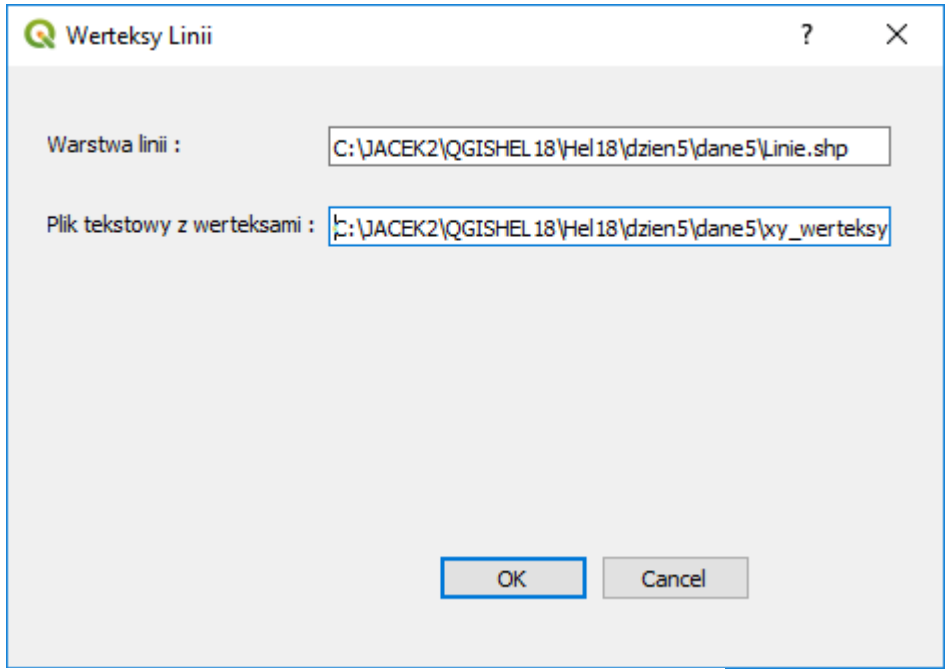
Kurs programowania w QGIS za pomocą Pythona

Dzień 5
Tworzenie wtyczek (plugins)

Hel wrzesień 2019

Wtyczka:



Wprowadzamy warstwę linii

otrzymujemy plik tekstowy werteksów

numer_linii, numer_werteksu, x,y

linie_werteksy.py | xy_werteksy.txt

```
0,0,439709.85,706167.63
0,1,439572.82,706163.21
0,2,439491.05,706154.37
0,3,439438.01,706132.27
0,4,439400.44,706074.81
0,5,439369.5,706021.77
0,6,439329.72,705981.99
0,7,439307.62,705940.0
0,8,439278.89,705926.74
0,9,439221.43,705895.8
0,10,439163.97,705878.12
0,11,439110.92,705844.97
0,12,439071.14,705833.92
0,13,439007.05,705814.03
0,14,438883.29,705787.51
0,15,438812.57,705685.84
1,0,439654.6,706435.05
1,1,439594.92,706410.74
1,2,439590.5,706328.97
1,3,439541.88,706315.71
1,4,439506.52,706267.09
1,5,439504.31,706225.09
1,6,439491.05,706154.37
2,0,438752.9,706236.15
2,1,438805.94,706127.85
2,2,438878.87,706081.44
2,3,438951.8,706043.87
2,4,438993.79,706032.82
```

Do wykonania wtyczki potrzebujemy:

1. Qt Creator (a właściwie jego element **Qt Designer**)
software development framework

C:\OSGeo4W64\apps\Qt5\bin\designer.exe

2. Program **pb_tool** i konfiguracja środowiska za pomocą  pyqgis.bat

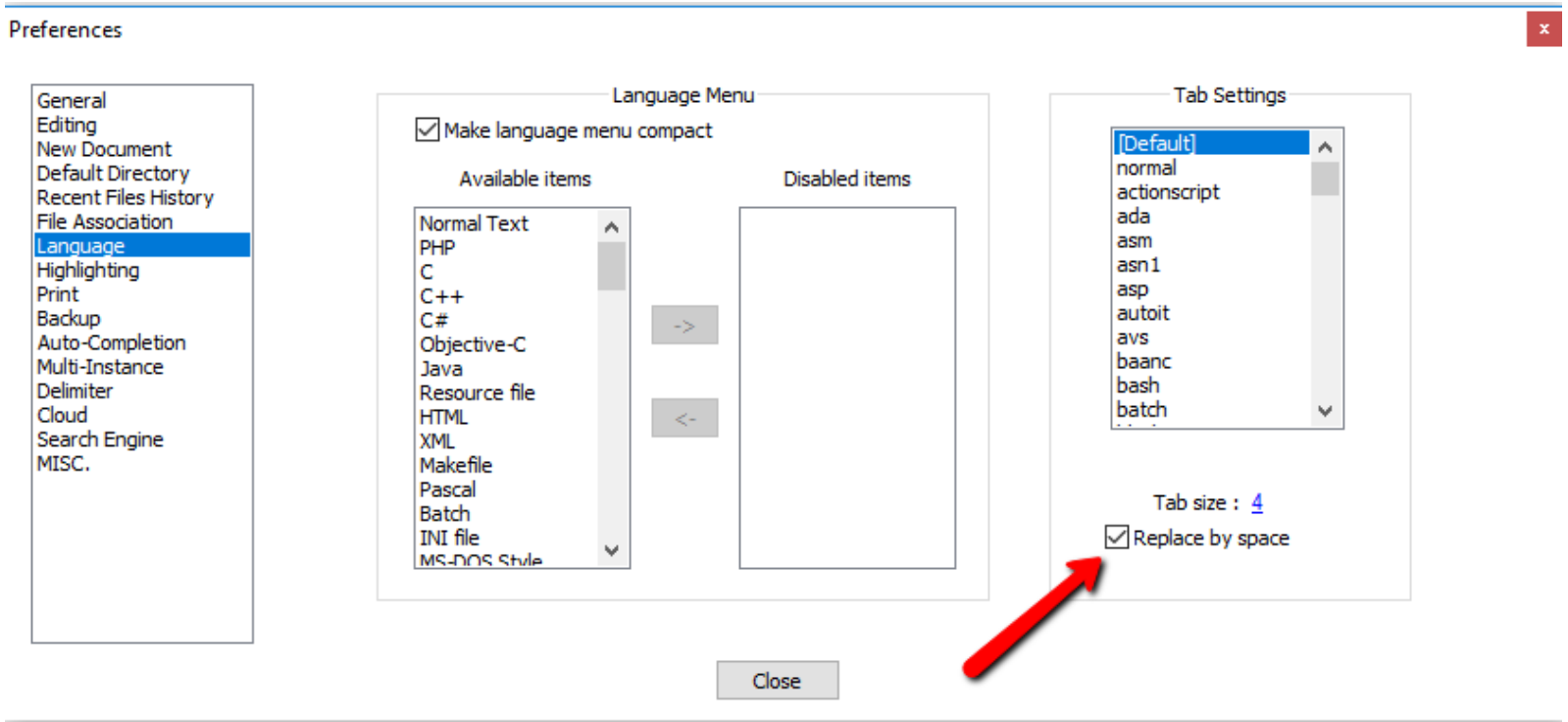
3. **Notpad++**
(inny edytor)

