

Міністерство освіти і науки України
НТУУ «КПІ ім. Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра цифрових технологій в енергетиці

Лабораторна робота №4
з дисципліни «Геоінформаційні системи в енергетиці»
Тема «Географічний аналіз за допомогою ArcGis
ArcToolBox. Побудова моделей в ModelBuilder»
Варіант №22

Студента 2-го курсу НН ІАТЕ гр. ТР-12
Ковальова Олександра
Перевірив: ст. в. Гурін Артем Леонідович

Мета роботи. Отримати навички використання інструментів геообробки ArcToolBox, навчитися створювати моделі в ModelBuilder, конвертувати файли інших ГІС.

Хід роботи.

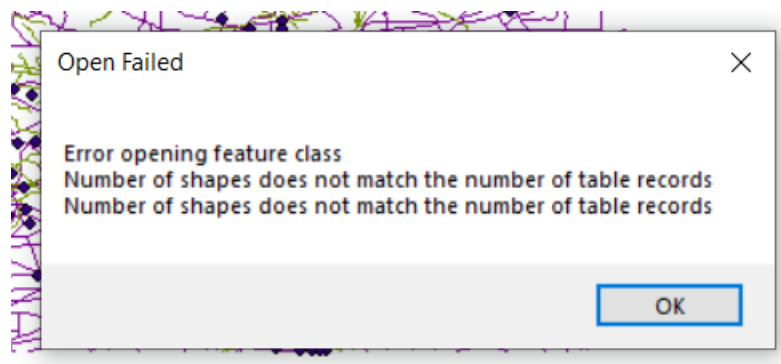
Завдання (22): Результат віднімання від буферу радіусом 200 м об'єднання (таблиць E, C) буфера таблиці D радіусом 600 м.

Файли:

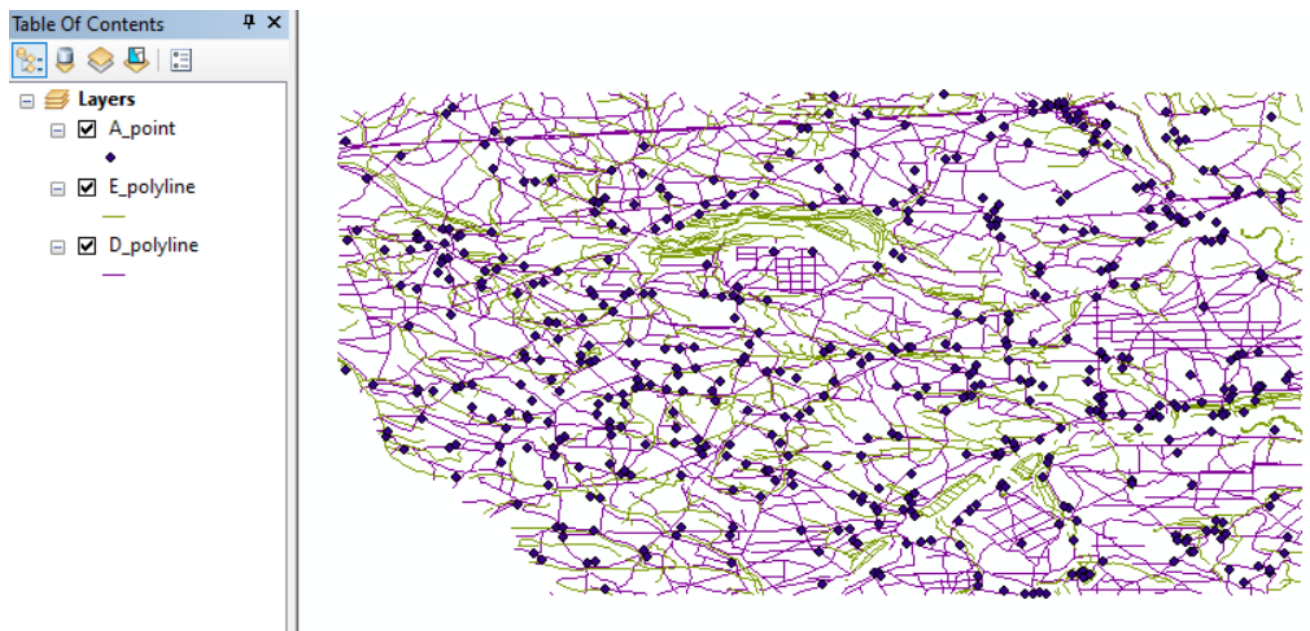
Назва файлу	Типи об'єктів	Образ
A	Точки	Техногенні елементи
D	Полілінії	Шляхи
E	Полілінії	Гідрографія

Частина перша. Виконати вручну завдання.

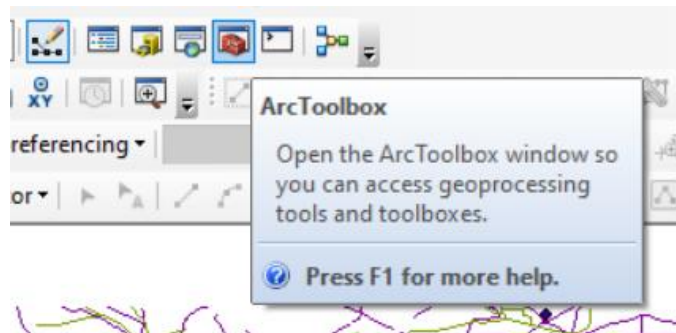
Шейп-файл C не відкривається, тому замість нього потрібно взяти файл A:



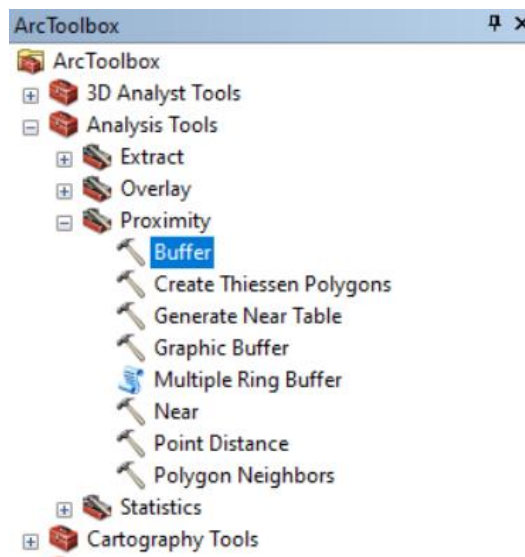
Відкриваємо шейп-файли A, D і E:



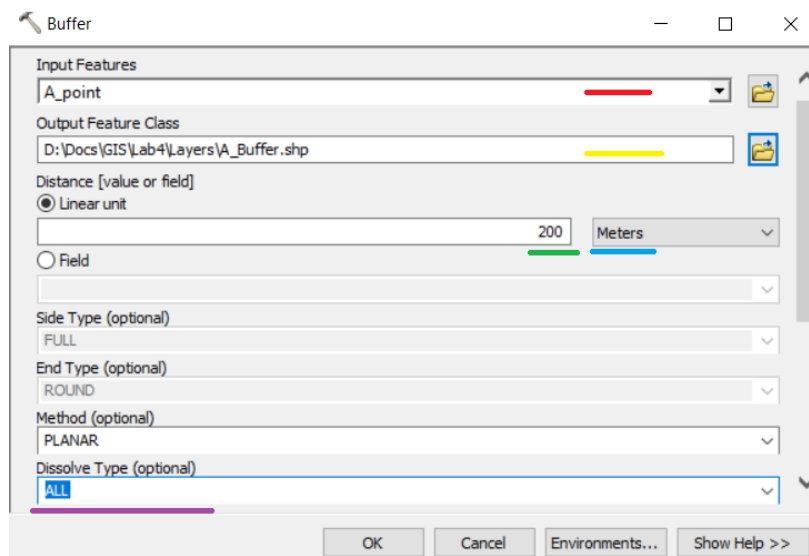
Вмикаємо панель ArcToolbox:



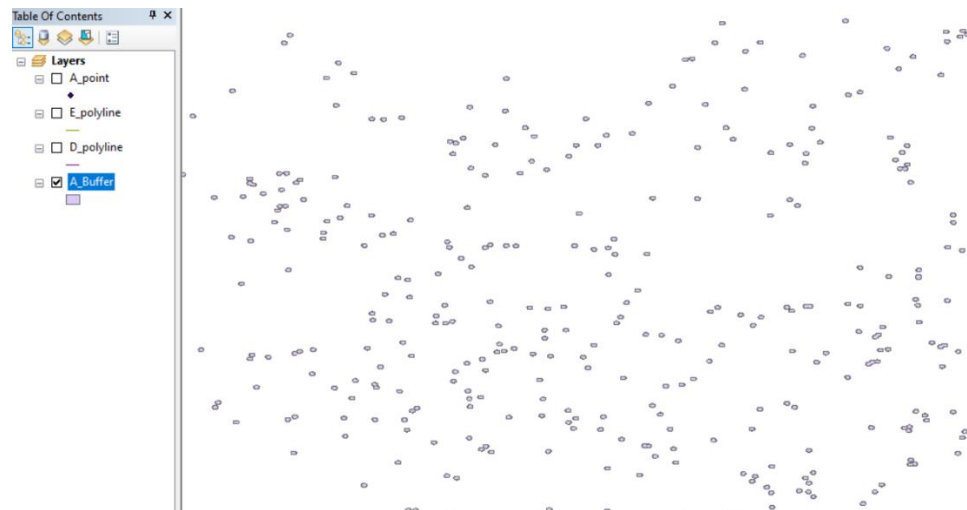
Зробимо буфер таблиці А. Спочатку, знаходимо “Buffer” в ArcToolbox:



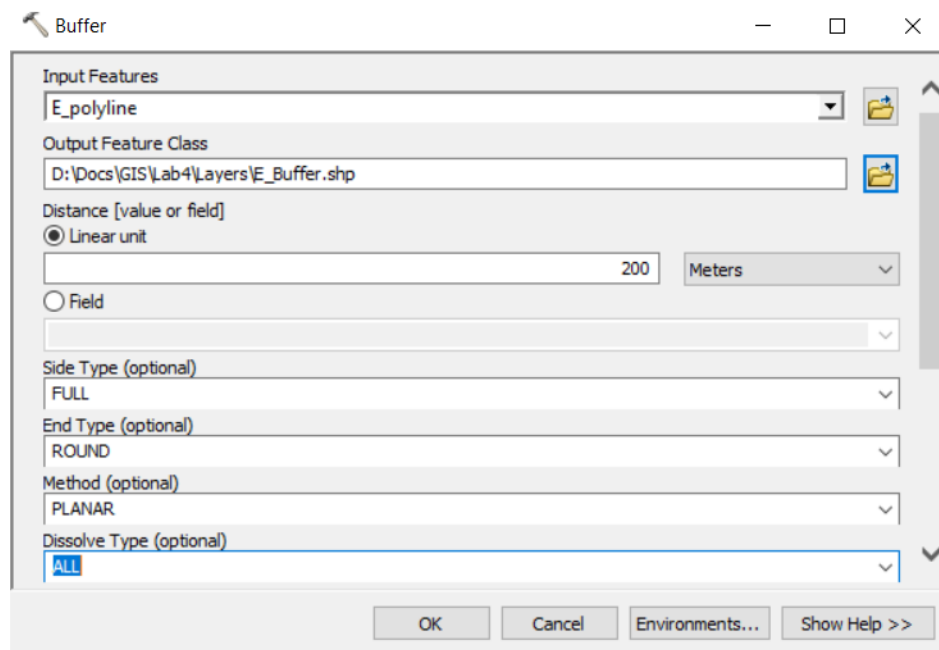
В “Input Features” обираємо шар, над яким буде виконуватись операція. “Output Feature Class” – майбутній результат. Вказуємо, що буфер повинен бути розміром в 200 метрів (зеленим, також треба обрати одиниці виміру – метри, синім). Dissolve type – ALL, це дозволить об’єднати буфери, які перетинаються.



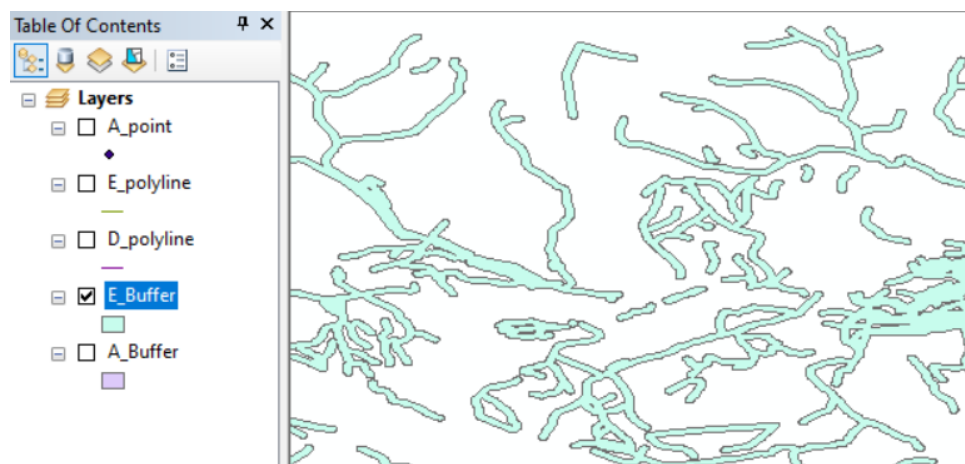
Вигляд буферу А:



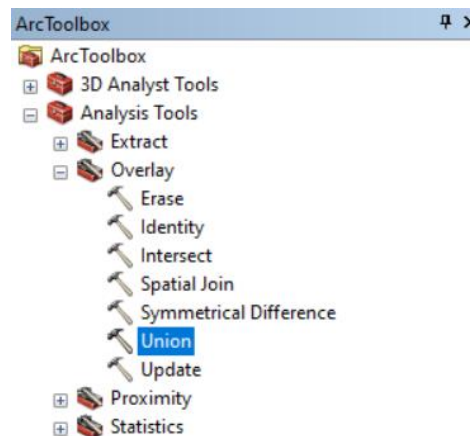
Те ж саме повторюємо з шаром Е:



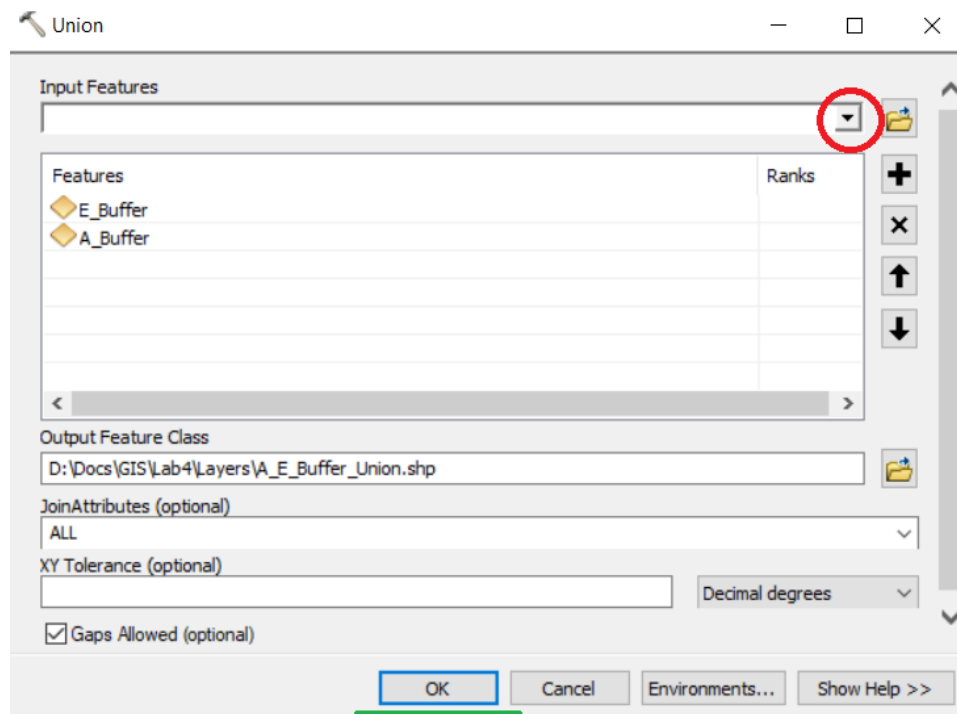
Отриманий буфер:



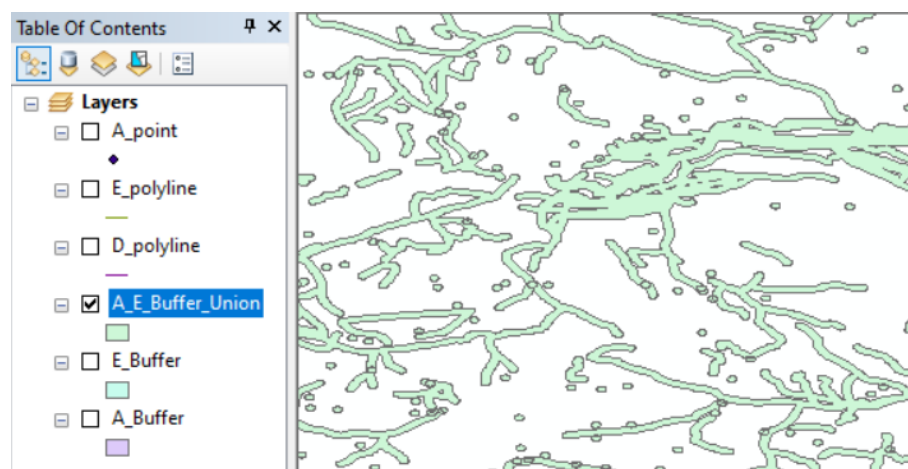
Для об'єднання буферів використовується інструмент Union:



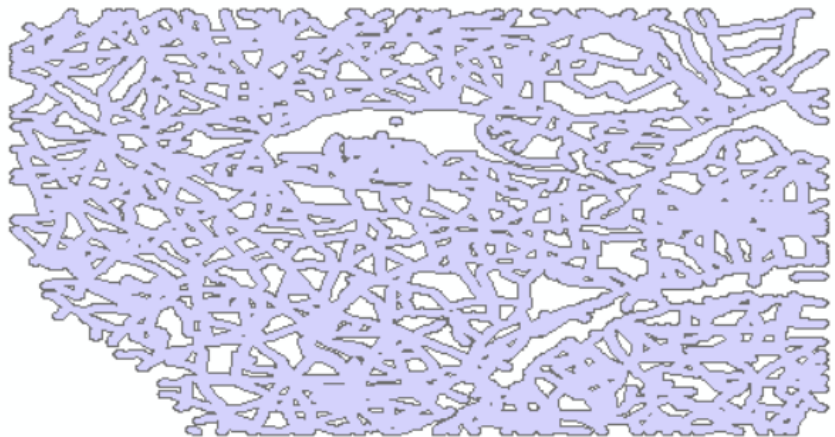
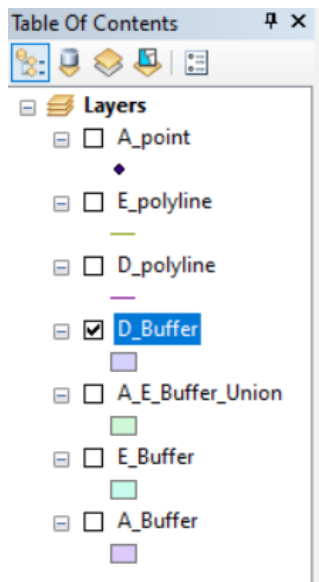
Об'єднуємо:



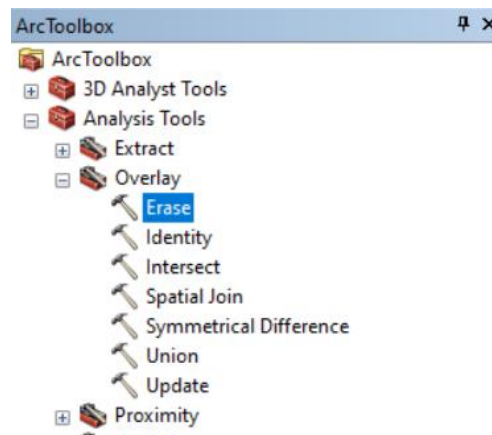
Результат:



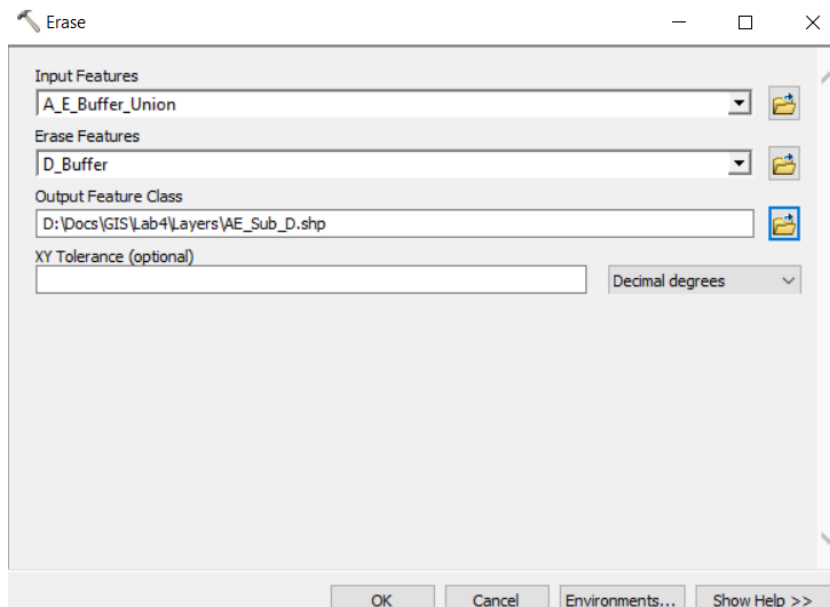
Створюємо буфер шару D радіусу 600 м.:



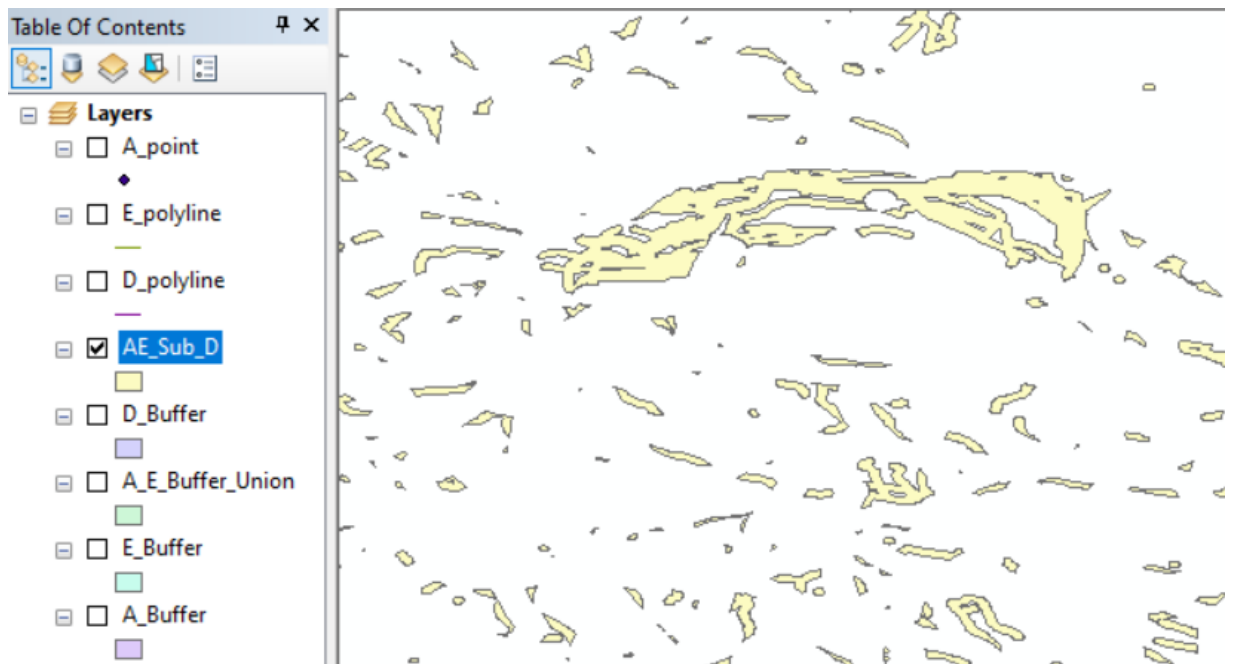
Для віднімання шарів використовується інструмент Erase:



На вхід подається два шари, між якими буде вирахувана різниця:

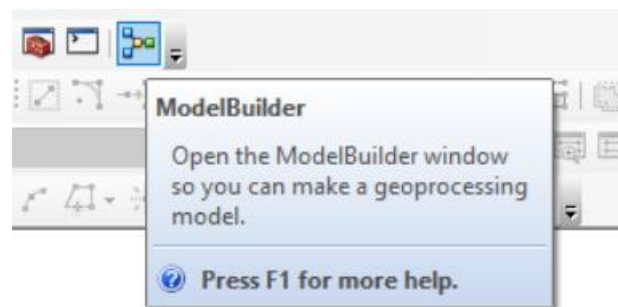


Результат:

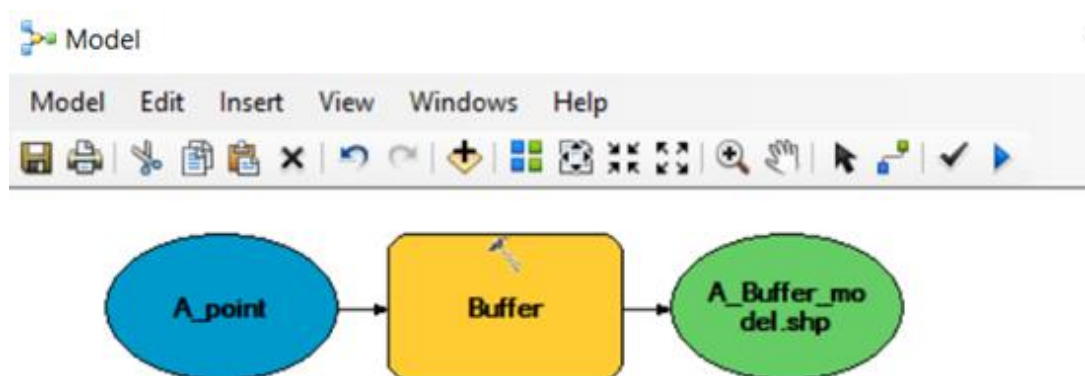


Частина друга. Створити модель по завданню в ModelBuilder і виконати її. У моделі вказувати наступні параметри: Назви вхідних шейп файлів, розміри радіусів буферних зон, назви – кінцевих шейп-файлів.

