**Lociranje turističkih objekata na teritoriji Grada Jagodina**

Kako bi smo započeli sa radom prvenstveno treba da učitamo biblioteke, u ovom sliučaju su to **geopandas, shapely.geomety, fiona.crs, pandas** i **matplotlib**biblioteke. Učitavanje biblioteka se vrši na sledeći način:

import geopandas as gpd

import matplotlib as plt

from shapely.geometry import Point

from fiona.crs import from\_epsg

import pandas as pd

**Geopandas** biblioteka služi za učitavanje geoprostornih podataka i shapefajlova. **Matplotlib** biblioteka služi za prikazivanje grafika i vizuelizaciju podataka. Biblioteka **shapely.geomety** služi za rad sa geometrijskim objektima i prostornim podacima, kao što su tačka, linija, poligon. Konretno za potrebe ovog projekata smo uvezli samo Point, tj tačku. **Fiona.crs** biblioteka je deo biblioteke **fiona**, služi za rad sa koordinatnim sistemima. Iz ove biblioteke smo povukli funkciju from.crs koja služi za stvaranje CRS objekata, na temeljima EPSG koda. **Pandas** biblioteka služi za rad sa tabelarnim podacima, omogućava čitanje, manipulaciju i analiziranje podataka.

Shapefajlove, u ovom slučaju je to granica Jagodine, smo dobili preko QGIS softvera. QGIS je open-source softver, koji služi za kreiranje, editovanje, vizuelizaciju i analizu geoprostornih podataka. Prvenstveno, neophodno je napraviti putanju do potrebnih shapefajlova. Putanja se ubaciju uz pomoć komande:

shapefile\_path = r'putanja\_do\_shape\_fajla'

naziv\_shapefajla = gpd.read\_file(shapefile\_path)

Kada smo uvezli shapefile proveravamo njegov koordinatni sistem i postavljamo odgovarajući koordinatni sistem. Postavljanje odgovarajućegkoordinatnog sistema vrši se komandom **from\_epsg**, koja je u okviru **fiona.crs** biblioteke. To se radi preko komande:

naziv\_sahpefajla.crs

print(naziv\_sahpefajla.crs)

naziv\_shapefajla.crs = from\_epsg(izaberemo odgovarajući EPSG koordinatnog sistema)

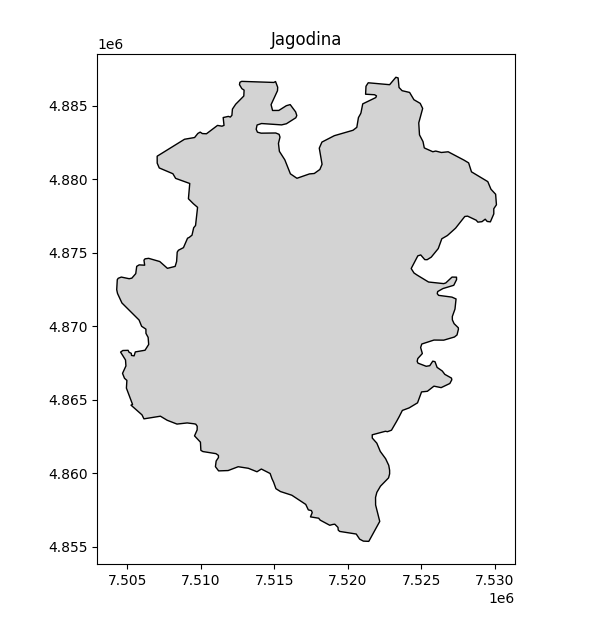
print(naziv\_shapefajla.crs)

Vizuelizacija podataka i pravljenje mape se radi uz pomoć **matplotlib** biblioteke. Komande za prikaz mape su:

naziv\_shapefajla.plot(color='izaberi\_boju', edgecolor='izaberi\_boju',)

plt.title(’Napiši naziv katre’)

plt.show()

Ubacivanje tačaka, odnosno koordinata turističkih objekata vršeno je “ručno”. Koordinate su direktono kucane u pycharm okruženju. Ovo smo uradili uz pomoć **shapely.geometry** biblitoke iz koje smo uvezli Point, na sledeći način:

oznaka = Point (x, y)

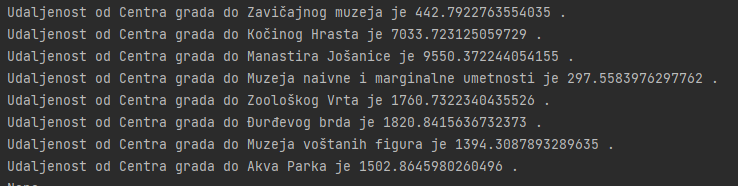
Kada smo završili sa ubacivanjem koordinata, kreira se lista “**turističke\_lokacije**“ koja sadrži parove tačaka i naziva. Svaki par je oblika („**tacka“, „naziv**“) gde je „**tacka**“ tačka definisana ranije, a „**naziv**“ je naziv koji je povezan s tom tačkom.