4. Wtyczki

Builder plugin

Reloader plugin

Settings



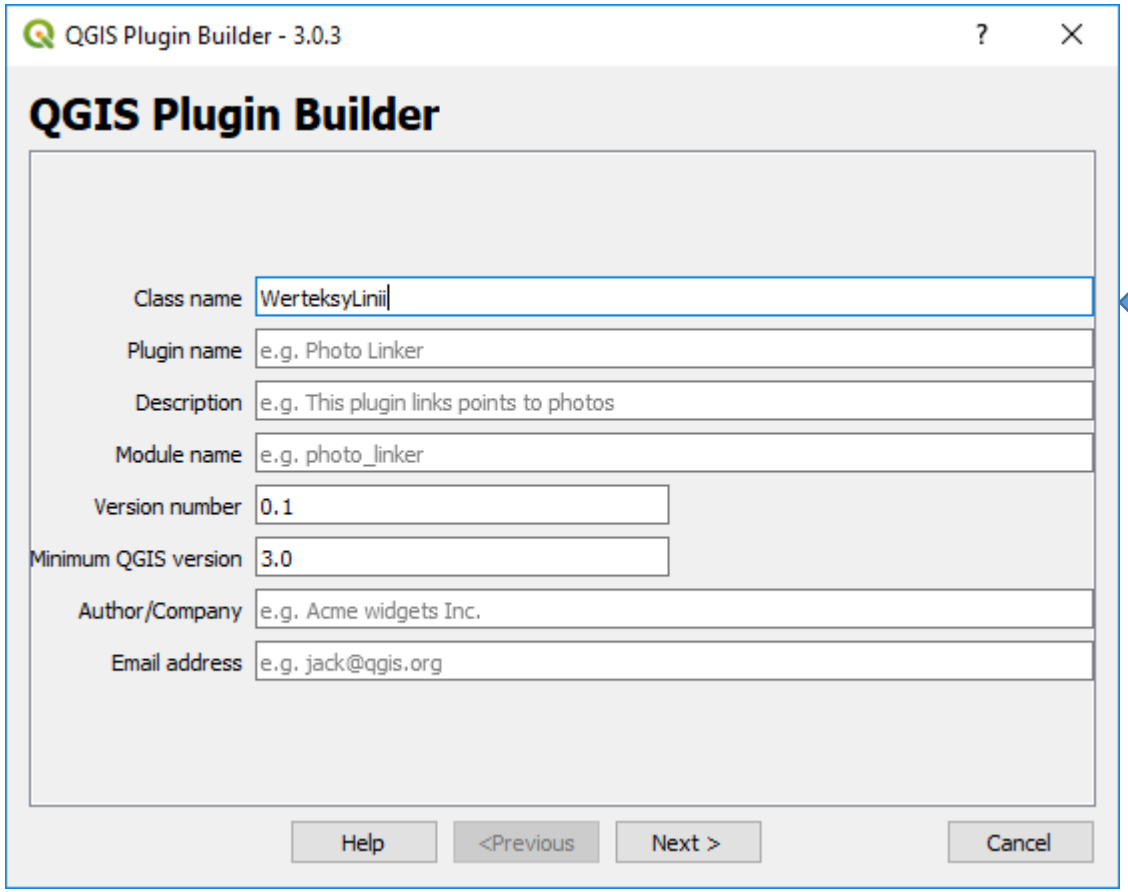
KROK 1

Uruchomić program w Python Consoli i przeprowadzić testowanie, określić parametry

```
from osgeo import ogr


driver=ogr.GetDriverByName('ESRI shapefile')

# parametry
ff=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\Linie.shp"

p_text=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\xy_werteksy.txt"


dataSource=driver.Open(ff,0)  # czytanie i pisanie

vlayer=dataSource.GetLayer()
#print vlayer.GetGeomType()
numFeature=vlayer.GetFeatureCount()
#print numFeature
file=open(p_text,"w")
for i in range (0, numFeature):
    feature=vlayer.GetFeature(i)

    geometry=feature.GetGeometryRef()
    nwtx=geometry.GetPointCount()

    for j in range(0,nwtx):
        x=geometry.GetX(j)
        y=geometry.GetY(j)

        ss= str(i)+','+str( j)+','+str( round(x,2))+','+str(round(y,2))+'\n'
        file.write(ss)
file.close()
feature.Destroy()
```

