

**Střední škola informatiky a finančních
služeb, Plzeň, Klatovská 200 G**

**Maturitní práce
2019_07 Počasí**

Školní rok 2018/2019

Milan Abrahám, 4.E

Anh Thai Hoang, 4.E

Prohlašujeme, že jsme tuto maturitní práci vypracovali samostatně a veškeré zdroje uvedli v seznamu použitých informačních zdrojů.

V Plzni dne 4. 1. 2019

Obsah

1	Úvod	4
2	Získávání a zpracování dat	5
2.1	Připojení k API serveru	5
2.2	Žádost	5
2.3	Zpracování dat	6
2.4	Třídy pro získání a zpracování dat	7
2.4.1	Třída WeatherByCity	7
2.4.2	Třída WeatherByCoordinates.....	7
2.4.3	Třída WeatherForecast	7
3	Databáze a její propojení s aplikací	8
3.1	Vytvoření databáze	8
3.2	Připojení k databázi	8
3.2.1	Třída MySQLConnect.....	9
3.2.2	Metoda getData	9
3.3	Skript pro aktualizaci teplot v databázi	10
4	Uživatelské rozhraní.....	11
4.1	Hlavní okno	11
4.2	Menší okno pro zobrazování výsledků.....	13
4.3	Vypočtení souřadnic místa po kliknutí.....	13
4.4	Ikony počasí.....	14
5	Návod k použití programu.....	15
5.1	Využití vyhledávacího pole	15
5.2	Kliknutí na pozici na mapě.....	16
6	Závěr.....	17
7	Shrnutí	18
8	Použité zdroje informací	19

1 ÚVOD

Tato práce se bude zabývat vytvořením aplikace pro snadné a rychlé zjištění informací o počasí. Mezi hlavní cíle práce bude patřit vytvoření připojení k serveru a získávání dat o počasí, které server poskytuje, vytvoření databáze a uživatelského rozhraní a grafů. Práce bude obsahovat popis jednotlivých částí aplikace a postup při zpracování, porovnání s ostatními aplikacemi a návod k použití. Aplikace bude vytvořena pro operační systém Windows v programovacím jazyce Java.

2 ZÍSKÁVÁNÍ A ZPRACOVÁNÍ DAT

Data pro aplikaci získáváme z webového serveru OpenWeatherMap¹. Tato webová stránka shromažďuje data o počasí z meteorologických stanic a dalších podobných služeb po celém světě a poté je poskytuje vývojářům zdarma. Přístup k těmto datům je realizován pomocí API (= Application Programming Interface). Je to rozhraní, které umožňuje přístup k datům, využívá k tomu různé metody, které jsou popsány v dokumentaci¹.

Na sever musíme odeslat žádost a obratem obdržíme odpověď se všemi potřebnými daty. Data musíme zpracovat tak, aby byla následně byla lehce dostupná kdekoli v aplikaci. Při zpracování takové žádosti se může objevit chyba, a proto je potřeba kód ošetřit pomocí výjimek.

2.1 Připojení k API serveru

Serveru nejdříve musíme poslat žádost a v ní specifikovat jaká data, z jaké oblasti potřebujeme. Je proto potřeba vhodně modifikovat URL adresu, na kterou se žádost pošle. V aplikaci používáme 3 typy žádostí a to žádost o počasí podle obce, žádost o počasí podle souřadnic a žádost o předpověď na další dny. Jelikož každá tato žádost vrací jiné informace, má každá aplikace vlastní třídu.

2.2 Žádost

Každá žádost musí obsahovat autentizační klíč, který má každá aplikace vlastní a který jsme si museli nechat webovým serverem vygenerovat. Je to unikátní řetězec skládající se z čísel a písmen. Dále musíme specifikovat vyhledávané místo, jednotky (v našem případě metrické) a jazyk odpovědi. Ve zvoleném jazyce ovšem budou jenom textová data např. popis počasí, nikoliv celá odpověď.

