}&appid={1}", TBCity.Text, APIKey);
40         var json = web.DownloadString(url);
41         WeatherInfo.root Info = JsonConvert.DeserializeObject<WeatherInfo.root>(json);
42
43         picIcon.ImageLocation = "https://openweathermap.org/img/w/" + Info.weather[0].icon + ".png";
44         labCondition.Text = Info.weather[0].main;
45         double tempKelvin = Info.main.temp;
46         double tempCelsius = tempKelvin - 273.15;
47         labDetails.Text = "Temperatura: " + tempCelsius.ToString("0.##") + " °C";
48         labSunset.Text = convertDateTime(Info.sys.sunset).ToShortTimeString();
49         labSunrise.Text = convertDateTime(Info.sys.sunrise).ToShortTimeString();
50
51         labWindSpeed.Text = Info.wind.speed.ToString();
52         labPressure.Text = Info.main.pressure.ToString() + " hPa";
53     }
54 }
55
56
57 DateTime convertDateTime(long sec)
58 {
59     DateTime day = new DateTime(1970, 1, 1, 0, 0, 0, 0, System.DateTimeKind.Utc).ToLocalTime();
60     day = day.AddSeconds(sec).ToLocalTime();
61
62     return day;
63 }
```

Rys. 4: Kluczowy Fragment Programu