KROK 2

Uruchomić wtyczkę QGIS Plugin Builder



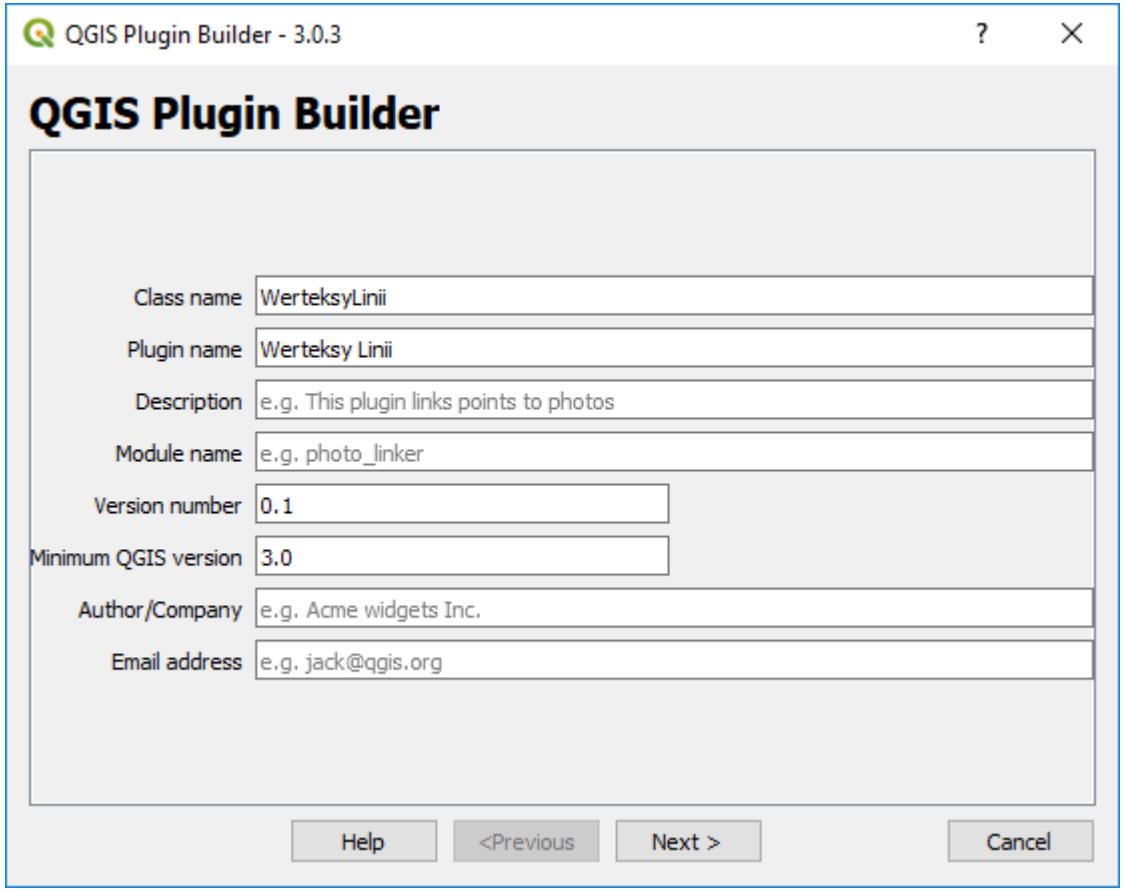
Nazwa Python Class zawierająca cały układ wtyczki