Obrázek 1 - Ukázka žádosti

```
String.format("http://api.openweathermap.org/data/2.5/weather?q=%s&units=metric&APPID=%s&lang=cz", city, API_KEY)
```

¹ Weather API – OpenWeatherMap. Current weather and forecast – OpenWeatherMap [online]

2.3 Zpracování dat

Server nám odešle odpověď ve formátu JSON, což je „textový způsob zápisu dat (datový formát) nezávislý na počítačové platformě.“²

Obrázek 2 - Ukázka části odpovědi

```
"main":{
  "temp":276.15,
  "pressure":1004,
  "humidity":93,
  "temp_min":276.15,
  "temp_max":276.15
},
"visibility":4400,
"wind":{
  "speed":0.5
},
```

Java žádnou knihovnu pro zpracování tohoto formátu nemá, museli jsme tedy najít nějakou externí. Použili jsme externí knihovnu JSON-java³, která nám práci s odpovědí výrazně usnadnila. K jednotlivým hodnotám se přistupuje pomocí klíčových slov, pod kterými jsou uloženy.

Obrázek 3 - Část kódu pro zpracování odpovědi ve formátu JSON

```
JSONObject jo = new JSONObject(this.response_json);

JSONObject j_coord = jo.getJSONObject("coord");
this.lon = j_coord.getDouble("lon");
this.lat = j_coord.getDouble("lat");

JSONArray j_weather = jo.getJSONArray("weather");
JSONObject jo_weather = j_weather.getJSONObject(0);
this.weather_id = jo_weather.getInt("id");
this.main = jo_weather.getString("main");
this.description = jo_weather.getString("description");
this.icon = jo_weather.getString("icon");
```

² JSON – textový formát pro výměnu dat. Promotic [online]

³ GitHub – stleary/JSON-java: A reference implementation of a JSON package in Java. GitHub [online]

2.4 Třídy pro získání a zpracování dat

Pro již zmíněné získávání a zpracování dat jsme si vytvořili 3 třídy, každá má na starost zpracovat jinou žádost. Tyto tři třídy nejsou nijak závislé na zbytku programu, dali by se proto využít i v jiných aplikacích. Je vždy potřeba vytvořit nový objekt dané třídy a v konstruktoru předat potřebné informace, které jsou pro každou třídu rozdílné. Pak lze použít metody objektu, které umožňují snadné získání potřebných dat.

2.4.1 Třída WeatherByCity

Tato třída slouží pro získání aktuálního počasí ve vyhledávaném městě. V konstruktoru je nutné předat řetězec obsahující název dané obce. Třidu používáme v aplikaci pro vyhledávání obcí, které uživatel zadá do vyhledávacího pole.

Obrázek 4 - Vytvoření objektu třídy WeatherByCity a použití metody pro získání popisu aktuálního počasí

```
WeatherByCity pocasiPlzen = new WeatherByCity("Plzeň");  
String popisPocasi = pocasiPlzen.getWeatherDescription();
```

2.4.2 Třída WeatherByCoordinates

Třída WeatherByCoordinates slouží pro získání aktuálního počasí na zadaných souřadnicích (zeměpisné šířce a délce). Této třídy využíváme pro zobrazení počasí pro místo na mapě, na které klikne uživatel. Tato třída umí získat data i pro oblast, ve které se žádné město nenachází, což je u vyhledávání na mapě potřeba.

Obrázek 5 - Vytvoření objektu třídy WeatherByCoordinates a použití metody pro získání popisu aktuálního počasí

```
WeatherByCoordinates pocasiSouadnice = new WeatherByCoordinates(15.64, 54.25);  
String weatherDescription = pocasiSouadnice.getWeatherDescription();
```

2.4.3 Třída WeatherForecast

Funkcí této třídy je získání předpovědi počasí na následující dny. Tato třída je použita vždy, když uživatel použije vyhledávání i při kliknutí na mapě.

Obrázek 6 - Konstruktor třídy WeatherForecast

```
WeatherForecast predpoved = new WeatherForecast("Praha");
```

Tato třída vytvoří pole polí pro potřeby naší aplikace. Každé vnitřní pole obsahuje informace pro jeden z následujících dnů a to teplotu, čas, popis počasí a ID ikony počasí.

3 DATABÁZE A JEJÍ PROPOJENÍ S APLIKACÍ

Server OpenWeatherMap limituje počet možných odeslaných žádostí za minutu na 60 (tento limit je možné zvětšit zaplacením docela vysoké částky měsíčně⁴) a naše aplikace při načtení zobrazuje teplotu ve všech 13 krajských měst České republiky. Pokud by aplikaci začalo využívat více uživatelů byl by tento limit velice snadno dosažen. A proto jsme se rozhodli vytvořit databázi, do které se průběžně ukládají teploty v těchto městech. Při spuštění proto aplikace nemusí posílat 13 žádostí, ale stačí pouze poslat jeden dotaz databázi. Databáze běží na serveru s linuxovou distribucí Ubuntu 16.04.4 a využívá technologii MySQL, což je relační databázový systém vytvořený společností Oracle⁵.

