Util

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package fr.insalyon.dasi.gustatif.util;

import java.util.*;
import javax.mail.*;
import javax.mail.internet.*;

/**
 *
 * @author Nico
 */
public class MailSender {

    public static void main(String [] args)
    {
        MailSender ms = new MailSender("gustatif.b3125@gmail.com","B3125-2016");
        ms.sendMail("n.gripont@gmail.com","Java Test Code","Does it work?");
    }

    final String username;
    final String password;

    public String getUsername() {
        return username;
    }

    public MailSender(String username, String password)
    {
        this.username = username;
        this.password = password;
    }
    public boolean sendMail(String recipientMail, String subject, String body)
    {
        boolean result = false;

        Properties props = new Properties();
        props.put("mail.smtp.auth", "true");
        props.put("mail.smtp.starttls.enable", "true");
        props.put("mail.smtp.host", "smtp.gmail.com");
        props.put("mail.smtp.port", "587");

        Session session = Session.getInstance(props,
        new javax.mail.Authenticator() {
              protected PasswordAuthentication getPasswordAuthentication() {
                      return new PasswordAuthentication(username,
                      password);
              }
        });

        try {
            Message message = new MimeMessage(session);
            message.setFrom(new InternetAddress());
            message.setRecipients(Message.RecipientType.TO,
                    InternetAddress.parse(recipientMail));
            message.setSubject(subject);
            message.setText(body);

            Transport.send(message);
            result = true;

        } catch (MessagingException e) {
            throw new RuntimeException(e);
        }
        return result;
    }
}

```

```java
package fr.insalyon.dasi.gustatif.util;

import com.google.maps.DirectionsApi;
import com.google.maps.DirectionsApiRequest;
import com.google.maps.GeoApiContext;
import com.google.maps.GeocodingApi;
import com.google.maps.model.DirectionsResult;
import com.google.maps.model.DirectionsRoute;
import com.google.maps.model.GeocodingResult;
import com.google.maps.model.LatLng;
import com.google.maps.model.TravelMode;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author DASI Team
 */
public class GeoTest {
        final static String MA_CLÉ_GOOGLE_API = "AIzaSyAQ1rgsSDdetI6uhC9egwf_OqdDprHwB-g";

//    final static String MA_CLÉ_GOOGLE_API = "AIzaSyAhf3JleYpal9S-xouJYH8lf7Dvz5Y2Nko";


    final static GeoApiContext MON_CONTEXTE_GEOAPI = new GeoApiContext().setApiKey(MA_CLÉ_GOOGLE_API);

    public static LatLng getLatLng(String adresse) {
        try {
            GeocodingResult[] results = GeocodingApi.geocode(MON_CONTEXTE_GEOAPI, adresse).await();

            return results[0].geometry.location;

        } catch (Exception ex) {
            return null;
        }
    }

    public static double toRad(double angleInDegree) {
        return angleInDegree * Math.PI / 180.0;
    }

    public static double getFlightDistanceInKm(LatLng origin, LatLng destination) {

        // From: http://www.movable-type.co.uk/scripts/latlong.html
        double R = 6371.0; // Average radius of Earth (km)
        double dLat = toRad(destination.lat - origin.lat);
        double dLon = toRad(destination.lng - origin.lng);
        double lat1 = toRad(origin.lat);
        double lat2 = toRad(destination.lat);

        double a = Math.sin(dLat / 2.0) * Math.sin(dLat / 2.0)
                + Math.sin(dLon / 2.0) * Math.sin(dLon / 2.0) * Math.cos(lat1) * Math.cos(lat2);
        double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1.0 - a));
        double d = R * c;

        return d;
    }

    public static Double getTripDurationByBicycleInMinute(LatLng origin, LatLng destination, LatLng... steps) {
        return getTripDurationOrDistance(TravelMode.BICYCLING, true, origin, destination, steps);
    }

    public static Double getTripDistanceByCarInKm(LatLng origin, LatLng destination, LatLng... steps) {
        return getTripDurationOrDistance(TravelMode.DRIVING, false, origin, destination, steps);
    }

    public static Double getTripDurationOrDistance(TravelMode mode, boolean duration,
            LatLng origin, LatLng destination, LatLng... steps) {