>> nazwa katalogu **(ciągła pisowania)**

Np. WerteksyLinii

KROK 2

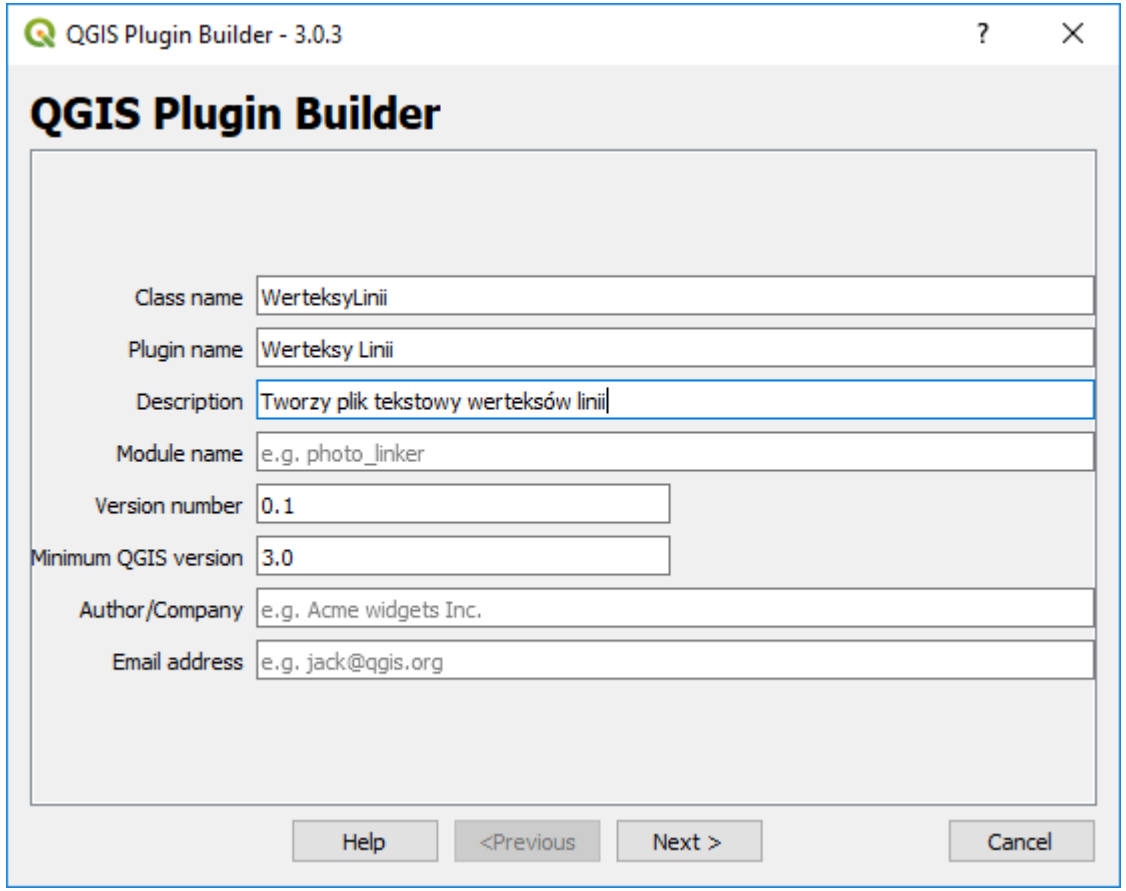
Uruchomić wtyczkę QGIS Plugin Builder



Nazwa w Plugin manager
>> dowolna
Np. Werteksy linii

KROK 2

Uruchomić wtyczkę QGIS Plugin Builder