3.1 Vytvoření databáze

Databázi jsme vytvořili v programu MySQL Workbench⁶. Databáze obsahuje tabulku s názvy měst a teplotami v nich.

Obrázek 7 - Část databázové tabulky

nazev	teplota
brno	-1.07
ceske budejovice	-3.39
hradec kralove	0.47

3.2 Připojení k databázi

Pro připojení jsme si vytvořili třídu MySQLConnect, k získání dat z databáze pak slouží metoda getData() v hlavní třídě PocasiMain.

⁴ Price – OpenWeatherMap. Current weather and forecast – OpenWeatherMap [online]

⁵ JUN, Adam. MySQL databáze – český manuál. Junext [online]

⁶ MySQL :: MySQL Workbench. MySQL [online]

3.2.1 Třída MySQLConnect

Tato třída obsahuje metody k navázání spojení s databází a pro ukončení spojení. Připojení k databázi je zabezpečeno uživatelským jménem a heslem, třída přihlášení také zajišťuje. Tato třída využívá třídy z knihovny java.sql⁷.

Obrázek 8 - Ukázka třídy MySQLConnect – metoda pro připojení k databázi

```
/**
 * Metoda sloužící k vytvoření spojení s databází
 * @return objekt připojení
 */
public Connection connect() {
    if (connection == null) {
        try {
            Class.forName(DATABASE_DRIVER);
            connection = DriverManager.getConnection(DATABASE_URL, getProperties());
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
    }
    return connection;
}
```

3.2.2 Metoda getData

Tato metoda pomocí objektu třídy MySQLConnect posílá dotaz databázi a vytvoří kolekci HashMap, kterou následně naplní názvy měst a teplotami. Tuto metodu využívá aplikace při vykreslování uživatelského rozhraní k zobrazení teplot v krajských městech na mapě.

⁷ Java Platform SE 8 - Package java.sql [online]

Obrázek 9 - metoda getData

```
/**
 * Metoda pro získání uložených teplot pro krajská města z databáze MySQL
 *
 * @return objekt HashMap obsahující název města a teplotu
 */
private HashMap getData(){
    HashMap<String, Double> result = new HashMap<>();
    MySQLConnect mysqlConnect = new MySQLConnect();
    String sql = "SELECT * FROM `mesta`";
    try {
        PreparedStatement statement = mysqlConnect.connect().prepareStatement(sql);
        ResultSet rs = statement.executeQuery();
        while (rs.next()) {
            String nazev = rs.getString("nazev");
            double tepl = rs.getDouble("teplota");
            result.put(nazev, tepl);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        mysqlConnect.disconnect();
    }
    return result;
}
```

3.3 Skript pro aktualizaci teplot v databázi

Pro aktualizaci dat v databázi slouží skript v programovacím jazyce Python. Skript každých 10 minut zjistí ze serveru OpenWeatherMap teplotu pro všechna města a aktualizuje databázi, aby to nemusela provádět sama aplikace při každém spuštění. Tento skript běží na stejném serveru jako databáze.

Obrázek 10 - skript pro aktualizaci databáze

```
import pyowm
import time
import mysql.connector
import datetime

mydb = mysql.connector.connect(
    host="139.59.139.93",
    user="app",
    passwd="pocasi",
    database="pocasi"
)

cursor = mydb.cursor()

#mesto, teplota (min, max), tlak, vlhkost, popis, ikona, rychlost_vetru, lat, long, vychod, zapad_slunce, cas(dt)

owm = pyowm.OWM("353b379036c2911483cfb9147c1ec9f0")

cities = ["praha", "brno", "ostrava", "plzen", "ceske budejovice", "hradec kralove", "jihlava", "karlovy vary", "liberec"]

while True:
    print(datetime.datetime.now().strftime("%d %B %H:%M"))
    for city in cities:
        try:
            weather = owm.weather_at_place(city + ",CZ")
            w = weather.get_weather()
            print("{}: {} °C".format(city, str(w.get_temperature('celsius')['temp'])))
            sql = "UPDATE mesta SET teplota = %s WHERE nazev = %s"
            #sql = "INSERT INTO mesta (nazev, teplota) VALUES (%s, %s)"
            val = (w.get_temperature('celsius')['temp'], city)
            cursor.execute(sql, val)
            time.sleep(2)
        except Exception as e:
            print("Chyba: " + str(e))

    mydb.commit()
    time.sleep(600)
```

