**<https://github.com/Drozd363/ArcGIS_arcpy_S9_15...git>**

**Інструменти вибору в векторних даних, перепроектування, витягування інформації з растру та представлення даних в GIS**

Необхідно написати сценарій та оформити його у вигляді інструменту GIS, який буде вирішувати наступні задачі:

1. Вибрати з шару **us\_cities.shp** міста які мають POPCLASS більше, або рівним 3, створити з вибраних об’єктів новий шейпфайл **us\_cities\_level\_3.shp**.
2. Зробити перепроектування растру **us.tmax\_nohads\_ll\_20140525\_float.tif**, згідно проекції і датуму **us\_cities.shp**. Зберегти новий растр з новим ім’ям, наприклад **us.tmax\_nohads\_ll\_20140525\_float\_NAD.tif**.
3. Обрізати **us.tmax\_nohads\_ll\_20140525\_float\_NAD.tif** за границями **us\_boundaries.shp** та зберегти його з новим ім’ям, наприклад **us.tmax\_nohads\_ll\_20140525\_float\_NAD\_extract\_by\_mask.tif**.
4. З’ясувати значення температури з растру **us.tmax\_nohads\_ll\_20140525\_float\_NAD\_extract\_by\_mask.tif** вмістах з файлу **us\_cities\_level\_3.shp**.
5. Додати в файл **us\_cities\_level\_3.shp** нове поле **TEMPERATURE**, та заповнити його значеннями температури в цих містах.
6. Додати в файл **us\_cities\_level\_3.shp** нове поле **EXCESS**, та заповнити його між середнім значенням температури за всіма містами те температурою в конкретному місті.
7. Збережіть інструмент GIS та всі проміжні файли в папку Results.

import arcpy  
import numpy as np  
arcpy.env.overwriteOutput = True  
  
cities = arcpy.GetParameterAsText(0)  
inpraster = arcpy.GetParameterAsText(1)  
clipzone = arcpy.GetParameterAsText(2)  
resultdirec = arcpy.GetParameterAsText(3)  
popsize = arcpy.GetParameterAsText(4)  
#cities = r"E:\programin\semestr2\samrob\s15\_GIS\_FILE\Programming\_in\_GIS\_2020\_L9\_s15\us\_cities1.shp"  
#inpraster = r"E:\programin\semestr2\samrob\s15\_GIS\_FILE\Programming\_in\_GIS\_2020\_L9\_s15\us.tmax\_nohads\_ll\_20140525\_float.tif"  
#clipzone = r"E:\programin\semestr2\samrob\s15\_GIS\_FILE\Programming\_in\_GIS\_2020\_L9\_s15\us\_boundaries1.shp"  
#resultdirec = r"E:\programin\semestr2\samrob\s15\_GIS\_FILE\Programming\_in\_GIS\_2020\_L9\_s15\Results"  
#popsize = 3  
arcpy.env.workspace = resultdirec  
  
# create file with selection  
arcpy.MakeFeatureLayer\_management(cities, "layercities", "POPCLASS >= " + str(popsize))  
citiesSelect = "us\_cities\_level\_{}.shp".format(str(popsize))  
arcpy.CopyFeatures\_management("layercities", citiesSelect)  
arcpy.AddMessage("Created a file with a selection of where 'POPCLASS >= {}'".format(str(popsize)))  
  
# reproject and clip by mask(polygon)  
arcpy.ProjectRaster\_management(inpraster, "reproject\_US\_raster\_temperature.tif", cities, geographic\_transform = "WGS\_1984\_(ITRF00)\_To\_NAD\_1983")  
clip = arcpy.sa.ExtractByMask("reproject\_US\_raster\_temperature.tif", clipzone)  
clip.save("reproject\_US\_raster\_temperature\_extract\_by\_mask.tif")  
arcpy.AddMessage("Made reproject and cut to the desired area")  
  
# Data from raster cells. If the data is missing, there will be an attempt to search for an uncut raster  
cursor = arcpy.da.SearchCursor(citiesSelect, "SHAPE@XY")  
try:  
 temperature = []  
 for row in cursor:  
 temp = arcpy.GetCellValue\_management("reproject\_US\_raster\_temperature\_extract\_by\_mask.tif", str(row[0][0])+' '+str(row[0][1]))  
 temperature.append(float(temp.getOutput(0).replace(",", ".")))  
 arcpy.AddMessage("Data from raster cells was found")  
except ValueError:  
 cursor.reset()  
 temperature = []  
 for row in cursor:  
 temp = arcpy.GetCellValue\_management("reproject\_US\_raster\_temperature.tif", str(row[0][0])+' '+str(row[0][1]))  
 temperature.append(float(temp.getOutput(0).replace(",", ".")))  
 arcpy.AddMessage("While trimming the raster behind the mask, the required data cells were lost."  
 " Therefore, whole raster, not truncated raster, is used to search for data."  
 " Data from raster cells was found")  
  
# Add new fields and update  
arcpy.AddField\_management(citiesSelect, "TEMPERATUR", 'FLOAT')  
arcpy.AddField\_management(citiesSelect, "EXCESS", 'FLOAT')  
with arcpy.da.UpdateCursor(citiesSelect, ["TEMPERATUR", "EXCESS"]) as Ucursor:  
 i = 0  
 for row in Ucursor:  
 row[0] = temperature[i]  
 row[1] = row[0] - np.mean(temperature)  
 Ucursor.updateRow(row)  
 i += 1  
arcpy.AddMessage('Added new fields and records added')