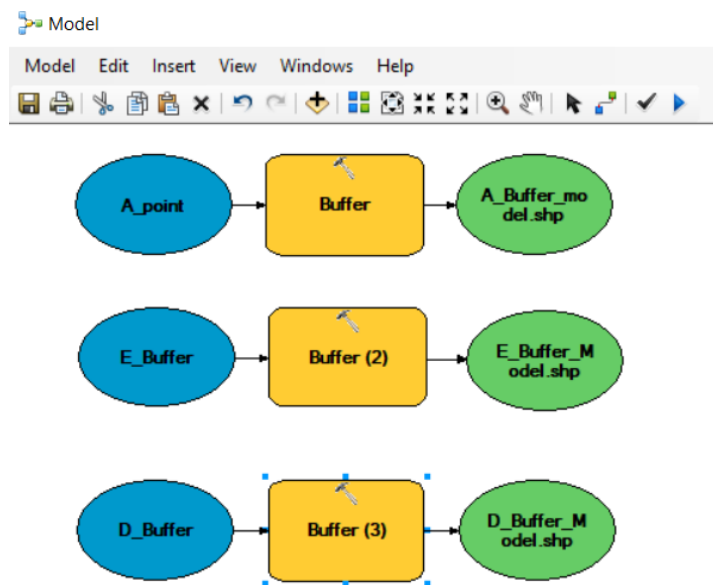
Вмикаємо ModelBuilder:



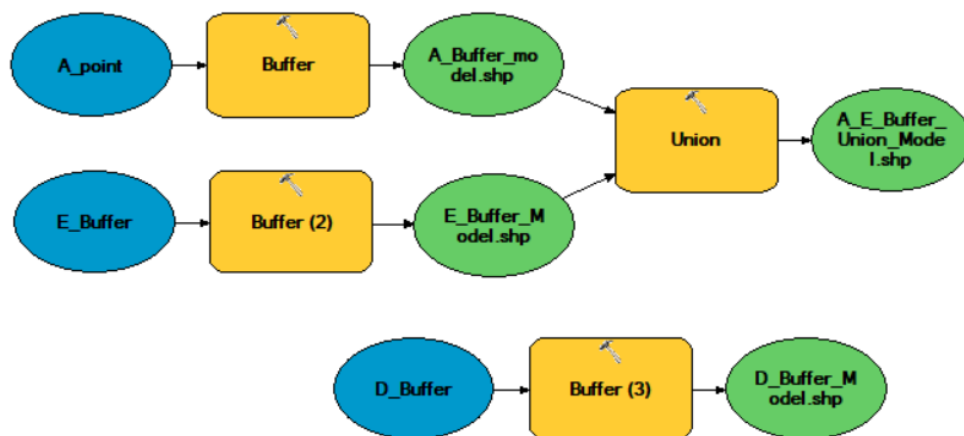
Перетягуємо “Buffer” у вікно ModelBuilder. Натискаємо на прямокутнику два рази, та вказуємо ті ж параметри що й у першій частині. Результат:



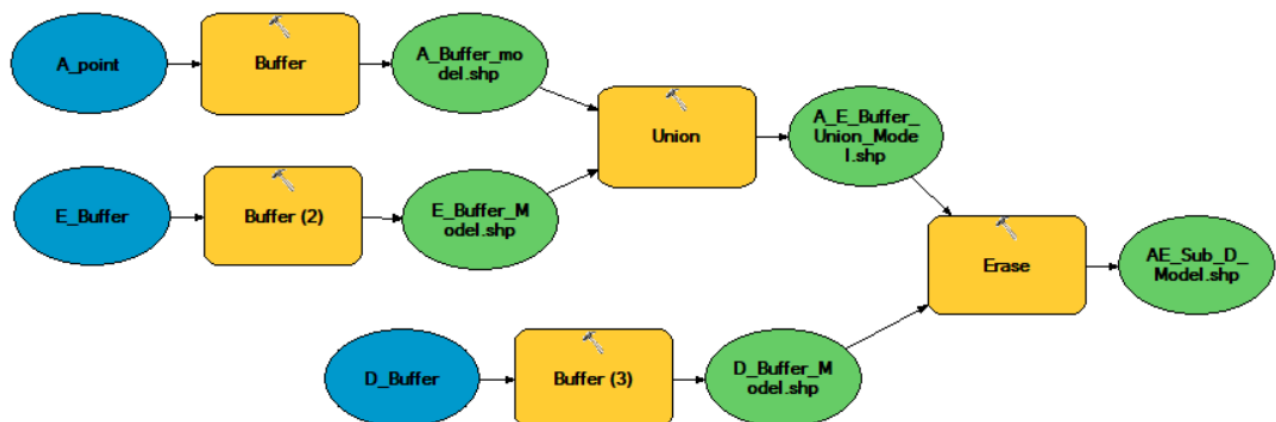
Автоматизуємо створення всіх трьох буферів:



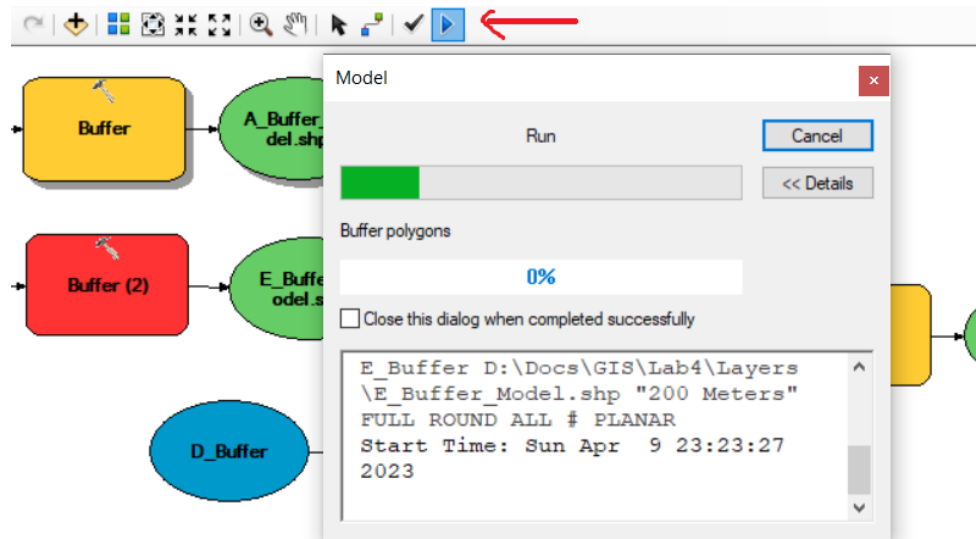
Об'єднання:



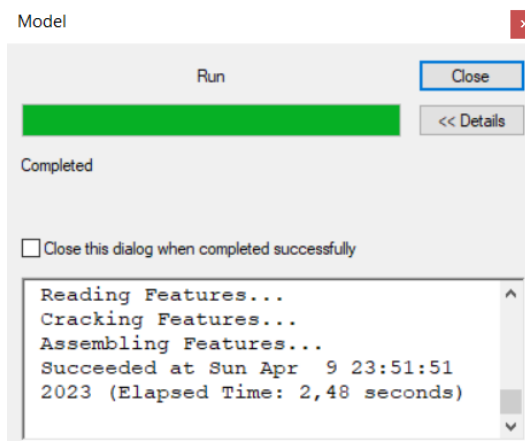
Результат:



Запуск моделі:



Модель виконана:

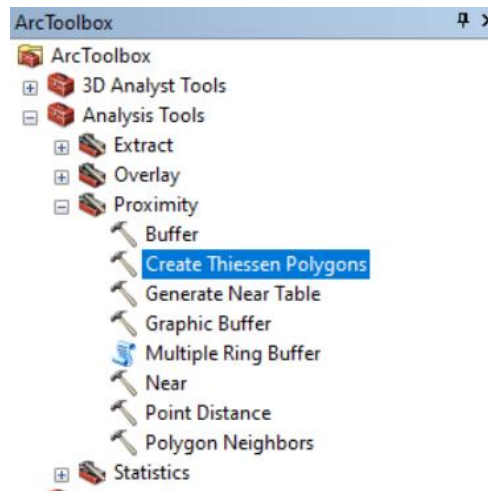


Так як за завданням не треба було встановлювати параметри для моделі, то після запуску отримуємо готовий шар:

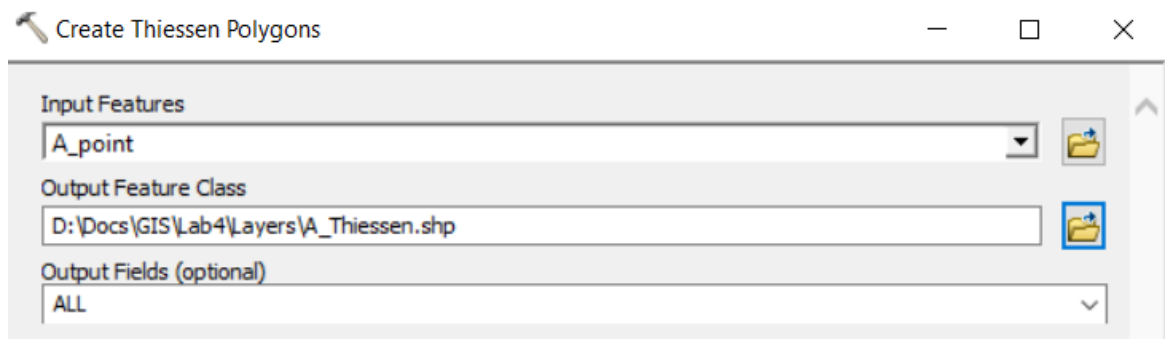


Частина третя. Для точкових файлів – створити полігони Тіссена, для поліліній – складні буферні зони.