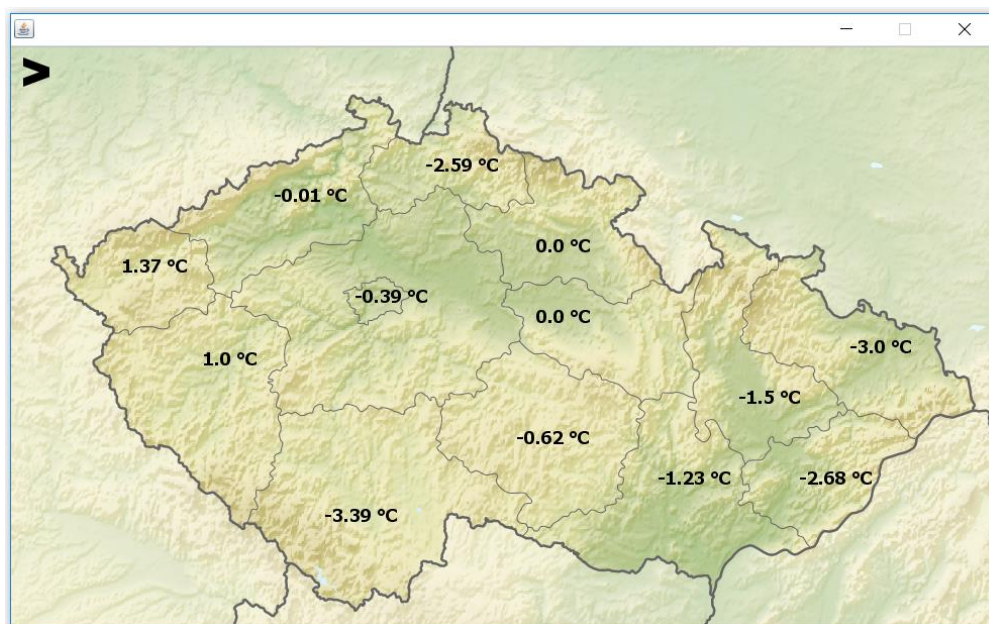
4 UŽIVATELSKÉ ROZHRAŇÍ

Uživatelské rozhraní je velice důležitou částí aplikace. Rozhodli jsme se pro vytvoření hlavního okna se zobrazením mapy a menšího okna pro zobrazování detailních informací o počasí. Pro vytvoření grafického rozhraní jsme využili možnosti návrhu uživatelského rozhraní ve vývojovém prostředí NetBeans, které využívá knihovny Swing a AWT. Obě okna vytváří třída PocasiMain.