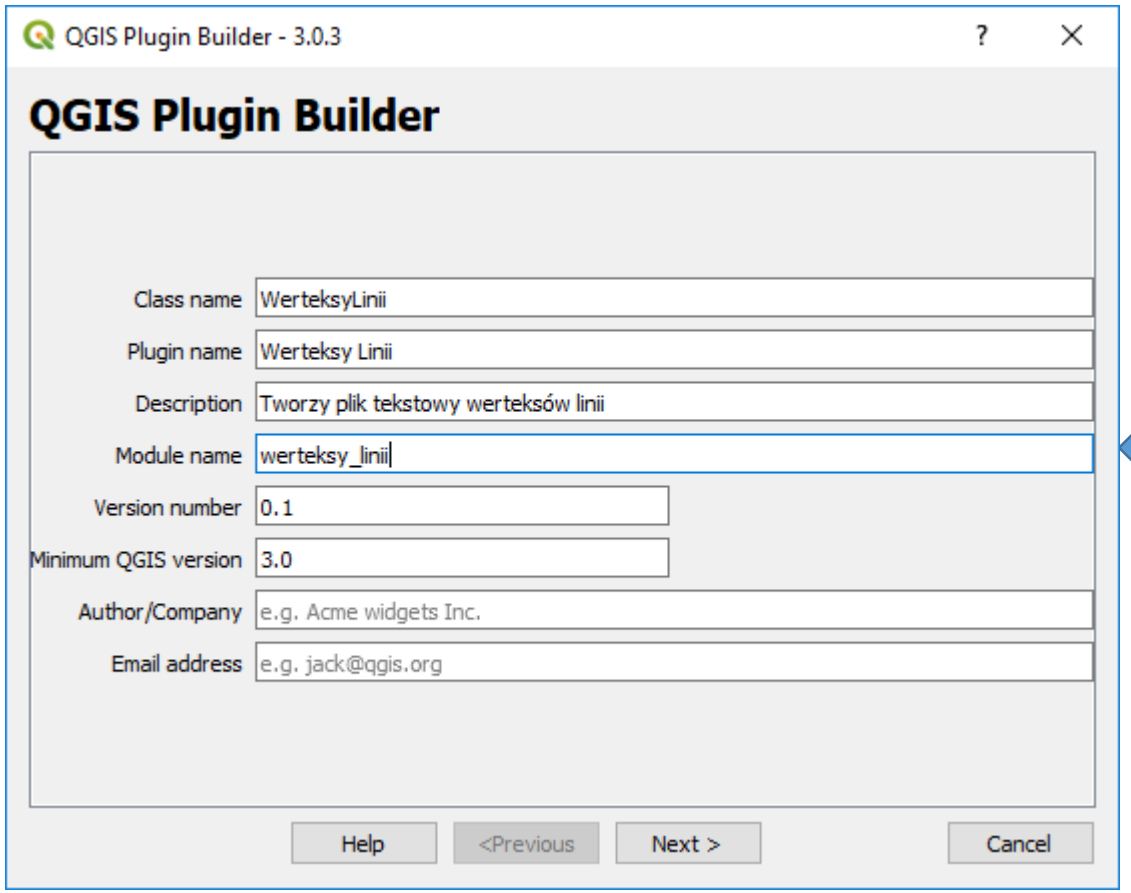


Krótki opis wtyczki 2-3 słowa

Np.. Tworzy plik tekstowy werteksów linii

KROK 2

Uruchomić wtyczkę QGIS Plugin Builder



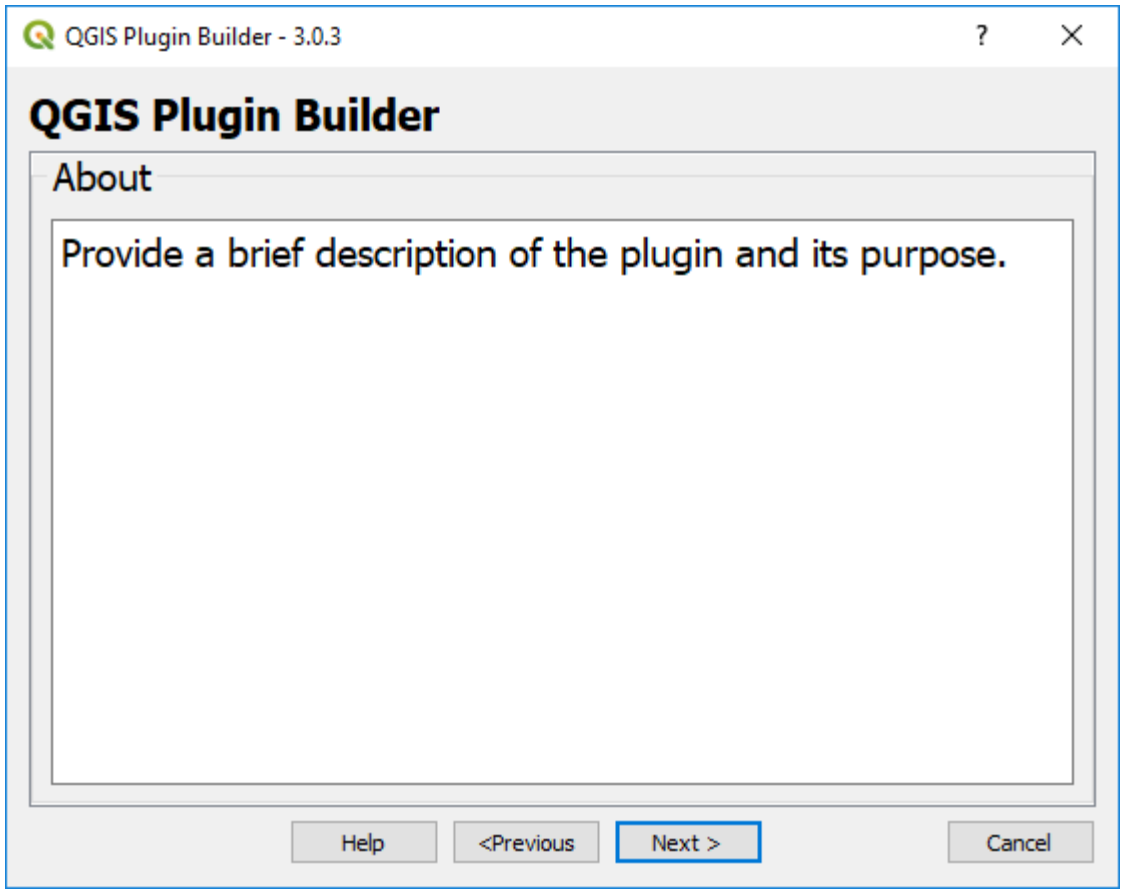
Nazwa głównego pliku Pythona

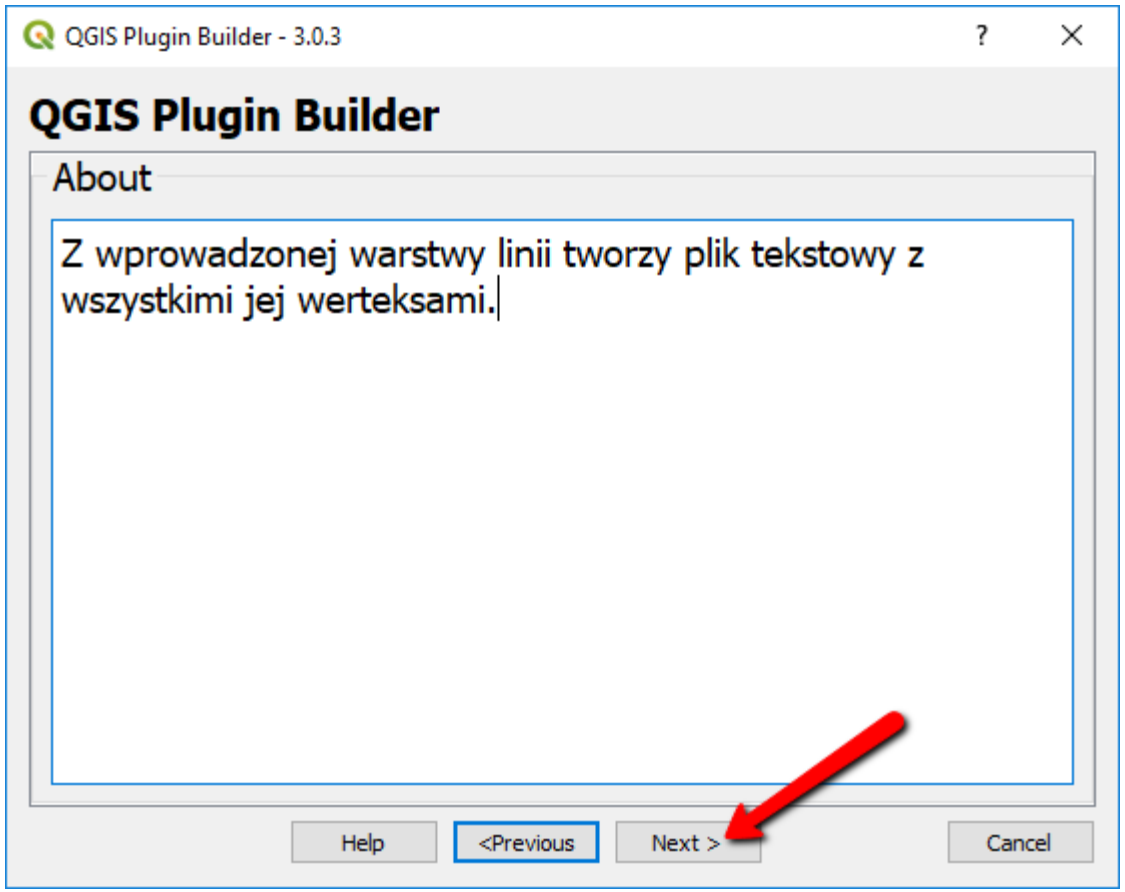
