

Projekt zaliczeniowy – Podstawy **programowania aplikacji GIS**

Przygotowali:

Bartek Koniec

Konrad Florek

Szymon Krawczyński

1. Założenie projektu

Stworzenie działającego narzędzia w Pythonie, które będzie dostępne do zaczytania w ArcGis, za pomocą ArcToolbox, opierające się na danych archiwalnych z IMGW (meteo oraz hydro).

2. Etapy powstawania

- 1) Na początku pobraliśmy odpowiednio interesujące nas dane archiwalne z IMGW(meteo i hydro),
- 2) Wybraliśmy dane miesięczne z lat 2012-2020 w obu przypadkach,
- 3) Potrzebne było odpowiednie odkodowanie danych w plikach(hydro) CSV co zrobiliśmy jak na poniższym obrazku,

	A	B	C	D	E	F
1	CHAŁUPKI	Odra (1)	2012	1	102	10.000
2	CHAŁUPKI	Odra (1)	2012	1	109	12.200
3	CHAŁUPKI	Odra (1)	2012	1	117	14.900
4	CHAŁUPKI	Odra (1)	2012	2	101	9.700
5	CHAŁUPKI	Odra (1)	2012	2	111	12.700
6	CHAŁUPKI	Odra (1)	2012	2	118	15.200
7	CHAŁUPKI	Odra (1)	2012	3	111	12.800
8	CHAŁUPKI	Odra (1)	2012	3	149	33.800
9	CHAŁUPKI	Odra (1)	2012	3	190	70.700
10	CHAŁUPKI	Odra (1)	2012	4	117	18.800
11	CHAŁUPKI	Odra (1)	2012	4	145	35.900
12	CHAŁUPKI	Odra (1)	2012	4	290	164.000
13	CHAŁUPKI	Odra (1)	2012	5	158	45.800
14	CHAŁUPKI	Odra (1)	2012	5	187	70.700
15	CHAŁUPKI	Odra (1)	2012	5	274	149.000
16	CHAŁUPKI	Odra (1)	2012	6	129	25.900
17	CHAŁUPKI	Odra (1)	2012	6	156	44.500
18	CHAŁUPKI	Odra (1)	2012	6	216	96.800
19	CHAŁUPKI	Odra (1)	2012	7	100	13.500
20	CHAŁUPKI	Odra (1)	2012	7	126	25.300
21	CHAŁUPKI	Odra (1)	2012	7	170	55.800
22	CHAŁUPKI	Odra (1)	2012	8	101	12.300
23	CHAŁUPKI	Odra (1)	2012	8	135	32.600
24	CHAŁUPKI	Odra (1)	2012	8	282	156.000
25	CHAŁUPKI	Odra (1)	2012	9	89	8.240
26	CHAŁUPKI	Odra (1)	2012	9	102	12.900
27	CHAŁUPKI	Odra (1)	2012	9	134	28.500
28	CHAŁUPKI	Odra (1)	2012	10	86	7.530
29	CHAŁUPKI	Odra (1)	2012	10	97	11.300
<div> <div> <div></div> <div>mies_2012</div> <div></div> </div> <div></div> </div>						

4) Odkodowane kolumny od lewej to – stacja, rzeka, rok hydrologiczny, miesiąc hydrologiczny, stan wody, przepływ wody,

5) Tak samo postąpiliśmy w danych meteo,

6) Po zorientowaniu się w odkodowanych danych przeszliśmy do tworzenia kodów,

```

import sqlite3 as sl, csv

con = sl.connect('meteo.dbf')

cursor = con.cursor()

cursor.execute("""
    CREATE TABLE meteo(
        kod_stacji INTEGER,
        stacja TEXT,
        rok INTEGER,
        miesiac INTEGER,
        suma_opadow FLOAT,
        dni_z_opad_snieg INTEGER,
        opad_maksymalny FLOAT
    )
""")