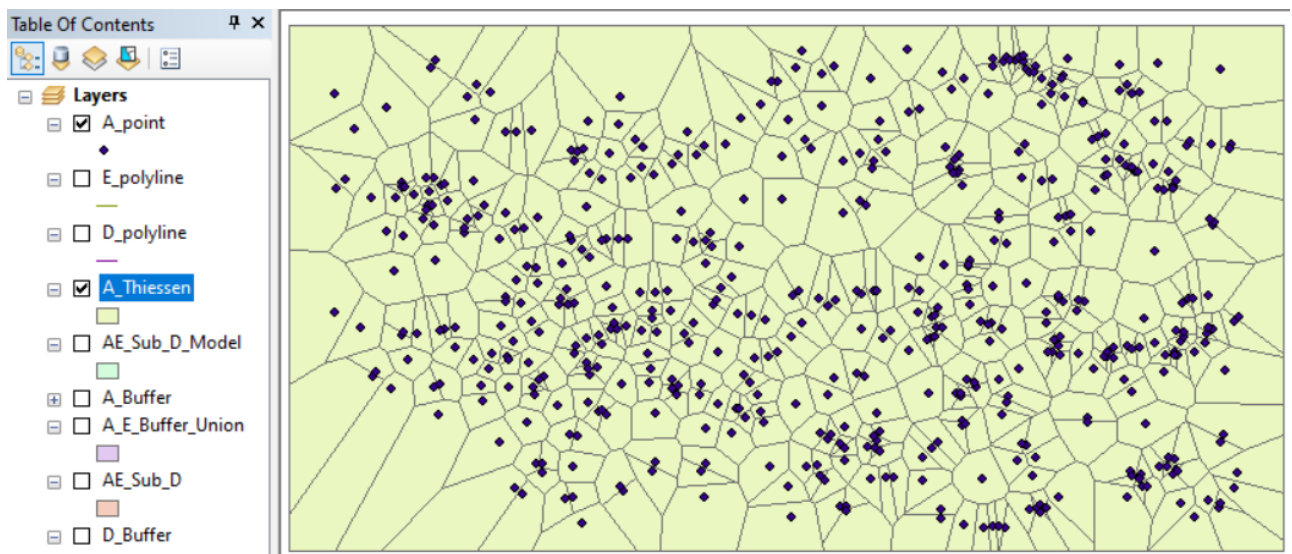
Інструмент для створення полігонів Тіссена:



Вікно створення:

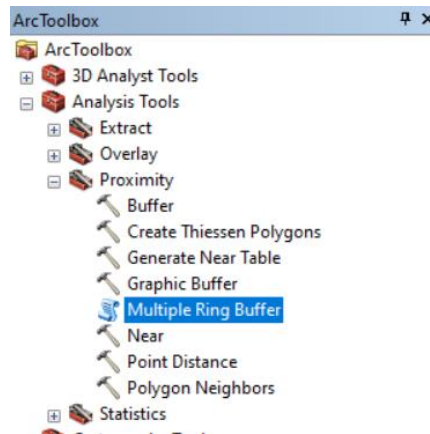


Результат:

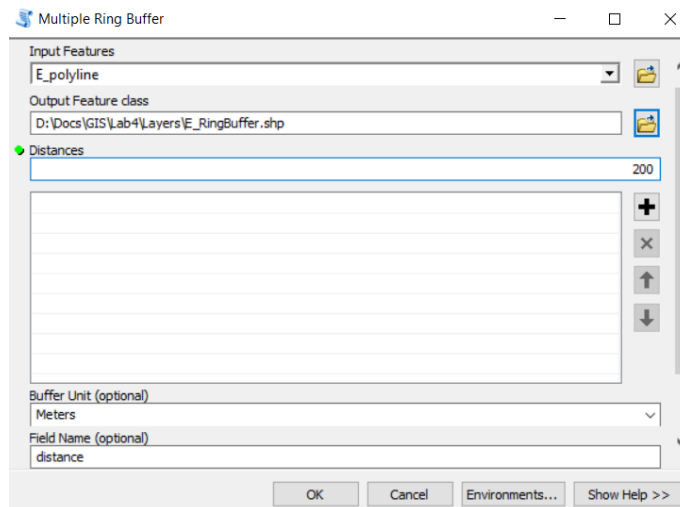


Полігон Тіссена (або діаграма Вороного) ділить площу на зони, для яких точка в середині неї є найближчою для будь-яких з цієї зони.

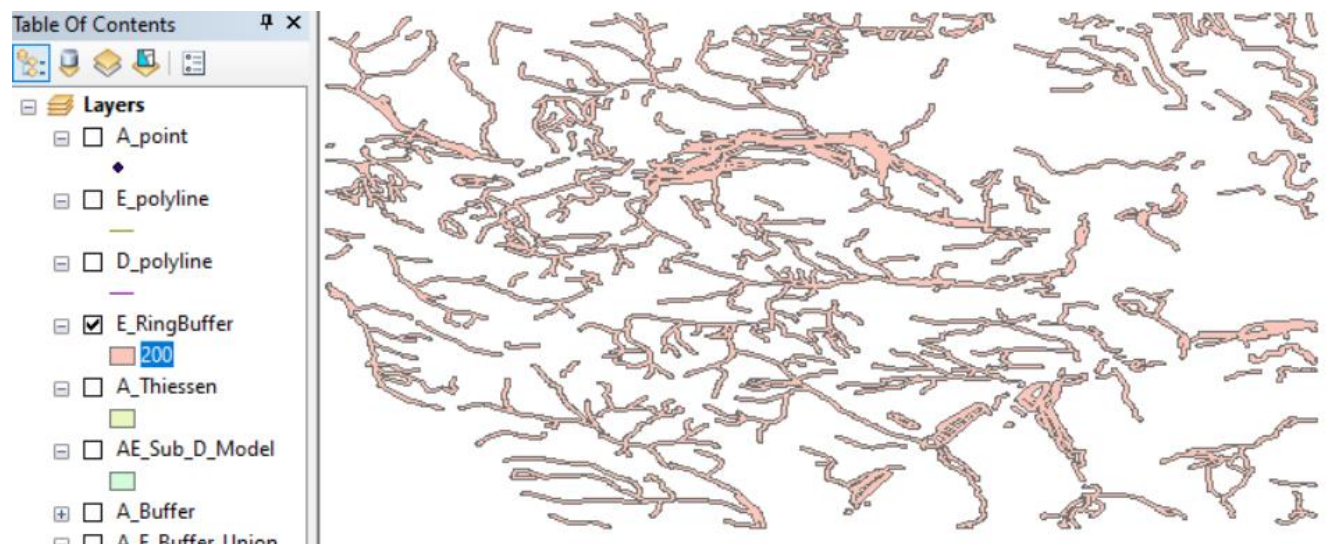
Інструмент для створення складної буферної зони:



Вікно створення:



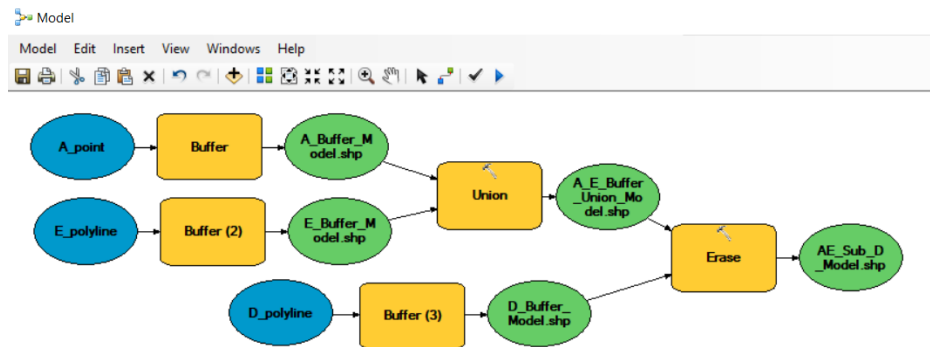
Результат:



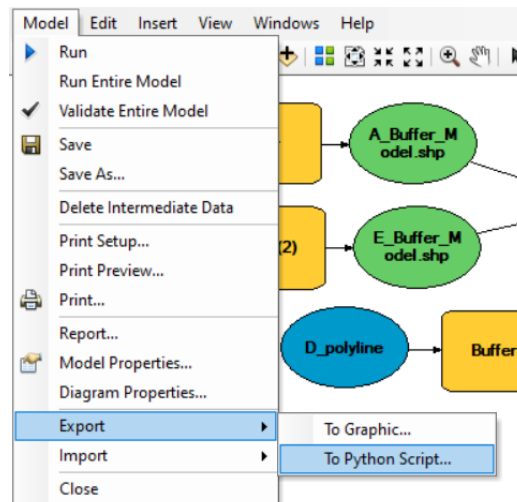
Складні буферні зони можна налаштовувати як діаграми, цим вони відрізняються від звичайних буферів.

Частина четверта. За допомогою автоматичного генерування в ModelBuilder створити скрипт по моделі на мові Python.

Повертаємось у ModelBuilder:



Експортуємо модель в скрипт:

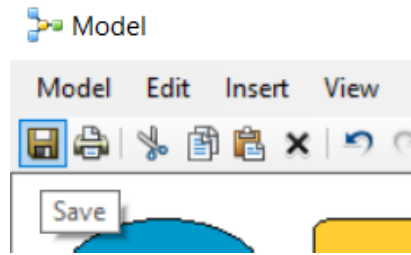


Результат:

```
ExportedModelScript.py
1  # -*- coding: utf-8 -*-
2  # -----
3  # ExportedModelScript.py
4  # Created on: 2023-04-10 01:02:29.00000
5  # (generated by ArcGIS/ModelBuilder)
6  # Description:
7  # -----
8
9  # Set the necessary product code
10 # import arcinfo
11
12
13 # Import arcpy module
14 import arcpy
15
16
17 # Local variables:
18 A_point_2_ = "A_point"
19 A_point = "A_point"
20 A_Buffer_Model_shp = "D:\\Docs\\GIS\\Lab4\\Layers\\A_Buffer_Model.shp"
21 E_polyline = "E_polyline"
22 E_Buffer_Model_shp = "D:\\Docs\\GIS\\Lab4\\Layers\\E_Buffer_Model.shp"
23 A_E_Buffer_Union_Model_shp = "D:\\Docs\\GIS\\Lab4\\Layers\\A_E_Buffer_Union_Model.shp"
24 D_polyline = "D_polyline"
25 D_Buffer_Model_shp = "D:\\Docs\\GIS\\Lab4\\Layers\\D_Buffer_Model.shp"
```

Частина п'ята. Додати отриману модель у ToolBox з назвою – “Kovalov”. Завантажити ToolBox з моделлю в набір інструментів геообробки ArcToolBox, щоб його можна було використовувати в ArcMap.