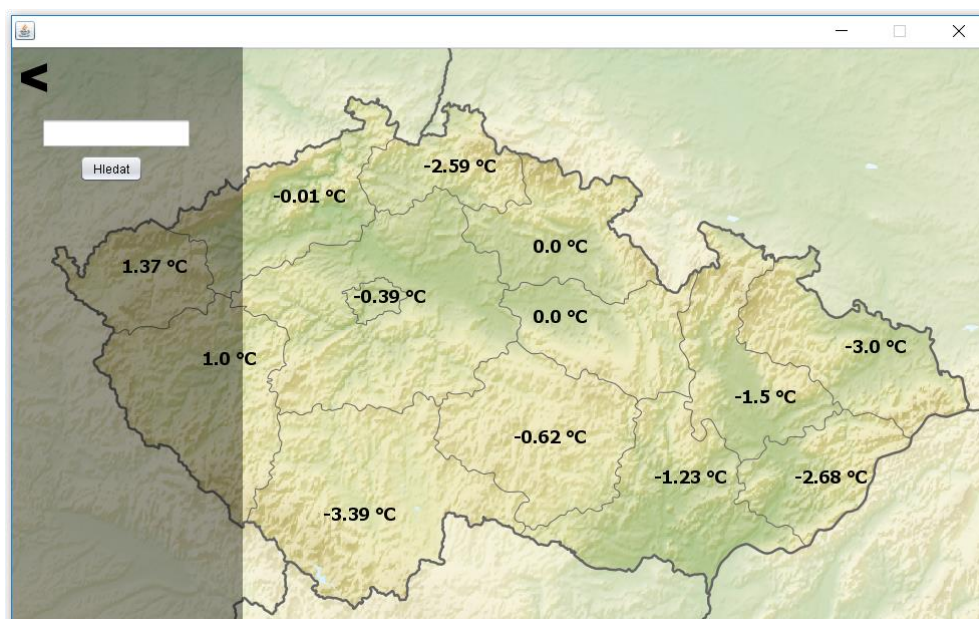
4.1 Hlavní okno

Hlavní okno obsahuje mapu, na které se na zobrazují teploty v krajských městech a vlevo nahoře se nachází tlačítko pro zobrazení vyhledávacího panelu. Po kliknutí na jakékoliv místo na mapě se provede zobrazení počasí v dané oblasti.

Obrázek 11 - Hlavní okno



Obrázek 12 - Hlavní okno s otevřeným vyhledávacím panelem.



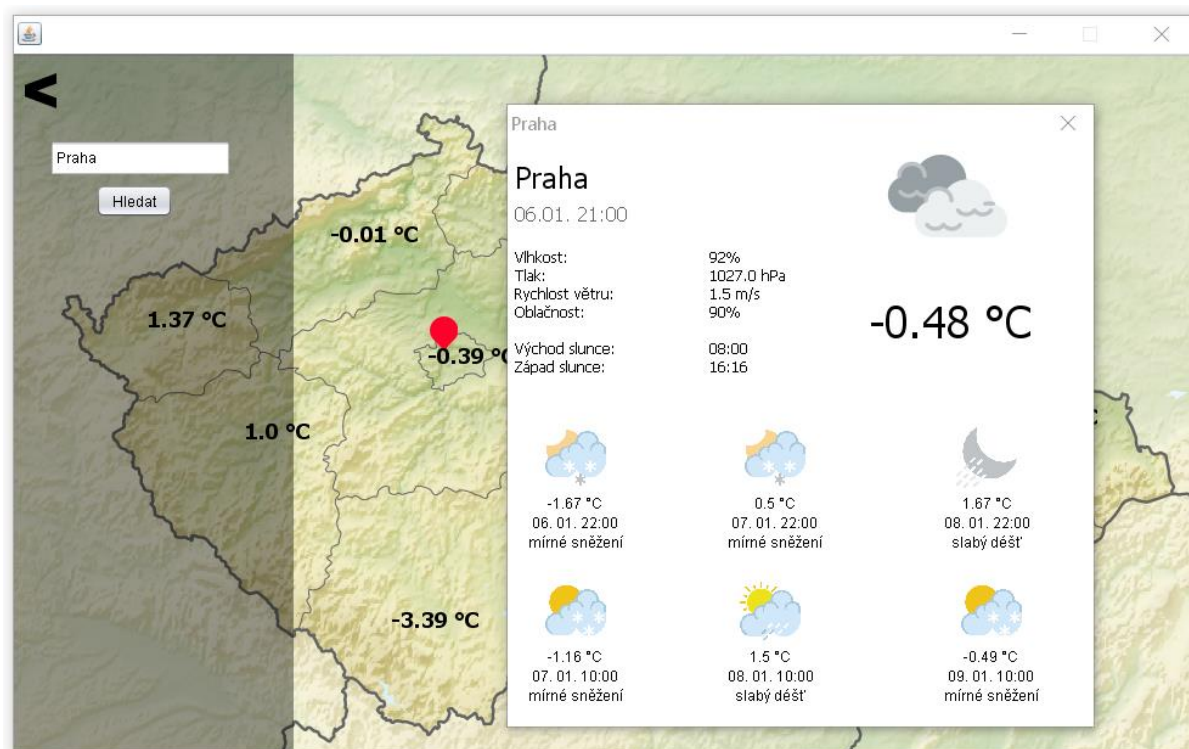
Celé okno je objektem třídy JFrame. Mapa a teploty měst jsou umístěna v objektu JLabel. Obrázek mapy podléhá licenci Creative Commons, smíme ho tedy používat za podmínky uvedení autora, kterým je Виктор В⁸. Vyhledávací panel je objekt JPanel a obsahuje textové pole pro zadávání textu (JTextField) a tlačítko pro inicializaci vyhledávání (JButton).