```

- 7) Widzimy import odpowiednich bibliotek z sqlite3, a następnie utworzenie połączenia z bazą przechowywaną na dysku,
- 8) Potem następuje utworzenie obiektu kursora oraz stworzenie opisanych tablic różnego typu,
- 9) Dodana została też funkcja zmieniająca polskie znaki, ponieważ ArcMapa znajduje w nich błędy,

```

# funkcja do zamiany znaków
def removeAccents(input_text):
    strange = 'ŮôŮBaě"ĀñěĐđóíñ3BŇl'žěñěý'ĀŌñiĚüürŽŎŇgŮλBŵé"ĒŤŮi"Ŧqž'EāŵŮkŮχ"ĀŮ
    ascii_replacements = 'UoyBdeAieDaoiizVNiIzeneyAŌiieyyrZŌNgulVoeETUŷ'
    translator = str.maketrans(strange, ascii_replacements)

    return input_text.translate(translator)

```

- 10) Na koniec pozostało zaczytanie potrzebnych plików CSV oraz ich odpowiednie odkodowanie (tym razem za pomocą kodu) i sformatowanie,

```
pliki = ['o_m_2012.csv', 'o_m_2013.csv', 'o_m_2014.csv', 'o_m_2015.csv', 'o_m_2016.csv',  
        'o_m_2017.csv', 'o_m_2018.csv', 'o_m_2019.csv', 'o_m_2020.csv']  
  
for x in pliki:  
    file = open(x, "r")  
    csv_reader = csv.reader(file, delimiter=';')  
    for row in csv_reader:  
        kod_stacji = int(row[0])  
        stacja = removeAccents(row[1])  
        rok = int(row[2])  
        miesiac = int(row[3])  
        suma_opadow = float(row[4].replace(',', '.'))  
        dni_z_opad_snieg = int(row[5])  
        opad_maksymalny = float(row[6].replace(',', '.'))  
        cursor.execute('INSERT INTO meteo(kod_stacji, stacja, rok, miesiac, suma_opadow,  
                                dni_z_opad_snieg, opad_maksymalny)  
                                VALUES (?, ?, ?, ?, ?, ?, ?)',  
                        (kod_stacji, stacja, rok, miesiac, suma_opadow,  
                          dni_z_opad_snieg, opad_maksymalny))  
  
con.commit()  
con.close()
```

- 11) To było opracowanie danych meteo, jeśli chodzi o dane hydro to są one analogiczne z kilkoma różnicami przy opisaniu tablic i kolejnych wartości oraz przy odczytywaniu i odkodowywaniu danych,

```

import csv
import sqlite3 as sl

con = sl.connect('hydro.dbf')

cursor = con.cursor()

cursor.execute("""
    CREATE TABLE hydro(
        kod_stacji INTEGER,
        stacja TEXT,
        rzeka TEXT,
        rok_hydro INTEGER,
        miesiac_hydro INTEGER,
        stan_wody INTEGER,
        przeplyw FLOAT
    )
""")

```

```

pliki = ['mies_2012.csv', 'mies_2013.csv',
        'mies_2014.csv', 'mies_2015.csv', 'mies_2016.csv', 'mies_2017.csv',
        'mies_2018.csv', 'mies_2019.csv', 'mies_2020.csv']

for x in pliki:
    file = open(x, "r")
    csv_reader = csv.reader(file, delimiter=';')
    for row in csv_reader:
        kod_stacji = int(row[0])
        stacja = removeAccents(row[1])
        rzeka = removeAccents(row[2])
        rok_hydro = int(row[3])
        miesiac_hydro = int(row[4])
        stan_wody = int(row[5])
        przeplyw = float(row[6])
        cursor.execute('INSERT INTO hydro(kod_stacji, stacja, rzeka, rok_hydro,
                                         miesiac_hydro, stan_wody, przeplyw)
                        VALUES (?, ?, ?, ?, ?, ?, ?)'
                        (kod_stacji, stacja, rzeka, rok_hydro,
                         miesiac_hydro, stan_wody, przeplyw))

con.commit()
con.close()