Stvara se prazan **GeoDataFrame** objekat „**prostor**“ s jednim kolonom „**geometry**“, koji sadrži geometriju tačaka.

Koristi se petlja **for** i funkcija **enumerat**e kako bi se iteriralo kroz listu „**turističke\_lokacije**“. Varijabla „**ID“** predstavlja indeks iteracije, a **(„tacka, naziv“**) predstavlja par tačke i naziva iz liste „**turističke\_lokacije**“. Na kraju petlje, „**prostor“ GeoDataFrame** sadrži geometrije tačaka i njihove nazive, tačke su dodijeljene u koloni „**geometry“**, a nazivi u koloni „Naziv tacke“.

Udaljenost od Centra grada do turističkih lokacija se računa uz pomoć **shapely.geometry** bibliotke, komandom **.distance**, na sledeći način:

naziv = oznaka.distance(oznaka do koje želimo da izračuna udaljenost)



Stupa se u proveru koordinatnog sistema i postavljanje odgovarajućeg koordinatnog sistema. To se radi preko komande:

naziv\_GeoDataFrame.crs

print(GeoDataFrame.crs)

naziv\_ GeoDataFrame = from\_epsg(izaberemo odgovarajući EPSG koordinatnog sistema)

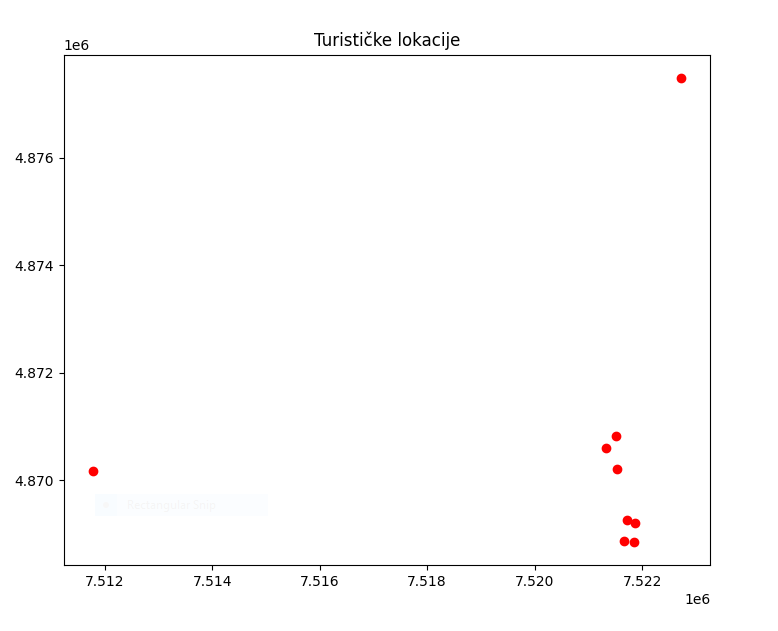
print(naziv\_ GeoDataFrame.crs)

Kao što smo prethodno naveli vizuelizacija podataka i pravljenje mape se radi uz pomoć matplotlib biblioteke. Komande za prikaz mape su:

naziv\_shapefajla.plot(facecolor='izaberi\_boju',)

plt.title(’Napiši naziv katre’)

plt.show()



Kako bi smo ova dva GeoDataFrame-a spojili koristimo **pandas** biblioteku, komandu .concat. Ova funkcija uzima listu objekata koje želimo da spojimo kao ulazni argument, u ovom slučaju, kolonu **'geometry**' iz „**prostor**“ i „**granica**“ objekata. Rezultat je novi DataFrame koji sadrži kombinovanu geometriju. Ova komanda se izvodi na sledeći način:

naziv\_novog = pd.concat([naziv\_GeoDataFrame.geometry,naziv\_shapefajla.geometry])

Kada smo ovo odradili stupamo u izradu karte kao i u prethodnim primerima. I dobijamo finalnu kartu na kojoj se nalaze svi turistički objekti na teritoriji Grada Jagodina.