⁸ Виктор В, Soubor: Relief Map of Czech Republic.png – Wikipedie. [online]

4.2 Menší okno pro zobrazování výsledků

Menší okno slouží pro zobrazování výsledků vyhledávání. Všechny objekty v tomto okně jsou JLabel. Okno obsahuje data jako teplotu, vlhkost, oblačnost, západ a východ slunce a také předpověď na další 3 dny po 12 hodinách. Pro vykreslení tohoto okna slouží metoda showInfo ve třídě PocasiMain.

Obrázek 13 - Menší okno s detailním počasím pro Prahu



4.3 Vypočtení souřadnic místa po kliknutí

Po kliknutí na mapu potřebujeme zjistit souřadnice daného místa, abychom mohli uživateli počasí v tomto místě zobrazit. Vypočítali jsme si proto jaký rozdíl v reálných souřadnicích představuje jeden bod mapy. Poté jsme zjistili skutečné souřadnice jednoho bodu mapy a podle toho můžeme dopočítat souřadnice jakéhokoliv jiného bodu. K tomu slouží metoda getCoordinates, které se jako parametry předává pozice bodu, na který uživatel kliknul. Opačný postup výpočtu používáme v metodě coordinatesToXY, která nám slouží k zobrazení ukazatele pozice hledaného města na mapě.

4.4 Ikony počasí

Použité ikony jsou z webové stránky FlatIcon⁹ a podléhají licenci Creative Commons, smíme je tedy používat za podmínky uvedení autora, kterým je Smashicons⁹. Ikony zpracovává třída IkonyPocasi. Tato třída nám vždy vytvoří danou ikonu podle ID počasí, které je obsažené v odpovědi serveru.

Obrázek 14 - Ukázka ikon počasí



⁹ Freepik CompanyS.L. Weather Set. FlatIcon [online]

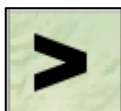
5 NÁVOD K POUŽITÍ PROGRAMU

Použití aplikace je velice snadné. Pro zobrazení počasí v požadované lokaci lze využít buďto vyhledávací panel anebo rovnou kliknout na mapu, na místo, kde se lokace nachází.

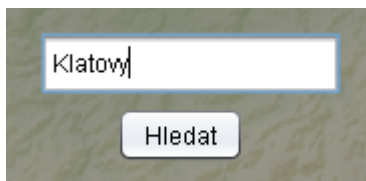
5.1 Využití vyhledávacího pole

Po otevření aplikace stačí kliknout na tlačítko vlevo nahoře. Otevře se vyhledávací panel. Hledané město zadáme do textové pole a stiskneme tlačítko „Hledat“. Zobrazí se okno s informacemi o počasí pro město a na mapě se ukazatelem označí poloha města.

Obrázek 15 - Tlačítko pro zobrazení vyhledávacího panelu



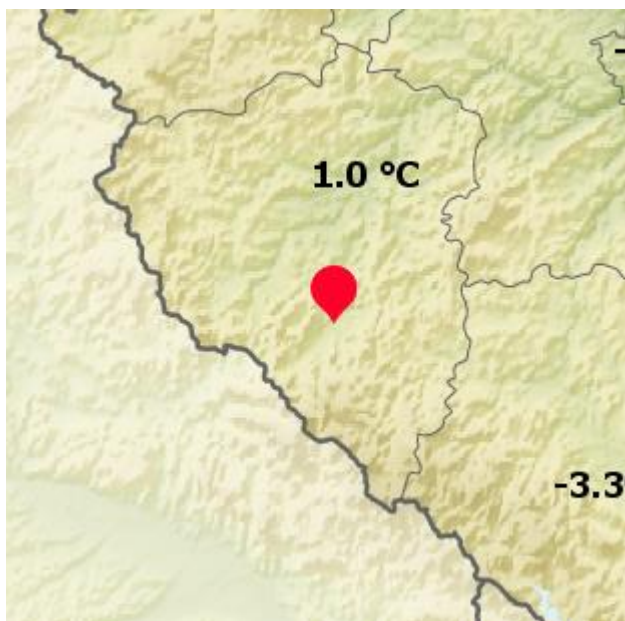
Obrázek 16 - Napsání textu do vyhledávacího pole



Obrázek 17 - Zobrazení informací o počasí



Obrázek 18 - Označení lokace města na mapě



5.2 Kliknutí na pozici na mapě

Druhou možností je kliknutí na mapu na místo, kde se lokace nachází. Takto vyhledávat lokaci může být nepřesné. Tento způsob vyhledávání slouží hlavně k rychlému zjištění počasí v nějaké oblasti. Zobrazené výsledky jsou stejné jako při předchozím způsobu vyhledávání.

