

# Dokumentacja aplikacji WeatherApp

## 1. Dokumentacja użytkownika:

- ➔ włączenie programu odbywa się poprzez uruchomienie WeatherPlot.py
- ➔ w pierwszej kolejności należy wyszukać miasto poprzez wpisanie jego nazwy do pola opisanego „Miasto” znajdującego się w prawym–górnym rogu (dodatkowo można zdefiniować kod państwa w polu obok)
- ➔ po uzupełnieniu tych danych trzeba kliknąć „Szukaj”, dzięki czemu pojawia się okno z wyborem miasta (odpowiednie okno dialogowe w przypadku jego nie znalezienia)
- ➔ wybranie miasta odbywa się poprzez kliknięcie na wiersz, w którym znajduje się poszukiwana wartość, a następnie naciśnięcie „Wybierz” – po tym okno zniknie, a w głównym oknie pokaże się prognoza pogody dla wybranego miasta
- ➔ w celu podglądu danych historycznych należy wybrać pożądaną opcję z rozwijalnej listy znajdującej się w lewym–górnym rogu, a następnie wciśnięcie przycisku „Wyświetl”
- ➔ powrót do prognozy umożliwia przycisk „Prognoza pogody”
- ➔ funkcje znajdujące się w lewym–górnym rogu nie działają przed wybraniem miasta – po ich przyciśnięciu pojawia się błąd

## 2. Dokumentacja kodu:

### A) klasa WeatherGui – zarządza oknem graficznym

- ➔ `__initGui()` – inicjalizuje komponenty i ich układ
- ➔ `__getCityLocation()` – wyświetla okno ze znalezionymi miastami
- ➔ `__setCity(city)` – ustawia miasto podane w argumencie jako aktualne
- ➔ `__showForecast()` – wyświetla wykres prognozy dla danego miasta
- ➔ `__showHistorical()` – wyświetla wykres historycznych danych
- ➔ `__drawPlot()` – część wspólna dwóch powyższych metod
- ➔ `center(win)` – centruje okno podane w argumencie względem pulpitu

### B) klasa WeatherApi – zarządza pobieraniem danych z openweathermap

- ➔ `getCityLocation(city, country)` – zwraca listę znalezionych miast na podstawie podanych argumentów – miasta i opcjonalnie państwa
- ➔ `setLocation(lat, lon)` – ustawia lokalizację dla której będzie wyszukiwana pogoda
- ➔ `__getData(params=None, endpoint='')` – wykonuje zapytanie i zwraca wyniki w postaci JSON, wykorzystywana przez dwie poniższe metody
- ➔ `getForecastWeather()` – ustawia parametry i endpoint dla prognozy pogody
- ➔ `getHistoricalData(dt)` – ustawia parametry i endpoint dla danych historycznych, w tym czas – parametr dt

### C) klasa WeatherPlot – zarządza generowaniem wykresu

- ➔ `forecastPlot(data, city, state, country)` – przetwarza dane prognozy pogody
- ➔ `historicalPlot(data, city, state, country)` – przetwarza dane historyczne
- ➔ `__weatherPlot(processed, text, dateFormat)` – dodaje wykresy na podstawie przetworzonych danych, wykorzystywana przez powyższe metody
- ➔ `__addPlot(x, y, z, text, pos, i, dateFormat)` – tworzy wykresy na podstawie podanych osi (pionowe Y, Z – pozioma X), text zależnie od rodzaju danych, pos, i – zależnie który z dwóch wykresów to jest, dateFormat – jak wyświetlana jest data w osi X