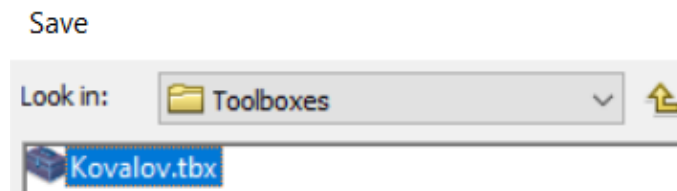
Зберігаємо модель:



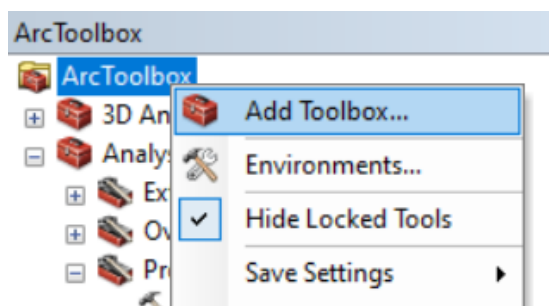
Модель можна зберегти лише в Toolbox, тому створюємо його:



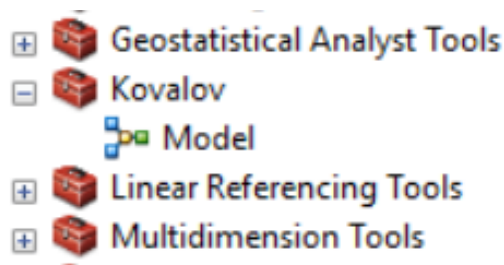
Називаємо так, як вказано в завданні, та зберігаємо в ньому модель:



Додаємо власний набір інструментів в ArcToolbox:



Результат:

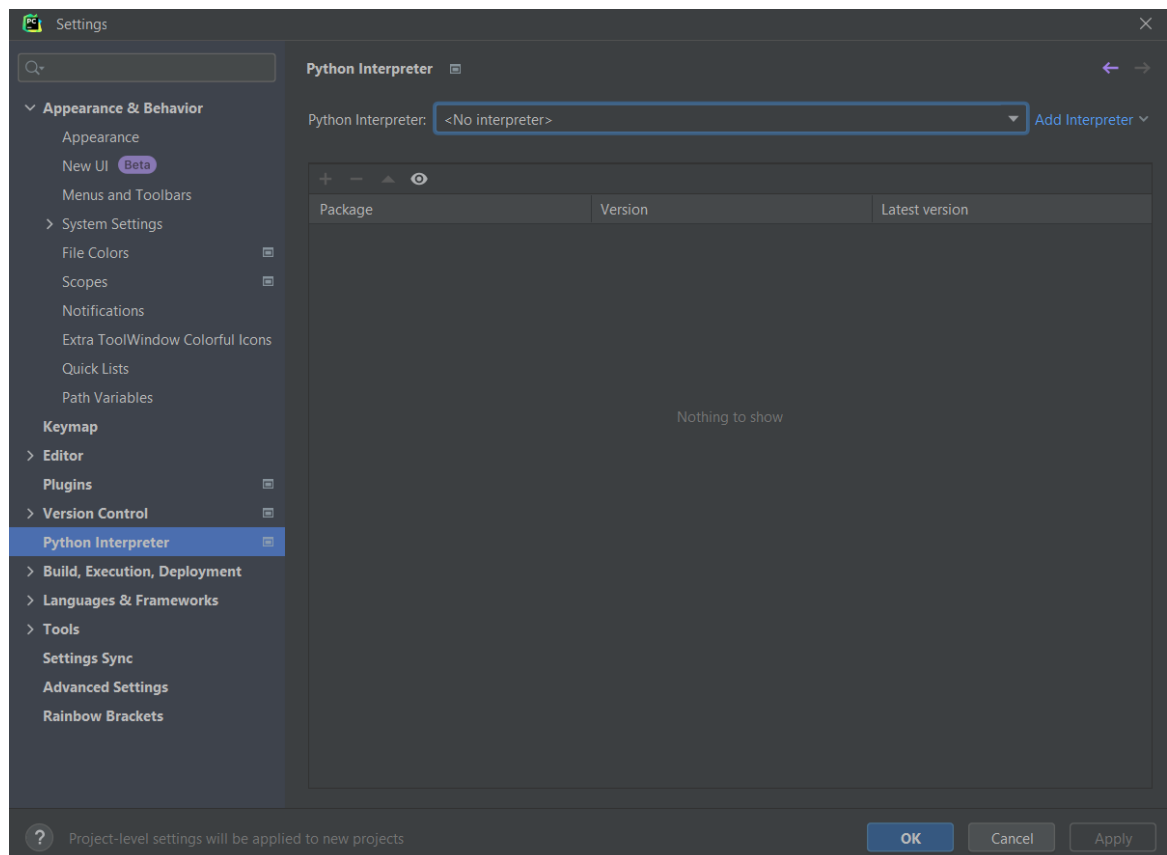


Частина шоста. Написати скрипт, додати його до власного набору інструментів, запустити окремо і в моделі.

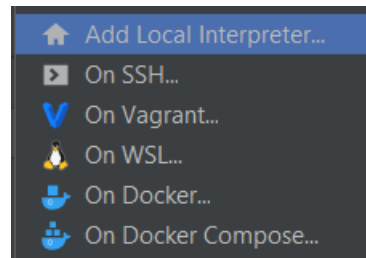
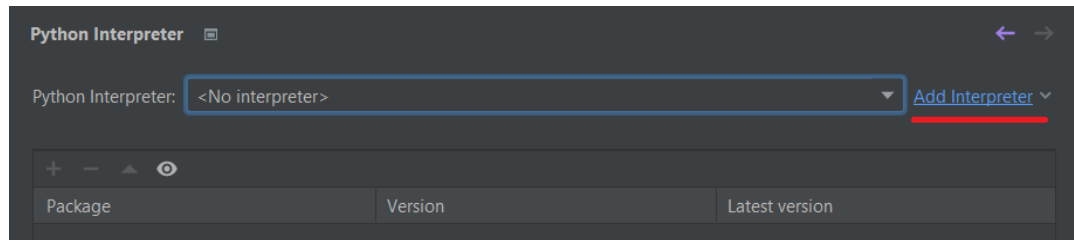
Будемо використовувати мову програмування Python. Скрипт можна писати у спеціальному вікні ArcMap, але можна і використовувати інтегроване середовище розробки – IDE. Для написання скрипту буде використовуватись IDE від JetBrains: PyCharm Proffesional 2023.1.



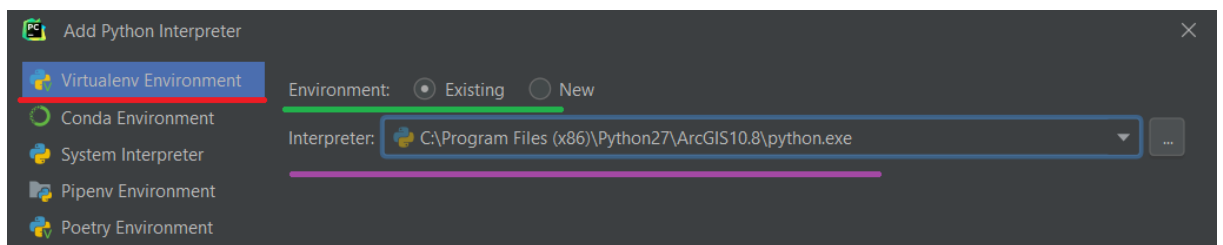
Для використання, потрібно додати інтерпретатор від ArcGis. На початковому вікні натискаємо Ctrl+Alt+S, щоб потрапити у вікно налаштувань. Потім, обираємо вкладку “Python Interpreter”:



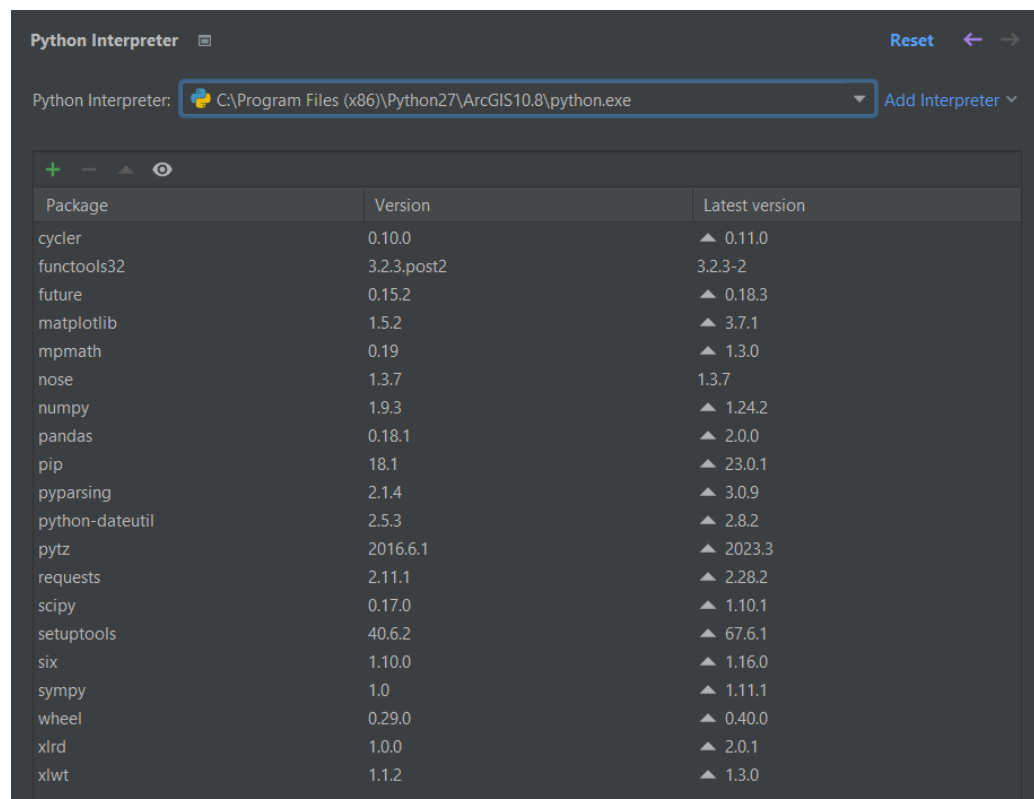
Додаємо інтерпретатор:



Інтерпретатор є частиною середовища Virtualenv. Також, він вже існує, тому ставимо Environment: Existing. Шлях вказуємо такий же, як на скріншоті.



Результат:



Встановлюємо virtualenv на наш інтерпретатор Python 2.7:

```
PS D:\Work\Python\ArcTest> & 'C:\Program Files (x86)\Python27\ArcGIS10.8\python.exe' -m pip install virtualenv --user
Collecting virtualenv
  Using cached https://files.pythonhosted.org/packages/6f/43/df7c7b1b7a5ac4e41fac24c3682c1cc32f2c1d683d308bba2500338d1e3e/vir
Collecting importlib-metadata>=0.12; python_version < "3.8" (from virtualenv)
  Using cached https://files.pythonhosted.org/packages/cf/b4/877779cd7b5a15536ecbe0655cfb35a0de0ede6d888151fd7356d278c47d/imp
Collecting platformdirs<3,>=2 (from virtualenv)
```

Пишемо скрипт:

```
main.py x
1 import arcpy
2 import os
3
4 # Workspace settings
5 arcpy.env.workspace = os.getcwd()
6 arcpy.env.overwriteOutput = True
7
8 # Greetings!
9 print("Hello! This script makes buffer from your shape file.")
10 print("Author: Alex Kovalov, TP-12\n")
11
12 # Angs input
13 Input_File = str(input("Input SHP filename: "))
14 Distance = input("Input distance: ")
15 Output_File = input("Result SHP filename: ")
16
17 # Arguments by default
18 Line_Side = "FULL"
19 Line_End_Type = "ROUND"
20 Dissolve_Type = "ALL"
21 Dissolve_Option = ""
22 Method = "PLANAR"
23
24 # Process: Buffer
25 arcpy.Buffer_analysis(Input_File, Output_File, Distance, Line_Side, Line_End_Type, Dissolve_Type, Dissolve_Option,
26                        Method)
```

Спробуємо запустити:

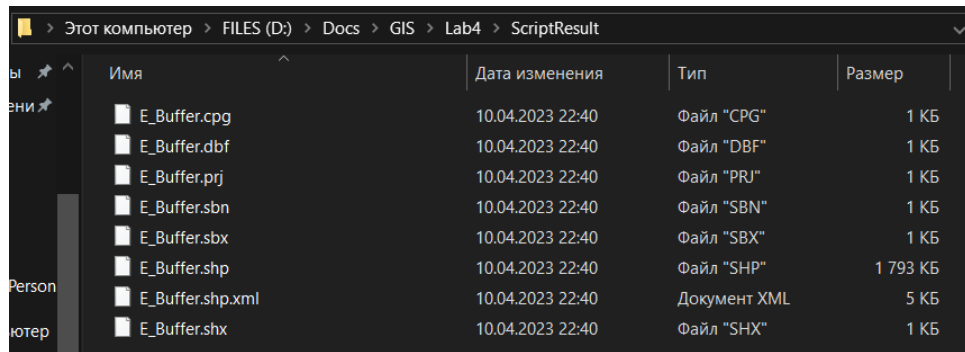
```
"C:\Program Files (x86)\Python27\ArcGIS10.8\python.exe" D:\Work\Python\ArcTest\main.py
Hello! This script makes buffer from your shape file.
Author: Alex Kovalov, TP-12

Input SHP filename: "D:\Docs\GIS\Lab4\SHp\E_polyline.shp"
Input distance: "500 Meters"
Result SHP filename: "D:\Docs\GIS\Lab4\ScriptResult\E_Buffer.shp"

Process finished with exit code 0
```

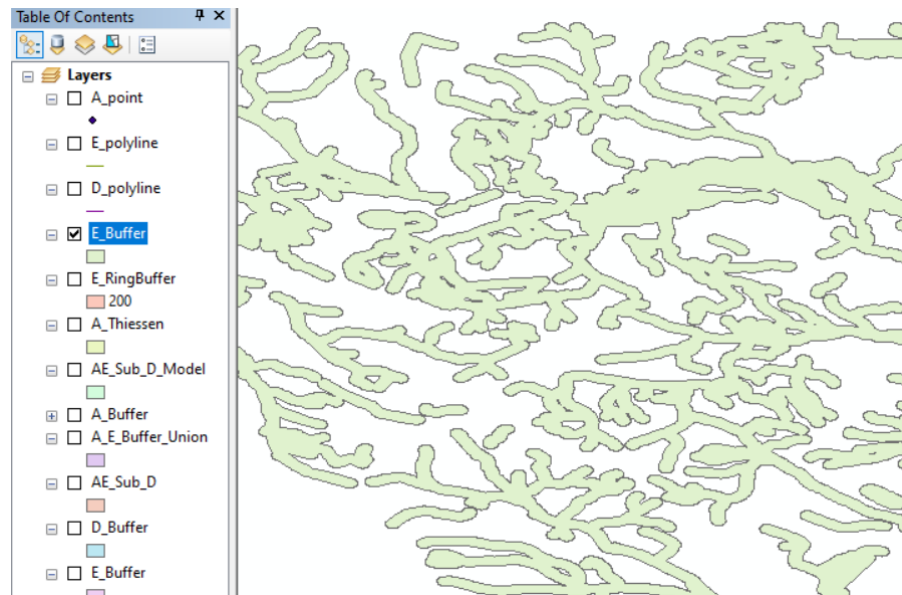
Скрипт повинен сформувати буфер файл зі вказаного шейп файлу. Також, в ньому передбачена можливість вводити дистанцію та назву результативного файлу.

Результат:



Имя	Дата изменения	Тип	Размер
E_Buffer.cpg	10.04.2023 22:40	Файл "CPG"	1 КБ
E_Buffer.dbf	10.04.2023 22:40	Файл "DBF"	1 КБ
E_Buffer.prj	10.04.2023 22:40	Файл "PRJ"	1 КБ
E_Buffer.sbn	10.04.2023 22:40	Файл "SBN"	1 КБ
E_Buffer.sbx	10.04.2023 22:40	Файл "SBX"	1 КБ
E_Buffer.shp	10.04.2023 22:40	Файл "SHP"	1 793 КБ
E_Buffer.shp.xml	10.04.2023 22:40	Документ XML	5 КБ
E_Buffer.shx	10.04.2023 22:40	Файл "SHX"	1 КБ

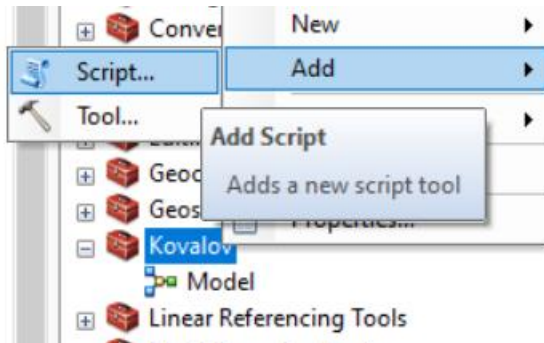
Переглянемо в ArcMap:



Так як в ArcMap не передбачений консольний інтерфейс, напишемо ще одну версію скрипту. В ній немає взаємодії з консоллю, імпортована бібліотека sys, та всі аргументи надходять як зовнішні аргументи.

```
main.py
1 import arcpy
2 import os
3 import sys
4
5 # Workspace settings
6 arcpy.env.workspace = os.getcwd()
7 arcpy.env.overwriteOutput = True
8
9 # Args input
10 Input_File = sys.argv[1]
11 Distance = sys.argv[2]
12 Output_File = sys.argv[3]
13
14 # Arguments by default
15 Line_Side = "FULL"
16 Line_End_Type = "ROUND"
17 Dissolve_Type = "ALL"
18 Dissolve_Option = ""
19 Method = "PLANAR"
20
21 # Process: Buffer
22 arcpy.Buffer_analysis(Input_File, Output_File, Distance, Line_Side, Line_End_Type, Dissolve_Type, Dissolve_Option,
23                       Method)
```

Для того, щоб додати скрипт до набору інструментів, ПКМ -> Add -> Script... :




Вказуємо назву та мітку скрипту. Також, можна додати опис, та увімкнути фонове виконання за замовчуванням.

