

# Vremenska prognoza

Zadatak je napraviti konzolni program za ažurni ispis vremenske prognoza koristeći JSON API koji obezbeđuje “openweather” servis. Program omogućava ispis trenutne temperature kao i ispis 5dnevne prognoze. Grad se takođe može promeniti a prodrazumevani grad je Beograd čiji su podaci: `city: Belgrade, countryCode: RS`. Za rešavanje zadatka neophodna je internet konekcija.

Obezbeđena je metoda *getWeatherInfo* čiji potpis izgleda ovako:

```
public static String getWeatherInfo(String city, String countryCode, Type type) throws IOException;
```

Za dobijanje informacije o trenutnom vremenu poziva se ovako:

```
String result = getWeatherInfo("Belgrade", "RS", Type.CURRENT)
```

Za dobijanje informacije o prognozi poziva se ovako:

```
String result = getWeatherInfo("Belgrade", "RS", Type.FORECAST)
```

Za detaljne informacije o znacaju svakog dobijenog parametra posetite sledece linkove:

[https://openweathermap.org/current#current\\_JSON](https://openweathermap.org/current#current_JSON)

<https://openweathermap.org/forecast5#JSON>

Primer ispisa programa (kompletan I/O):

Dobrodošli u konzolnu vremensku prognozu!

-----

Komande:

-1) Izlaz iz programa

0) Promena grada [String]

1) Trenutna temperatura

2) Prognoza za 5 dana

99) Ispis komandnog menija

-----

komanda: 1

Trenutna: 21.46°C

Maksimalna: 22.22°C

Minimalna: 21°C

-----

komanda: 2

Uto: 19.61°C

Sre: 20.14°C

Čet: 17.71°C

Pet: 14.89°C

Sub: 16.67°C

-----

komanda: 0

Unesite grad u formatu: city, countryCode

London, GB

-----

komanda: 1

Trenutna: 17.34°C

Maksimalna: 18.89°C

Minimalna: 16°C

-----

komanda: -1

#### Sugestije:

*Za dohvatanje imena trenutnog dana možete koristiti sledeći snippet*

```
Calendar calendar = Calendar.getInstance(); //dobijamo instancu
int index = calendar.get(Calendar.DAY_OF_WEEK) - 1;
String[] dani = {"Ned", "Pon", "Uto", "Sre", "Čet", "Pet", "Sub"};
//dani[index] daje ime dana
```

*Za pronalazak narednog dana možete ikoristiti mogućnost dodavanja dana na na instancirani kalendar*

```
calendar.add(Calendar.DATE, 1);
```

Implementacija metode za dohvaćanje ažurne informacije o vremenskoj prognozi sa interneta. Ne menjati izvorni kod.

```
public class Program {

    static long lastRequestTimestamp;

    public static enum Type { CURRENT, FORECAST };

    public static String getWeatherInfo(String city, String countryCode, Type type) throws
IOException {

        String authToken = "d0f1969fd9856fe09e3f7d0753d84ed4";

        String addr = String.format("http://api.openweathermap.org/data/2.5/%s?
q=%s,%s&appid=%s&units=metric", type == Type.CURRENT ? "weather" : "forecast", city,
countryCode,authToken );

        if(System.currentTimeMillis() - lastRequestTimestamp < 5000) {

            try {

                Thread.sleep(5000);

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

        URLConnection yc = new URL(addr).openConnection();

        BufferedReader in = new BufferedReader(new InputStreamReader(yc.getInputStream()));

        StringBuilder buffer = new StringBuilder();

        String inputLine;

        while ((inputLine = in.readLine()) != null)

            buffer.append(inputLine);

        in.close();

        lastRequestTimestamp = System.currentTimeMillis();

        return buffer.toString();

    }

}
```