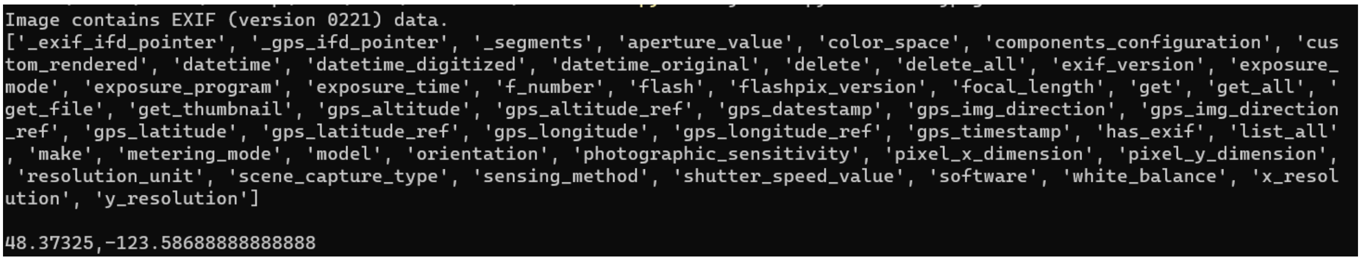
LABORATORIJSKA VJEŽBA 4

Jakov Melvan

Kod:

```
import os, sys, optparse
from exif import Image
import webbrowser
from PyPDF2 import PdfFileReader, PdfFileWriter


def convertGPScoordinate(coordinate, coordinate_ref):
    decimal_degrees = coordinate[0] + \
                      coordinate[1] / 60 + \
                      coordinate[2] / 3600

    if coordinate_ref == "S" or coordinate_ref == "W":
        decimal_degrees = -decimal_degrees

    return decimal_degrees

def figMetaData(file_path):
    img_doc = Image(open(file_path, "rb"))

    if not img_doc.has_exif:
        sys.exit(f"Image does not contain EXIF data.")
    else:
        print(f"Image contains EXIF (version {img_doc.exif_version}) data.")

    print(f"{dir(img_doc)}\n")

    latitude = convertGPScoordinate(img_doc.gps_latitude, 
img_doc.gps_latitude_ref)
    latitude = str(latitude)
    longitude = convertGPScoordinate(img_doc.gps_longitude, 
img_doc.gps_longitude_ref)
    longitude = str(longitude)

    print(latitude + "," + longitude)

    url = "http://www.google.com/maps/place/" + latitude + "," + longitude

    webbrowser.open_new_tab(url)


def pdfMetaData(file_path):
    pdf_doc = PdfFileReader(open(path, "rb"))
    if pdf_doc.isEncrypted:
        try:
            if pdf_doc.decrypt("PASSWORD_GOES_HERE") != 1:
                sys.exit("target pdf document is encrypted")
        except:
            sys.exit("target pdf document is encrypted")
    pdfWriter = PdfFileWriter()
    for pageNum in range(pdf_doc.numPages):
        pdfWriter.addPage(pdf_doc.getPage(pageNum))
    resultPdf = open('decrypted_output.pdf', 'wb')
    pdfWriter.write(resultPdf)
    resultPdf.close()


if __name__ == "__main__":
    parser = optparse.OptionParser("Usage: python <script_name> -f <file>")
    parser.add_option("-f", dest="file", type="string", help="please provide full
path to the document")

    (options, args) = parser.parse_args()

    path = options.file
    if not path:
        print("please provide full path to the document")
        sys.exit(parser.usage)

    if any(path.endswith(ext) for ext in (".jpg", ".bmp", ".jpeg",)):
        figMetaData(path)
    elif path.endswith(".pdf"):
        pdfMetaData(path)
    else:
        print("File extension not supported/recognized... Make sure the file has
the correct extension...")
```



```
Image contains EXIF (version 0221) data.
['_exif_ifd_pointer', '_gps_ifd_pointer', '_segments', 'aperture_value', 'color_space', 'components_configuration', 'cus
tom_rendered', 'datetime', 'datetime_digitized', 'datetime_original', 'delete', 'delete_all', 'exif_version', 'exposure_
mode', 'exposure_program', 'exposure_time', 'f_number', 'flash', 'flashpix_version', 'focal_length', 'get', 'get_all', '
get_file', 'get_thumbnail', 'gps_altitude', 'gps_altitude_ref', 'gps_datestamp', 'gps_img_direction', 'gps_img_direction
_ref', 'gps_latitude', 'gps_latitude_ref', 'gps_longitude', 'gps_longitude_ref', 'gps_timestamp', 'has_exif', 'list_all'
, 'make', 'metering_mode', 'model', 'orientation', 'photographic_sensitivity', 'pixel_x_dimension', 'pixel_y_dimension',
 'resolution_unit', 'scene_capture_type', 'sensing_method', 'shutter_speed_value', 'software', 'white_balance', 'x_resol
ution', 'y_resolution']

48.37325,-123.58688888888888
```