

```
# Travelinator działa i dostarcza wymaganych informacji. Załóżmy jednak, że w tym
# momencie zadzwonił do nas klient i powiedział, że podoba mu się działanie programu,
# ale chciałby użyć go jako interaktywnego chatbot. Naszym celem będzie dokonanie
# refaktoryzacji - zmian w kodzie, dzięki którym sprosta on nowym wymaganiom.
```

```
origin_place = None
destination_place = None
while True:
    chosen_option = int(input('''Jaką akcję chcesz wykonać?
    1.Podaj/zmień miejsce startowe
    2.Podaj/zmień miejsce docelowe
    3.Sprawdź lokalizację miejsca startowego
    4.Sprawdź lokalizację miejsca docelowego
    5.Sprawdź pogodę miejsca startowego
    6.Sprawdź pogodę miejsca docelowego
    7.Dowiedz się więcej o walucie
    8.Koniec'''))

    if chosen_option == 1:
        pass
    elif chosen_option == 2:
        pass
    elif chosen_option == 3:
        pass
    elif chosen_option == 4:
        pass
    elif chosen_option == 5:
        pass
    elif chosen_option == 6:
        pass
    elif chosen_option == 7:
        pass
    elif chosen_option == 8:
        quit()
    else:
        print('Podano błędną opcję')
    print("Naciśnij enter aby kontynuować...")
    input()
```

```
# REFAKTORYZACJA
import requests
from pprint import pprint

API_KEY = "62a5baf7f44e76a6b77b95d6357197f4"

def check_coordinates(city, API_KEY):
    response =
requests.get(f"http://api.openweathermap.org/geo/1.0/direct?q={city}&appid={API_KEY}")
    print(response.status_code)
    pprint(response.json())
    lat = response.json()[0]['lat'] # szerokość geograficzna
    lon = response.json()[0]['lon'] # długość geograficzna
    city = response.json()[0]['name']
    country = response.json()[0]['country']
    return lat, lon, city, country

def get_weather_info(lat, lon, API_KEY):
    response =
requests.get(f"https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={
API_KEY}")
    response_json = response.json()
    weather = response_json['weather'][0]['description'] # pogoda
    temperature = response_json['main']['temp'] # temperatura
    pressure = response_json['main']['pressure'] # ciśnienie
    humidity = response_json['main']['humidity'] # wilgotność
    return weather, temperature, pressure, humidity

def get_country_full_name(country_code):
    url = f"https://restcountries.com/v3.1/alpha/{country_code.upper()}"
    try:
        response = requests.get(url)
        country_name = response.json()[0]['name']['common']
    except:
        return country_code
    else:
        return country_name

def get_currency_code(country_code):
    url = f"https://restcountries.com/v3.1/alpha/{country_code.upper()}"

    response = requests.get(url)
    currency_code = list(response.json()[0]['currencies'].keys())[0]
    return currency_code
```

```

def get_currency_ratio(ori_curr, dest_curr):
    if ori_curr != "PLN":
        url = f"http://api.nbp.pl/api/exchangerates/rates/A/{ori_curr.lower()}/"
        response = requests.get(url)
        ori_ratio = response.json()['rates'][0]['mid']
    else:
        ori_ratio = 1

    if dest_curr != "PLN":
        url = f"http://api.nbp.pl/api/exchangerates/rates/A/{dest_curr.lower()}/"

        response = requests.get(url)
        dest_ratio = response.json()['rates'][0]['mid']
    else:
        dest_ratio = 1
    ratio = float(ori_ratio)/float(dest_ratio)
    return ratio


def print_weather_info(place):
    lat,lon,_ = check_coordinates(place, API_KEY)
    weather, temperature, pressure, humidity = get_weather_info(lat,lon, API_KEY)
    print(f"Pogoda dla miasta {place} : {weather}")
    print(f"Temperatura: {temperature} st.Celsjusza")
    print(f"Ciśnienie: {pressure} hPa")
    print(f"Wilgotność: {humidity} %")


# Travelinator działa i dostarcza wymaganych informacji. Załóżmy jednak, że w tym
# momencie zadzwonił do nas klient i powiedział, że podoba mu się działanie programu,
# ale chciałby użyć go jako interaktywnego chatbot. Naszym celem będzie dokonanie
# refaktoryzacji - zmian w kodzie, dzięki którym sprosta on nowym wymaganiom.


origin_place = None
destination_place = None
while True:
    print('''Jaką akcję chcesz wykonać?
    1.Podaj/zmień miejsce startowe
    2.Podaj/zmień miejsce docelowe
    3.Sprawdź lokalizację miejsca startowego
    4.Sprawdź lokalizację miejsca docelowego
    5.Sprawdź pogodę miejsca startowego
    6.Sprawdź pogodę miejsca docelowego
    7.Dowiedz się więcej o walucie
    8.Koniec''')
    chosen_option = int(input())

    if chosen_option == 1:
        origin_place = input("Podaj miasto startowe.\n")

    elif chosen_option == 2:
        destination_place = input("Podaj miasto docelowe.\n")

```

```

elif chosen_option == 3:
    if origin_place is not None:
        lat,lon,country = check_coordinates(origin_place, API_KEY)
        country_name = get_country_full_name(country)
        print(f"Miejsce {origin_place} leży w kraju {country_name}\n Długości
geograficzna: {lon}, szerokość geograficzna: {lat}")
    else:
        print("Najpierw musisz podać miasto startowe")
    # POPRAWIAMY FUNCKJE check_coordinates [nie musi zwracać już nazwy miasta]
    # i get_country_full_name [zabezpieczenie przed odrzuceniem zapytania]

elif chosen_option == 4:
    if destination_place is not None:
        lat,lon,country = check_coordinates(destination_place, API_KEY)
        country_name = get_country_full_name(country)
        print(f"Miejsce {destination_place} leży w kraju {country_name}\n Długości
geograficzna: {lon}, szerokość geograficzna: {lat}")
    else:
        print("Najpierw musisz podać miasto docelowe")

elif chosen_option == 5:
    # NOWA FUNKCJA print_weather_info
    if origin_place is not None:
        print_weather_info(origin_place)
    else:
        print("Najpierw musisz podać miasto startowe")

elif chosen_option == 6:
    if destination_place is not None:
        print_weather_info(destination_place)
    else:
        print("Najpierw musisz podać miasto startowe")

elif chosen_option == 7:
    if origin_place is not None:
        if destination_place is not None:
            _,_,origin_country_code = check_coordinates(origin_place, API_KEY)
            _,_,destination_country_code =
check_coordinates(destination_place,API_KEY)
            origin_currency = get_currency_code(origin_country_code)

            destination_currency = get_currency_code(destination_country_code)

            currency_ratio = get_currency_ratio(origin_currency,destination_currency)

            print(f"W mieście startowym obowiązuje waluta: {origin_currency}")
            print(f"W mieście docelowym obowiązuje waluta:{destination_currency}")

            print(f"Kurs walut 1 {origin_currency} =
{currency_ratio}{destination_currency}")
        else:
            print("Najpierw musisz podać miasto docelowe")
    else:
        print("Najpierw musisz podać miasto startowe")

```

```
elif chosen_option == 8:  
    quit()  
else:  
    print('Podano błędną opcję')  
print("Naciśnij enter aby kontynuować...")  
input()
```