```

- 12) Przechodzimy teraz do kluczowego kodu, który będzie zaczytywany do stworzenia naszego narzędzia docelowego,

```
import arcpy
import sqlite3 as sl

try:
    hydro_meteo = arcpy.GetParameterAsText(0)
    baza = arcpy.GetParameterAsText(1)
    data_rok = arcpy.GetParameter(2)
    plik_wy = arcpy.GetParameterAsText(3)
    if ".sqlite" not in plik_wy:
        plik_wy += ".sqlite"

    data_rok = int(data_rok)

    if hydro_meteo == "hydro":
        conn = sl.connect(baza)
        cursor = conn.cursor()
        cursor.execute("select * from hydro")

        con = sl.connect(plik_wy)
        cursor2 = con.cursor()
        cursor2.execute("""
            CREATE TABLE hydro(
                kod_stacji INTEGER,
                stacja TEXT,
                rzeka TEXT,
                rok_hydro INTEGER,
                miesiac_hydro INTEGER,
                stan_wody INTEGER,
                przeplyw FLOAT
            )
        """)
```

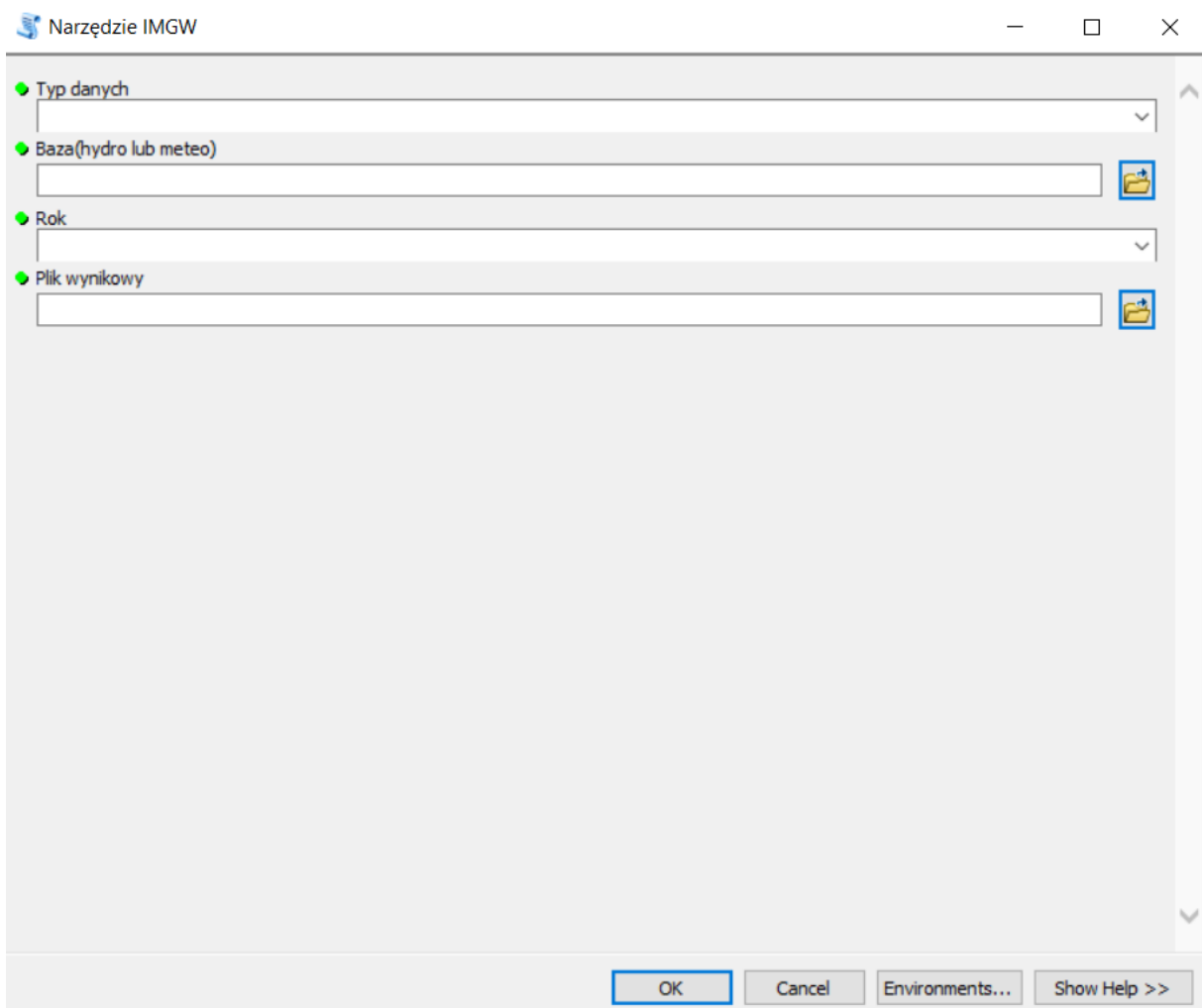
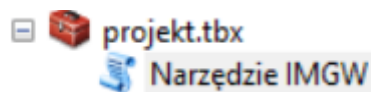
- 13) Korzystamy z importu pakietu arcpy, a następnie tworzymy zmienne konfigurujące nasze narzędzie, kolejne kroki to powtórzenie operacji z poprzednich plików,