Np.. werteksy_linii

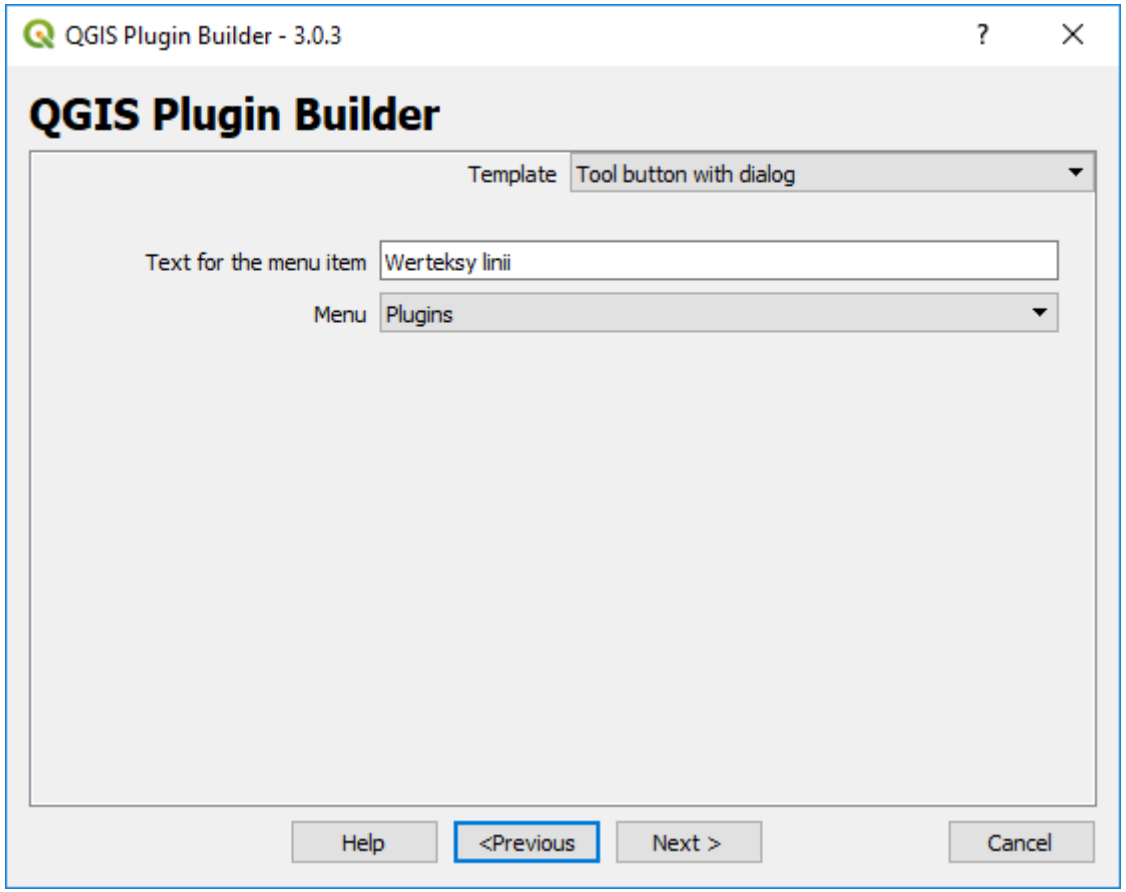
werteksy_linii.py

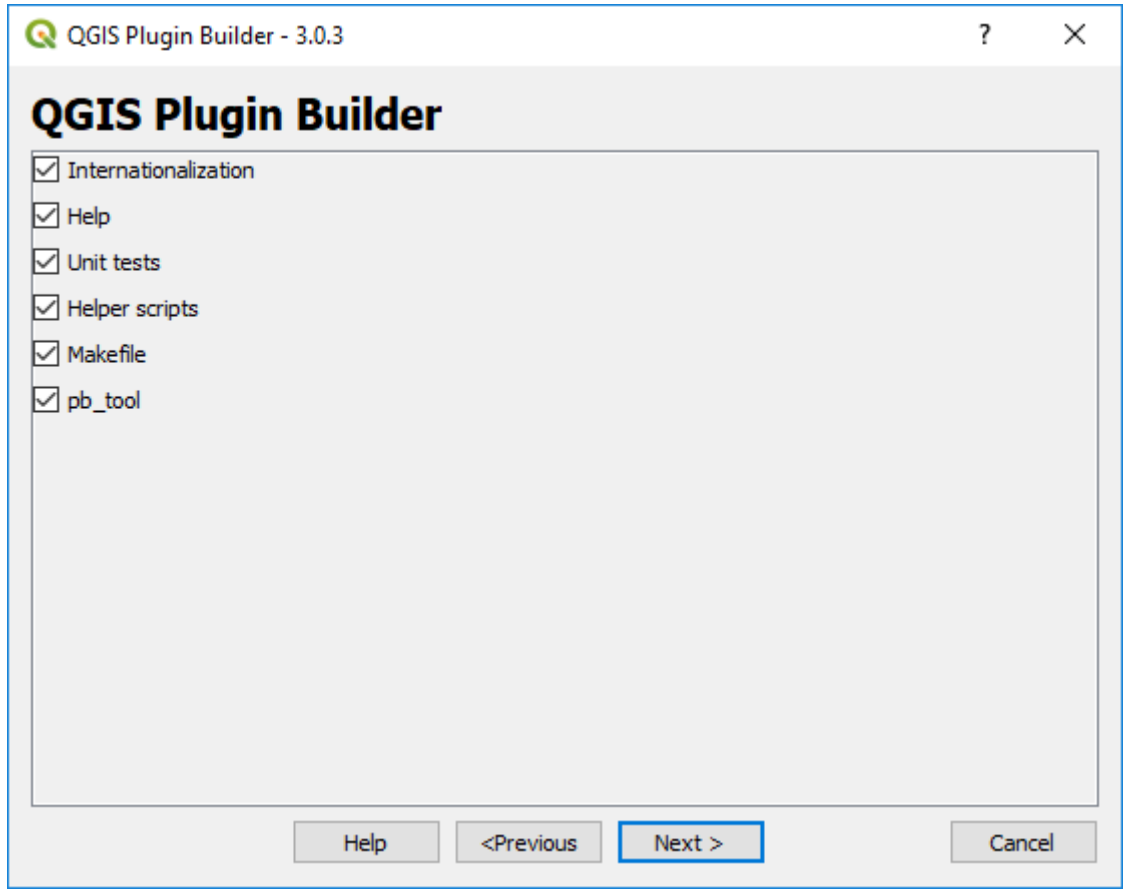
KROK 2

Uruchomić wtyczkę QGIS Plugin Builder



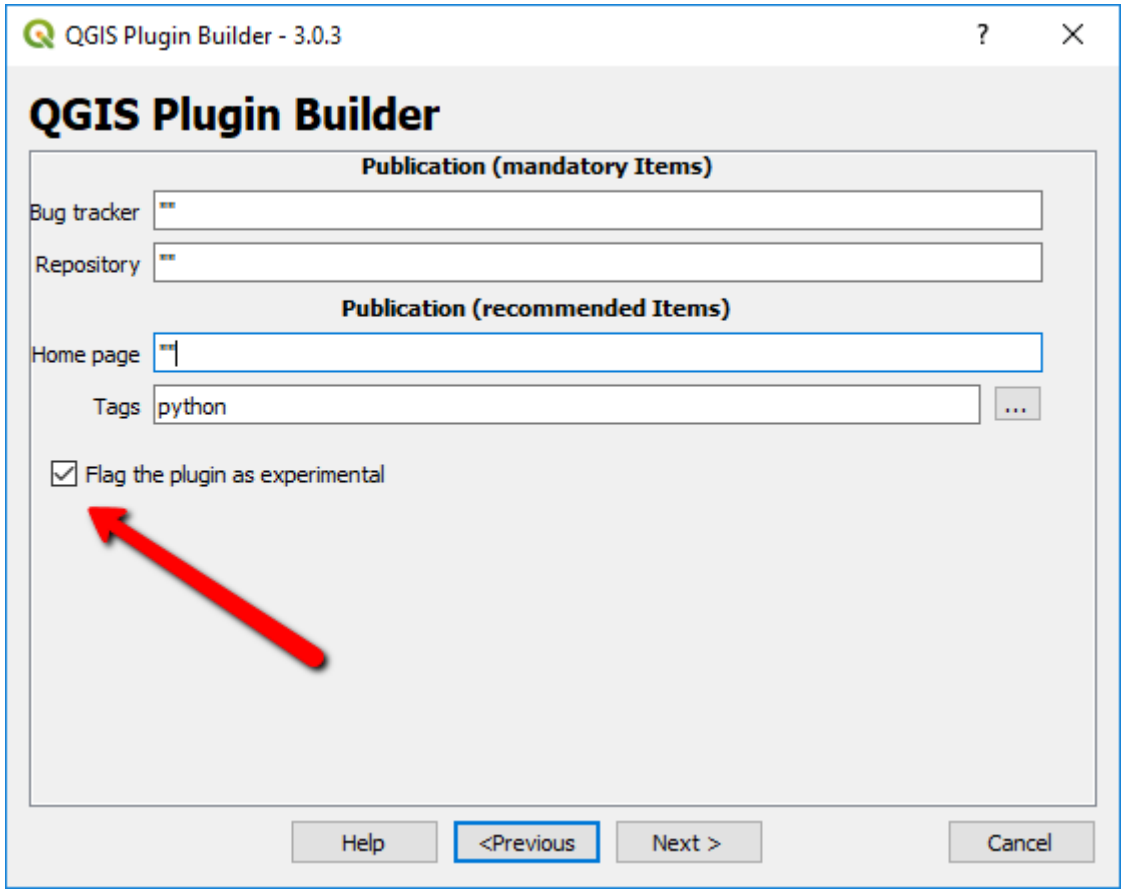



QGIS Plugin Builder - 3.0.3
QGIS Plugin Builder
Template
Tool button with dialog
Text for the menu item
Werteksy linii
Menu
Plugins
Help
<Previous
Next >
Cancel


QGIS Plugin Builder - 3.0.3
QGIS Plugin Builder
Internationalization
Help
Unit tests
Helper scripts
Makefile
pb_tool
Help
<Previous
Next >
Cancel

