**Оформление результата**

* Отправляется как один zip-архив с git-репозитарием или ссылка на репозитарий на github.com.
* В проекте в папке docs документация (при необходимости), в папке liquibase Liquibase и миграции к нему.
* Проект собирается Maven.
* Для тестового задания предполагается логин/пароль MySQL root/root, имя базы данных distance-calculator.
* Data source в WildFly java:/magenta/datasource/test-distance-calculator .

**Overview**

Design and implement web service (REST) application for distance calculation:

* Database holds two entities:
  + City
    - Name
    - Latitude
    - Longitude
  + Distance
    - From city
    - To city
    - Distance
* Application should make it possible to calculate the distance in two ways:
  + “Crowflight” (straight distance) between cities. Lookup formula for distance calculation on the sphere in the internet.
  + Lookup distance between two cities via “distance matrix” (distance table in the database)
* API has 3 endpoints:
  + List of all cities in the DB. Fields:
    - ID
    - Name
  + Calculate distance
    - Input:
      * Calculation Type: <Crowflight, Distance Matrix, All>
      * From City: <List of cities>
      * To City: <List of Cities>
    - Output:
      * Results: all distance calculation results as requested
  + Upload data to the DB. Uploads XML file with cities and distances into the application. Application parses it and stores it into the database.
    - Input:
      * Multipart/form-data form submission with single “File” input.
    - Output:
      * HTTP response code 200 without body

**Tools/Libraries**

* IDEA Community Edition
* Git
* Maven
* MySQL DB
* Liquibase для миграции к структуре DB
* Java 8 (можно использовать более свежие версии)
* JAXB

Выбрать одно из двух:

* JEE 8 + WildFly 14
* Spring + Tomcat

**Requirements**

* Use Java exceptions to indicate that distance cannot be calculated (for example, it is not in the distance table).
* Make sure you are up to speed on the following Java basics: interfaces, classes, inheritance, overriding, collections.

**Optional Requirements**

The following requirements are optional. Please work on them if you have capacity (after you submitted result of your assignment back to us):

* Test if your application would scale to 10.000 cities and 1.000.000 entries in distance table (assume that distance is defined only for some cities in the distance matrix). Test if XML file of this size can be loaded fine. Fix scale/performance issues if they would appear.