Add Script ×

Name:

Label:

Description:


Stylesheet:
 

☐ Store relative path names (instead of absolute paths)

☒ Always run in foreground

Завантажуємо скрипт:

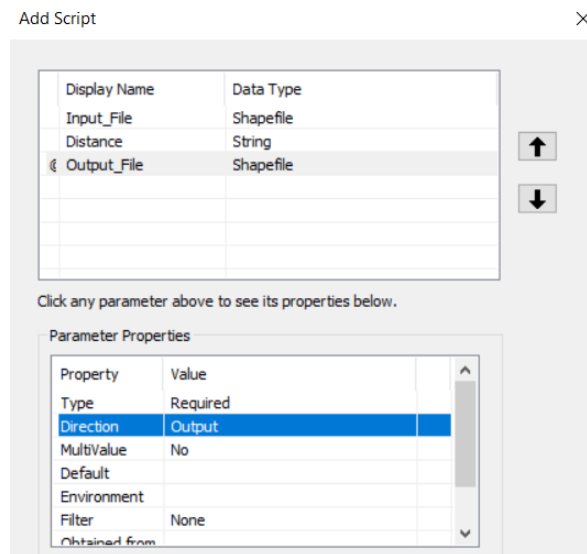
Add Script

Script File:
 

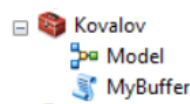
☐ Show command window when executing script

☒ Run Python script in process

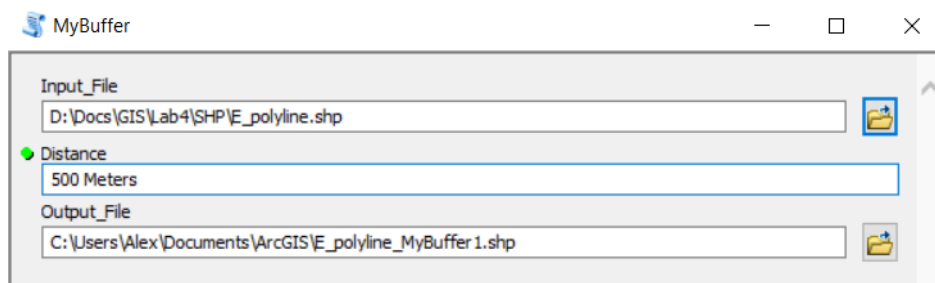
Додаємо параметри, які будуть передаватись скрипту. Input_File приймає тип Shapefile, Distance – String. Output_File, на відміну від двох попередніх, має Direction Output, а не Input. Також приймає Shapefile:



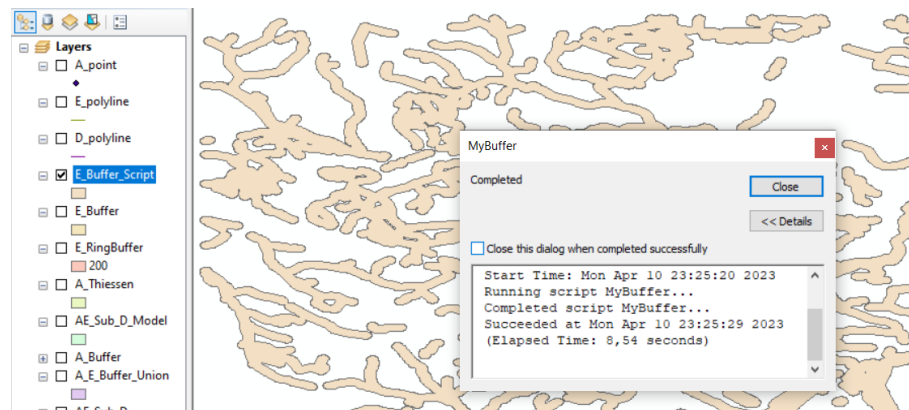
Скрипт в Toolbox:



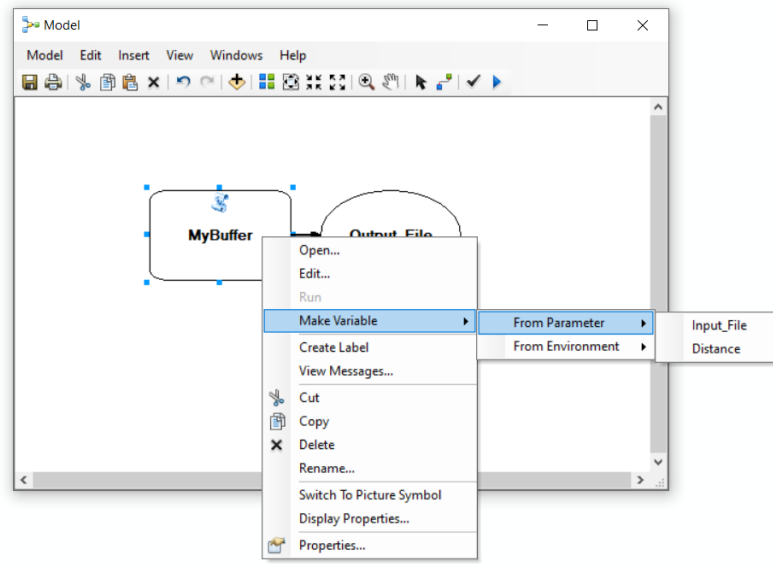
Вікно дозволяє обрати шейп-файл, радіус, та відразу генерує назву для файлу з результатом (можна зберегти під своєю):



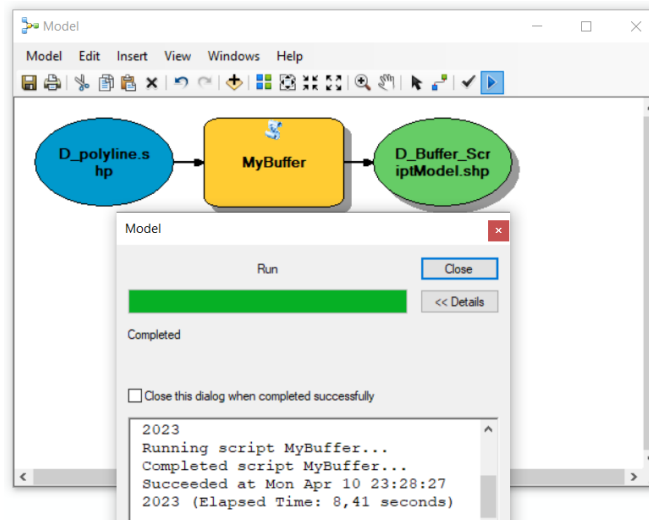
Результат:



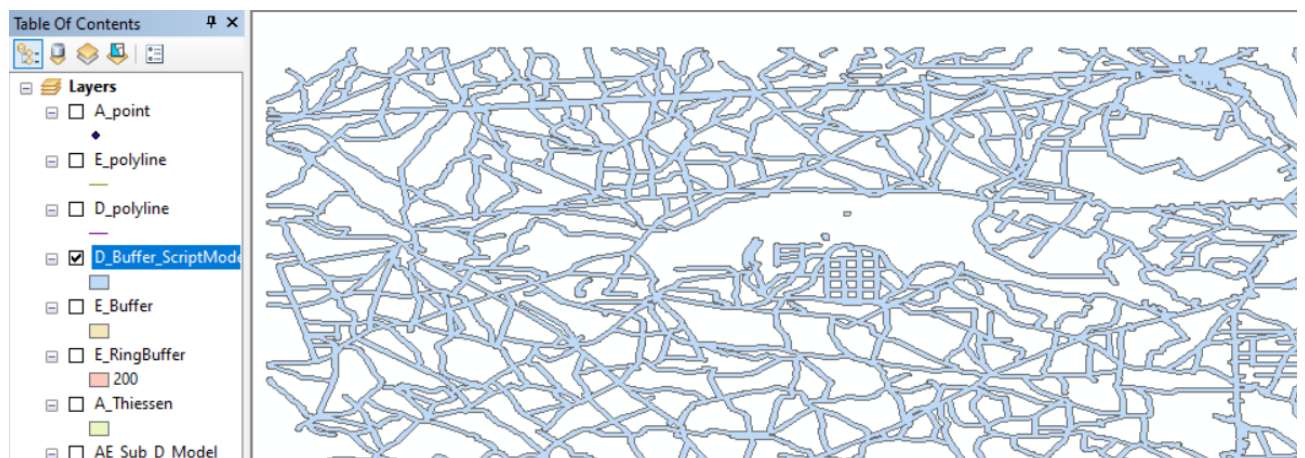
Спробуємо запустити скрипт в моделі. Все працює аналогічно звичайному буферу, лише спрощене введення аргументів:



Запуск:



Результат:



Висновок: за результатами виконання цієї лабораторної роботи були отримані базові навички роботи з інструментами геообробки ArcToolBox, з інструментом побудови моделей ModelBuilder. Була проведена робота з буфером, та такими операціями як об'єднання та різниця полігонів, створена діаграма Тісенна (Вороного) та складна буферна зона. За допомогою експорту був автоматично створений скрипт мовою програмування Python з побудованої моделі. Був створений власний набір інструментів, куди потрапила створена модель. Також, туди потрапив власний самописний скрипт, для створення буферів. Для написання було налаштоване інтегроване середовище розробки, та під'єднаний окремий інтерпретатор від ArcGis, з попередньо налаштованими пакетами та мовою програмування Python 2.7. Скрипт був відлагоджений та протестований вручну і з використанням ModelBuilder.