        DirectionsApiRequest request = DirectionsApi.getDirections(MON_CONTEXTE_GEOAPI,
                origin.toString(), destination.toString());
        request.mode(mode);
        request.region("fr");

        if (steps.length > 0) {

            String[] stringSteps = new String[steps.length];
            for (int i=0; i<steps.length; i++) {
                stringSteps[i] = steps[i].toString();
            }

            request.waypoints(stringSteps);
        }

        double cumulDistance = 0.0;
        double cumulDuration = 0.0;

        try {
            DirectionsResult result = request.await();
            DirectionsRoute[] directions = result.routes;

            for (int legIndex = 0; legIndex < directions[0].legs.length; legIndex++) {

                cumulDistance += directions[0].legs[legIndex].distance.inMeters / 1000.0;
```

```java
                cumulDuration += Math.ceil(directions[0].legs[legIndex].duration.inSeconds / 60.0);
            }

        } catch (Exception ex) {
            return null;
        }

        if (duration) {
            return cumulDuration;
        }
        else {
            return cumulDistance;
        }
    }

    public static void main(String[] args) {
        Logger logger = Logger.getLogger(GeoApiContext.class.getName());
        logger.setLevel(Level.WARNING);

        String adresse1 = "7 Avenue Jean Capelle Ouest, Villeurbanne";
        LatLng coords1 = getLatLng(adresse1);
        System.out.println("Lat/Lng de Adresse #1: " + coords1);

        String adresse2 = "37 Avenue Jean Capelle Est, 69100 Villeurbanne";
        LatLng coords2 = getLatLng(adresse2);
        String adresse3 = "61 Avenue Roger Salengro, Villeurbanne";
        LatLng coords3 = getLatLng(adresse3);

        Double duree = getTripDurationByBicycleInMinute(coords1, coords3, coords2);
        System.out.println("Durée de Trajet à Vélo de Adresse #1 à Adresse #3 en "
                + "passant par Adresse #2: " + duree + " min");

        Double distance = getTripDistanceByCarInKm(coords1, coords3);
        System.out.println("Distance en Voiture de Adresse #1 à Adresse #3 "
                + "(trajet direct par la route): " + distance + " km");

        Double distanceVolDOiseau = getFlightDistanceInKm(coords1, coords3);
        System.out.println("Distance à Vol d'Oiseau de Adresse #1 à Adresse #3"
                + " (distance géographique): " + distanceVolDOiseau + " km");
    }
}
```

```java
package fr.insalyon.dasi.gustatif.util;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.List;

/**
 *
 * @author DASI Team
 */
public class Saisie {

    public static String lireChaine(String invite) {
        String chaineLue = null;
        System.out.print(invite);
        try {
            InputStreamReader isr = new InputStreamReader(System.in);
            BufferedReader br = new BufferedReader(isr);
            chaineLue = br.readLine();
        } catch (IOException exception) {
            exception.printStackTrace(System.err);
        }
        return chaineLue;

    }

    public static Integer lireInteger(String invite) {
        Integer valeurLue = null;
        try {
            valeurLue = new Integer(lireChaine(invite));
        } catch (java.lang.NumberFormatException e) {
            System.out.println("erreur de saisie");
            valeurLue = lireInteger(invite);
        }
        return valeurLue;
    }

    public static Integer lireInteger(String invite, List<Integer> valeursPossibles) {
        Integer valeurLue = null;
        try {
            valeurLue = new Integer(lireChaine(invite));
            if (!(valeursPossibles.contains(valeurLue))) {
                throw new Exception();
            }
        } catch (Exception e) {
            System.out.println("erreur de saisie");
            valeurLue = lireInteger(invite, valeursPossibles);
        }
        return valeurLue;
    }

    public static void main(String[] args) {
        System.out.println("Bonjour !");
        String nom = Saisie.lireChaine("Entrez votre nom: ");
        System.out.println("Bonjour, " + nom + " !");
        Integer age = Saisie.lireInteger("Entrez votre âge: ");
        System.out.println("Vous avez " + age + " ans.");
        System.out.println("Au revoir !");
    }
}
```