- 14) Program kończymy wystąpieniem ewentualnego błędu lub komunikatu o pomyślnym procesie,

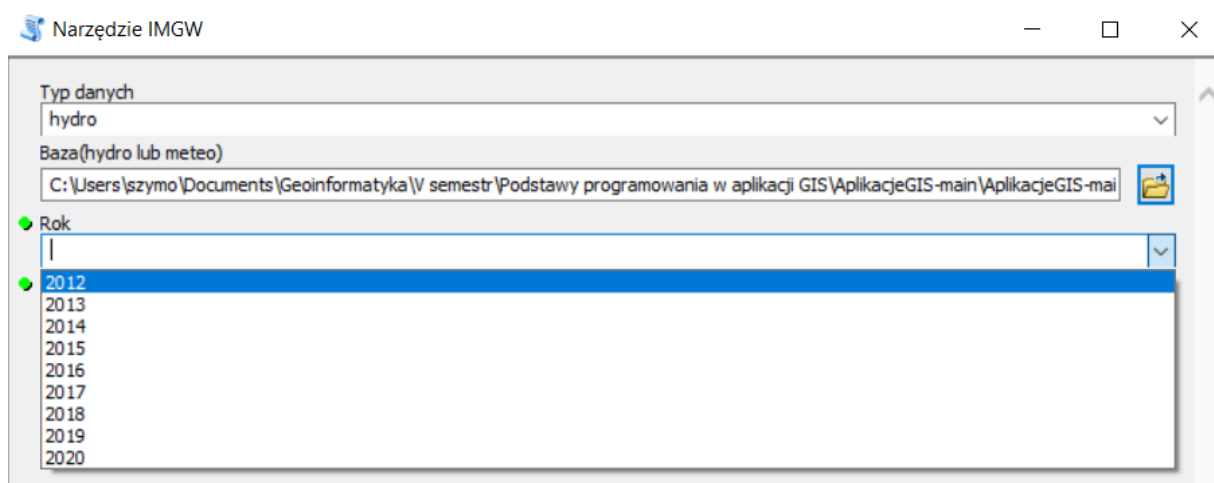
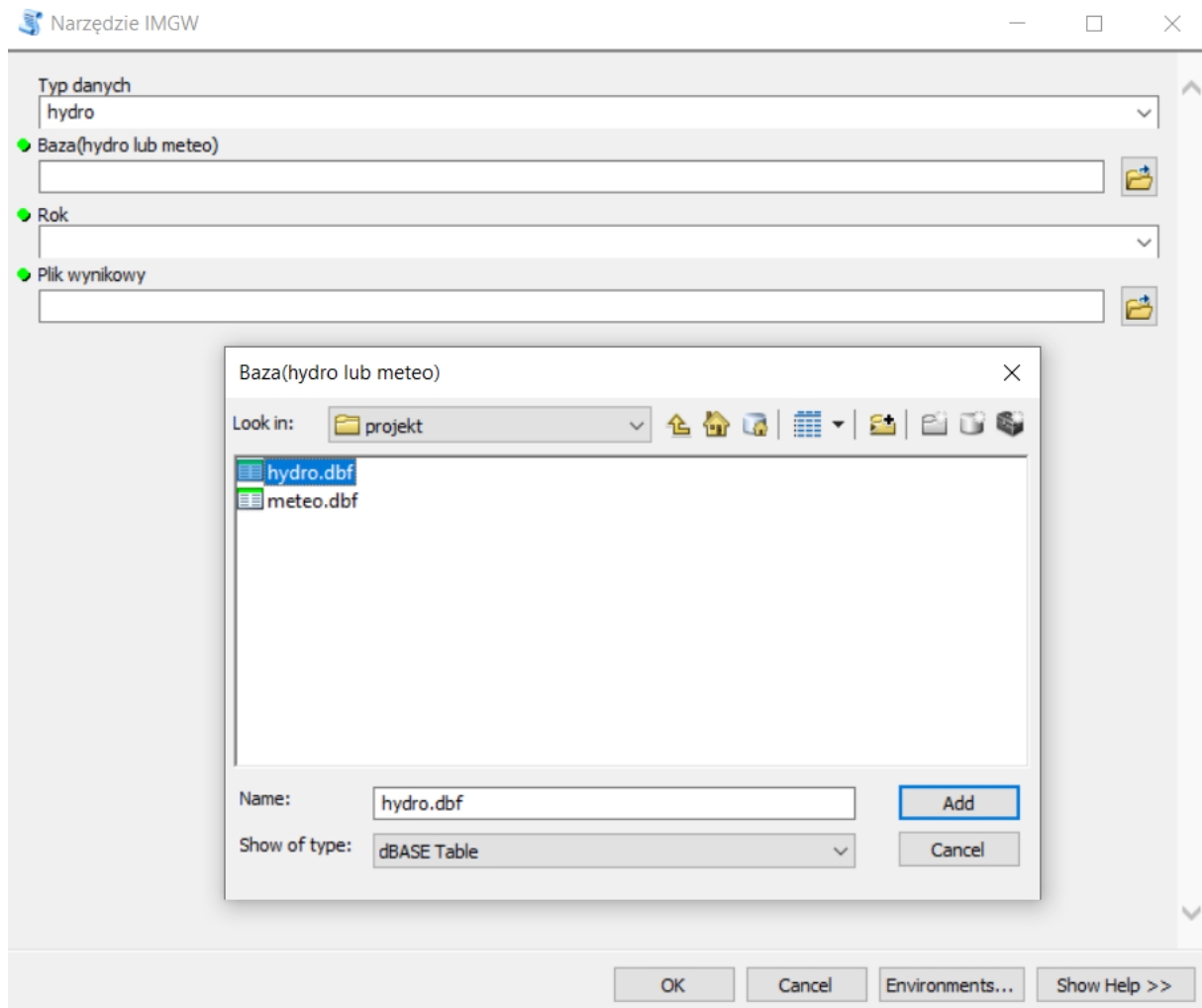
```
except:
    arcpy.AddMessage("Wystąpił nieoczekiwany błąd. Ponów próbę, sprawdzając przy tym poprawność danych")

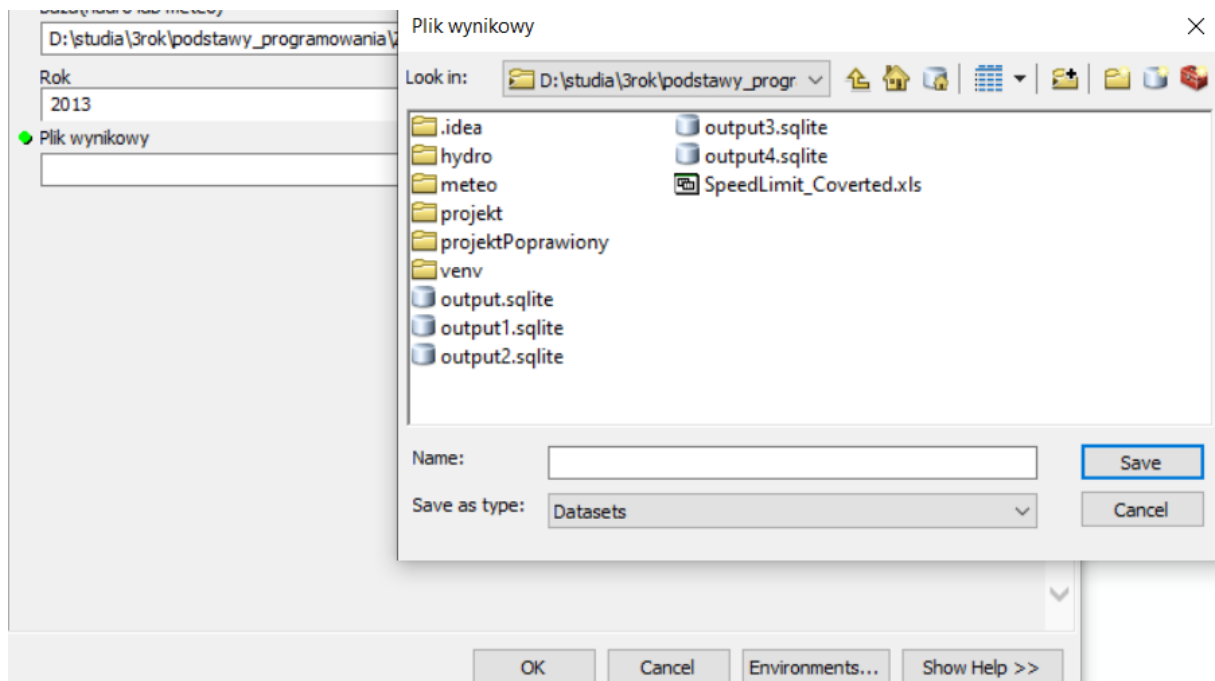
finally:
    arcpy.AddMessage("Proces przebiegł pomyślnie")
```