6 ZÁVĚR

Tato práce měla za cíl popsat vytvoření aplikace pro zobrazování a vyhledávání informací o počasí. Podařilo se nám úspěšně zrealizovat odesílání žádostí serveru, zpracovat data o počasí a vytvořit databázi a skript, který v ní aktualizuje data a hlavně také grafické uživatelské prostředí.

Jedním z bodů, které se nacházejí v zadání je vytváření grafů. Původním záměrem bylo vytvořit pouze aplikaci s oknem pro vyhledávání a zobrazení dat a grafů (bez mapy). Po zpětné vazbě ze strany komise předkola SOČ, kde jsme tuto práci prezentovali, jsme se rozhodli tuto část aplikace nahradit. Usoudili jsme, že pro běžného uživatele jsou grafy nepotřebné a tak jsme je nahradili oknem s mapou, které má pro uživatele mnohem více využití.

Aplikace je volně dostupná ve službě GitHub v podobě kódu a spustitelného souboru. Na aplikaci budeme dále pracovat a přidat nové funkce.

7 SHRUTÍ

Tato práce řeší vývoj aplikace pro vyhledávání informací o počasí. Hlavními cíli této práce jsou vytvoření připojení k serveru a získávání dat o počasí, které server poskytuje, vytvoření databáze, skriptu, který bude databázi aktualizovat a vytvoření uživatelského rozhraní. Výstupem této práce je funkční program pro operační systém Windows, napsaný v programovacím jazyce Java. Program je volně dostupný ve službě GitHub.

This work describes the development of an application for querying information about weather. The main goals of this work are establishing connection to the server, retrieving data which the server provides, the creation of a database, creation of scripts for updating the database, and creating a graphical user interface. The result of this work is a functional application for Windows operating system, written in programming language Java. Application is freely available on GitHub.

8 POUŽITÉ ZDROJE INFORMACÍ

- [1] OpenWeather, *Weather API – OpenWeatherMap*. [online]. Copyright © 2012 [cit. 2018-12-05]. Dostupné z: <https://openweathermap.org/api>
- [2] Promotic, *JSON – textový formát pro výměnu dat*. [online]. [cit. 2018-12-15]. Dostupné z: <https://www.promotic.eu/cz/pmdoc/Appendix/FileFmt/json/FmtJSON.htm>
- [3] stleary, *GitHub – stleary/JSON-java: A reference implementation of a JSON package in Java*. [online]. Copyright © 2019 [cit. 2018-12-15]. Dostupné z: <https://www.github.com/stleary/JSON-java>
- [4] OpenWeather, *Price – OpenWeatherMap*. [online]. Copyright © 2012 [cit. 2018-12-15]. Dostupné z: <https://openweathermap.org/price>
- [5] JUN, Adam. *MySQL databáze – český manuál*. [online]. [cit. 2018-12-27]. Dostupné z: <https://www.junext.net/mysql/>
- [6] Oracle Corporation, *MySQL Workbench*. [online]. Copyright © 2019, Oracle Corporation and [cit. 2018-12-27]. Dostupné z: <https://www.mysql.com/products/workbench/>
- [7] Oracle Corporation, *Java Platform SE 8 - Package java.sql* [online]. [cit. 2018-12-27] Dostupné z: <https://docs.oracle.com/javase/8/docs/api/index.html?java/sql/package-summary.html>
- [8] Виктор В, *Soubor: Relief Map of Czech Republic.png – Wikipedie*. [online]. [cit. 2019-01-03] Dostupné z: https://cs.wikipedia.org/wiki/Soubor:Relief_Map_of_Czech_Republic.png
- [9] Freepik CompanyS.L. *Weather Set*. [online]. [cit. 2019-01-03]. Dostupné z: <https://www.flaticon.com/packs/weather-set-2>