- 15) Mając działający skrypt możemy go załadować do Arc Mapy w ArcToolboxie,

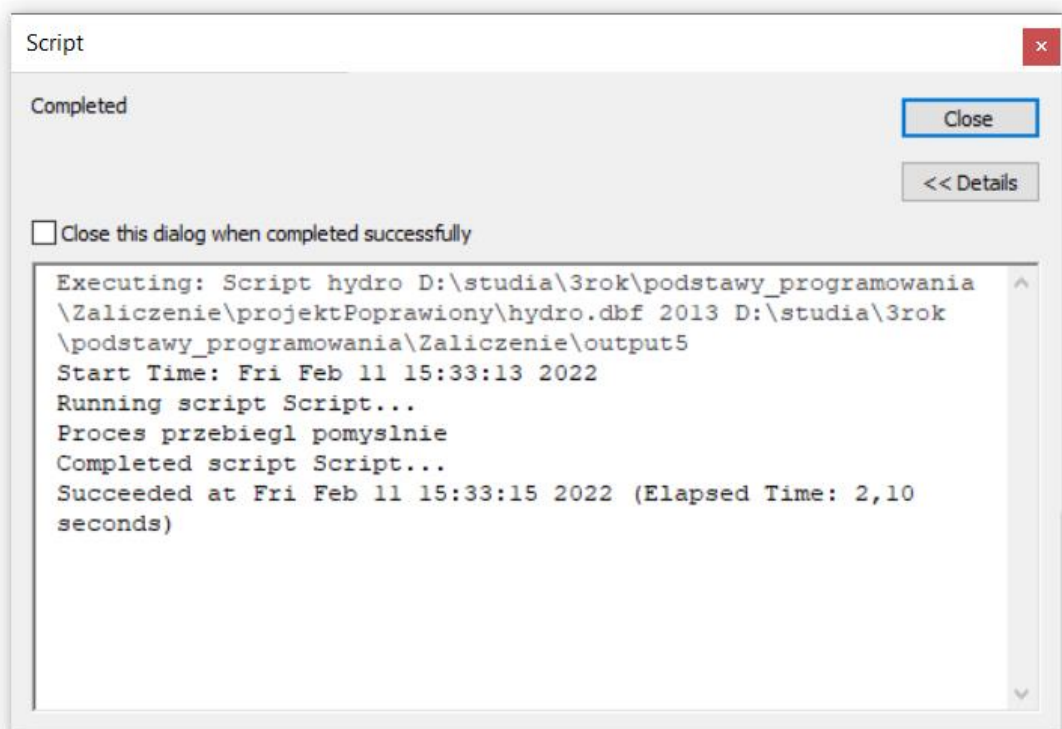
The image is a screenshot of a software dialog box titled 'Narzędzie IMGW'. It contains four input fields, each preceded by a green diamond icon. The first field is labeled 'Typ danych' and has a dropdown arrow. The second field is labeled 'Baza(hydro lub meteo)' and has a folder icon to its right. The third field is labeled 'Rok' and has a dropdown arrow. The fourth field is labeled 'Plik wynikowy' and has a folder icon to its right. At the bottom of the dialog, there are four buttons: 'OK', 'Cancel', 'Environments...', and 'Show Help >>'. The dialog box has standard Windows window controls (minimize, maximize, close) in the top right corner.

- 16) Narzędzie pozwala nam więc na wybór typu danych(hydro/meteo),
wybranie bazy do danych z dysku, roku z którego pochodzą pomiary
oraz zapis do pliku wynikowego na dysku,

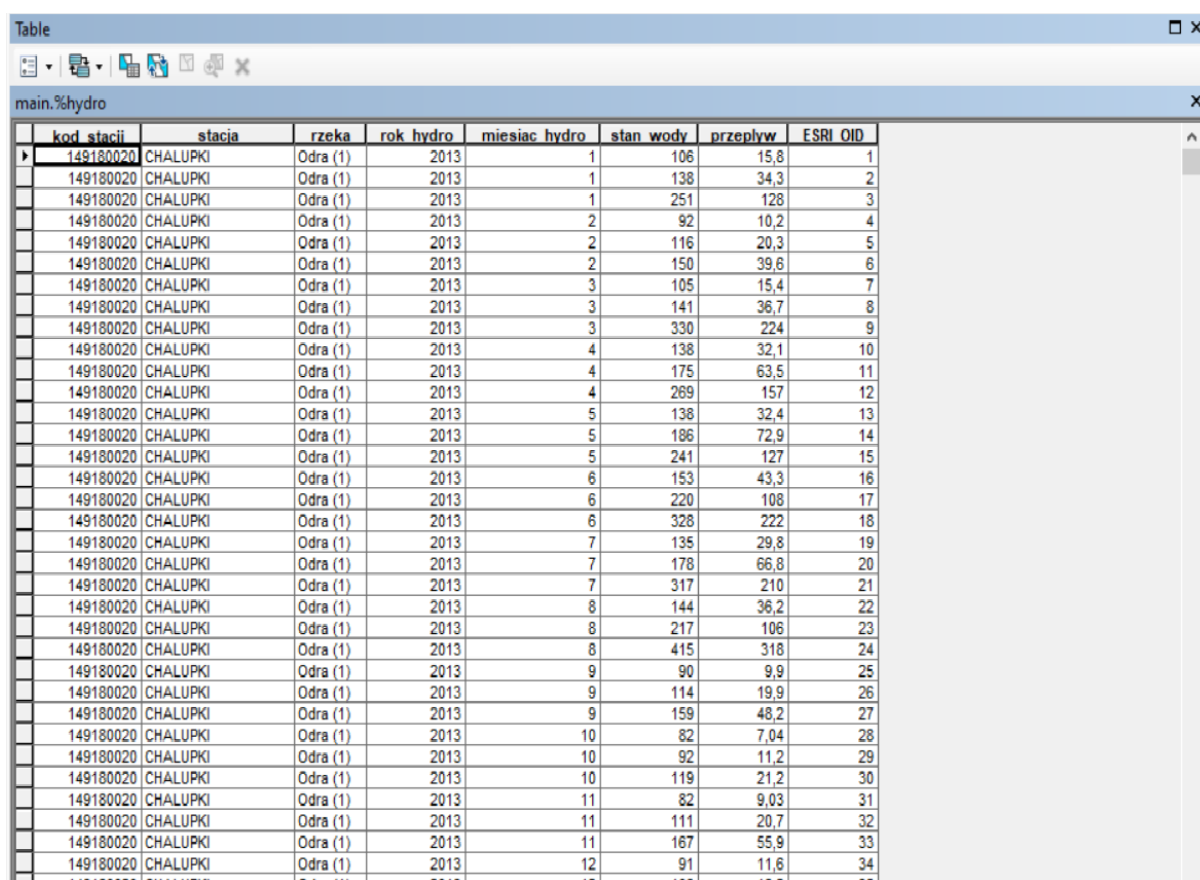




17) Poniżej potwierdzenie poprawności działania narzędzia z komunikatem ,



- 18) Na koniec widok tabeli atrybutów po wykonaniu procesu stworzonego narzędzia IMGW.



	kod_stacji	stacja	rzeka	rok_hydro	miesiac_hydro	stan_wody	przeplyw	ESRI_OID
▶	149180020	CHALUPKI	Odra (1)	2013	1	106	15,8	1
	149180020	CHALUPKI	Odra (1)	2013	1	138	34,3	2
	149180020	CHALUPKI	Odra (1)	2013	1	251	128	3
	149180020	CHALUPKI	Odra (1)	2013	2	92	10,2	4
	149180020	CHALUPKI	Odra (1)	2013	2	116	20,3	5
	149180020	CHALUPKI	Odra (1)	2013	2	150	39,6	6
	149180020	CHALUPKI	Odra (1)	2013	3	105	15,4	7
	149180020	CHALUPKI	Odra (1)	2013	3	141	36,7	8
	149180020	CHALUPKI	Odra (1)	2013	3	330	224	9
	149180020	CHALUPKI	Odra (1)	2013	4	138	32,1	10
	149180020	CHALUPKI	Odra (1)	2013	4	175	63,5	11
	149180020	CHALUPKI	Odra (1)	2013	4	269	157	12
	149180020	CHALUPKI	Odra (1)	2013	5	138	32,4	13
	149180020	CHALUPKI	Odra (1)	2013	5	186	72,9	14
	149180020	CHALUPKI	Odra (1)	2013	5	241	127	15
	149180020	CHALUPKI	Odra (1)	2013	6	153	43,3	16
	149180020	CHALUPKI	Odra (1)	2013	6	220	108	17
	149180020	CHALUPKI	Odra (1)	2013	6	328	222	18
	149180020	CHALUPKI	Odra (1)	2013	7	135	29,8	19
	149180020	CHALUPKI	Odra (1)	2013	7	178	66,8	20
	149180020	CHALUPKI	Odra (1)	2013	7	317	210	21
	149180020	CHALUPKI	Odra (1)	2013	8	144	36,2	22
	149180020	CHALUPKI	Odra (1)	2013	8	217	106	23
	149180020	CHALUPKI	Odra (1)	2013	8	415	318	24
	149180020	CHALUPKI	Odra (1)	2013	9	90	9,9	25
	149180020	CHALUPKI	Odra (1)	2013	9	114	19,9	26
	149180020	CHALUPKI	Odra (1)	2013	9	159	48,2	27
	149180020	CHALUPKI	Odra (1)	2013	10	82	7,04	28
	149180020	CHALUPKI	Odra (1)	2013	10	92	11,2	29
	149180020	CHALUPKI	Odra (1)	2013	10	119	21,2	30
	149180020	CHALUPKI	Odra (1)	2013	11	82	9,03	31
	149180020	CHALUPKI	Odra (1)	2013	11	111	20,7	32
	149180020	CHALUPKI	Odra (1)	2013	11	167	55,9	33
	149180020	CHALUPKI	Odra (1)	2013	12	91	11,6	34
	149180020	CHALUPKI	Odra (1)	2013	12	400	10,2	35

3. Podsumowanie

Widzimy zatem że wynikiem naszych działań jest odkodowana tabela atrybutów z opisanymi dokładnie kolumnami, w tym przypadku z 2013 roku z bazy danych hydro. Możemy oczywiście wybrać drugą z baz czyli meteo, bądź też dowolnie inny rok pomiarów z lat 2012 – 2020. Mamy zatem całkiem sporą dowolność w tym zakresie w zależności co dokładnie nas interesuje.