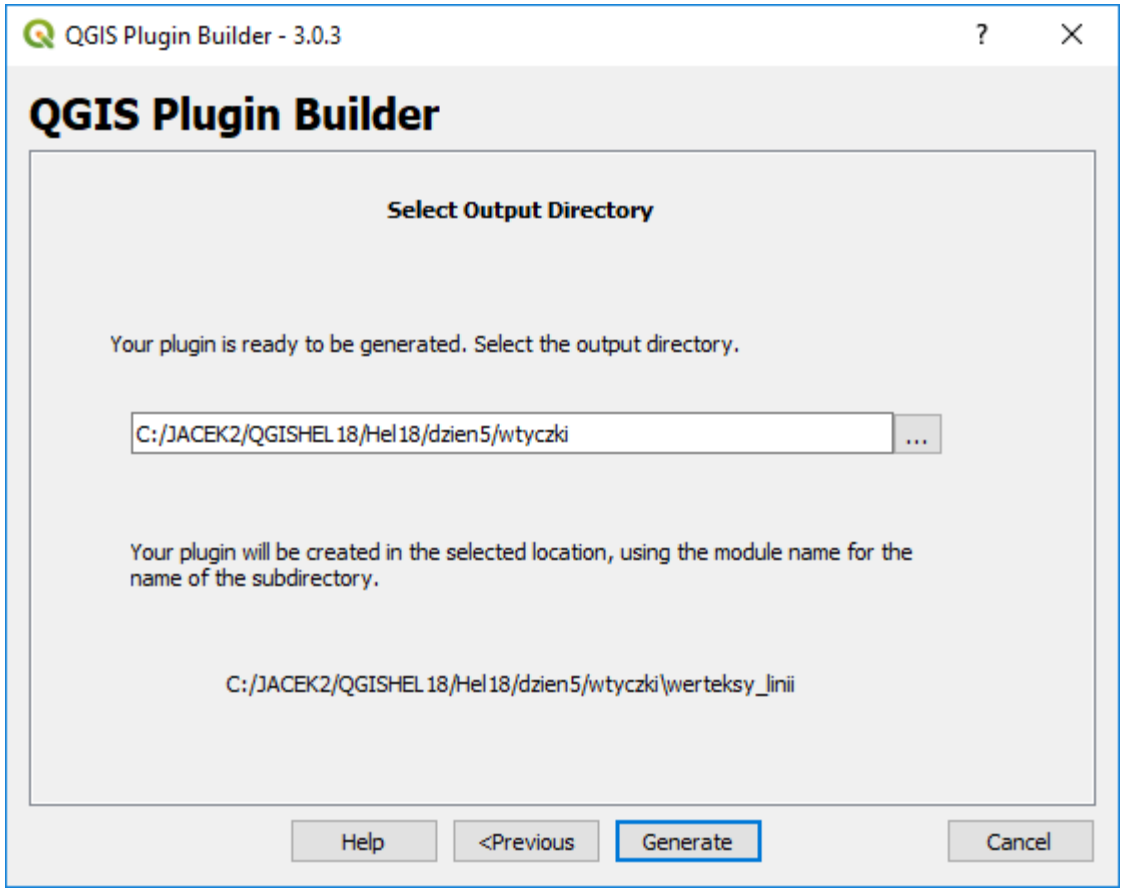
KROK 2

Uruchomić wtyczkę QGIS Plugin Builder

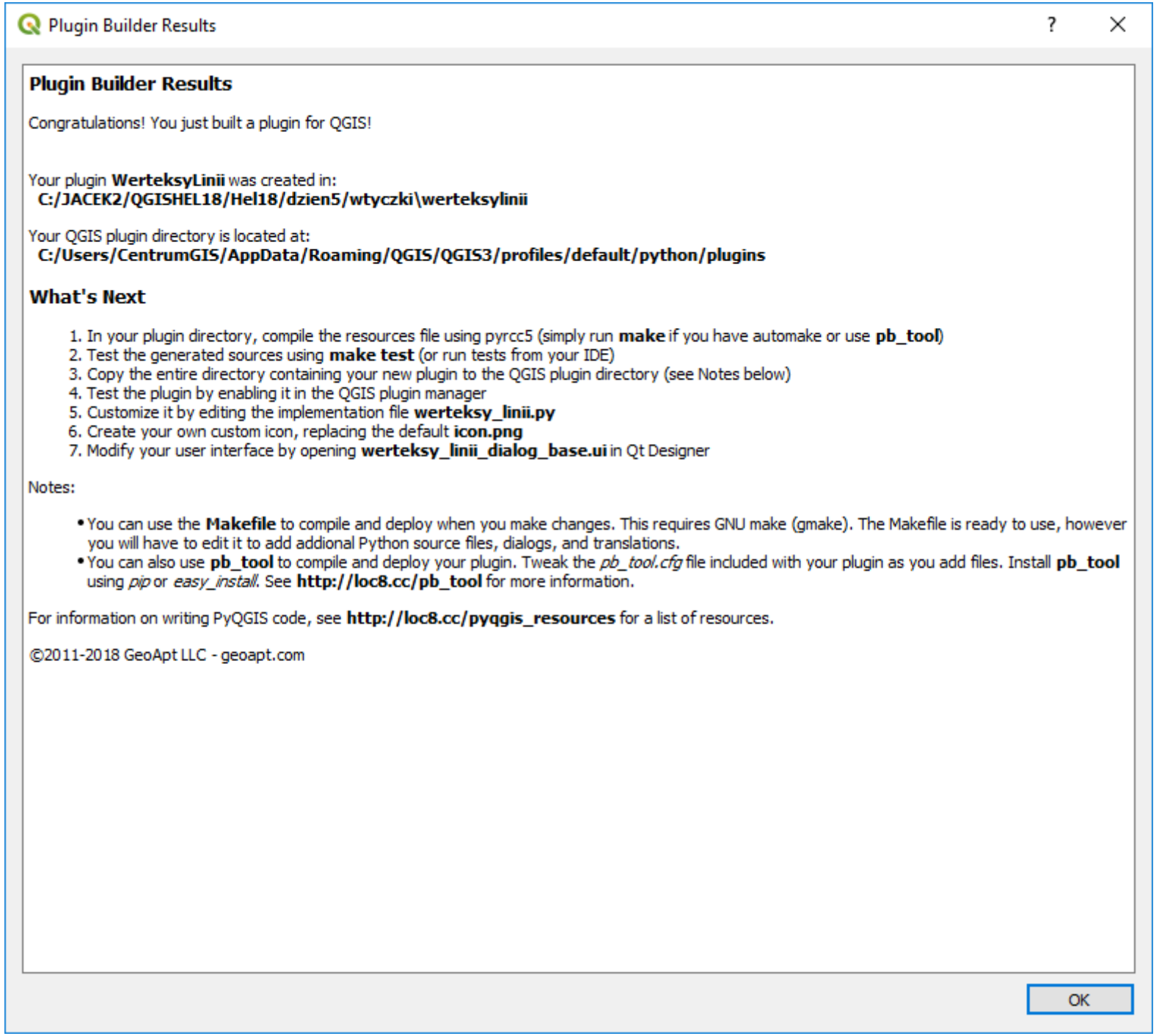


KROK 2

Uruchomić wtyczkę QGIS Plugin Builder



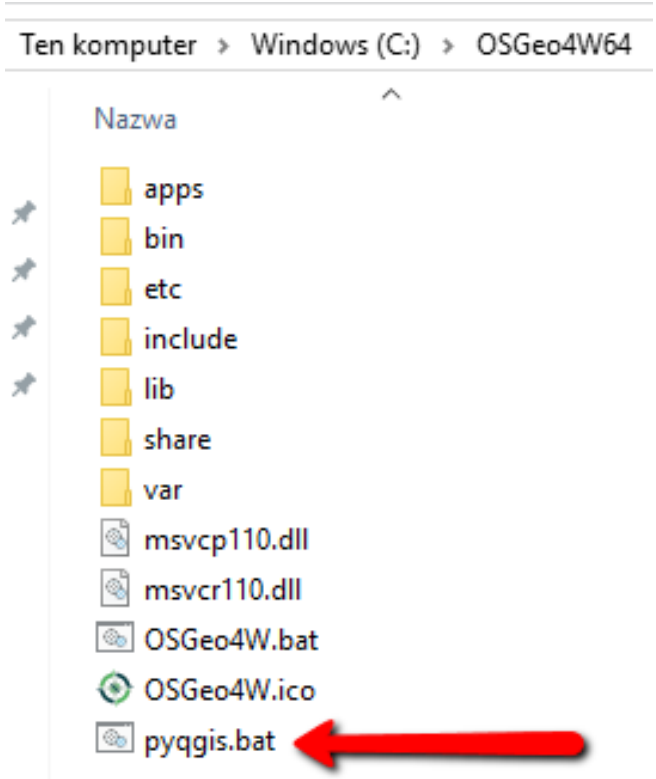
KROK 2 koniec

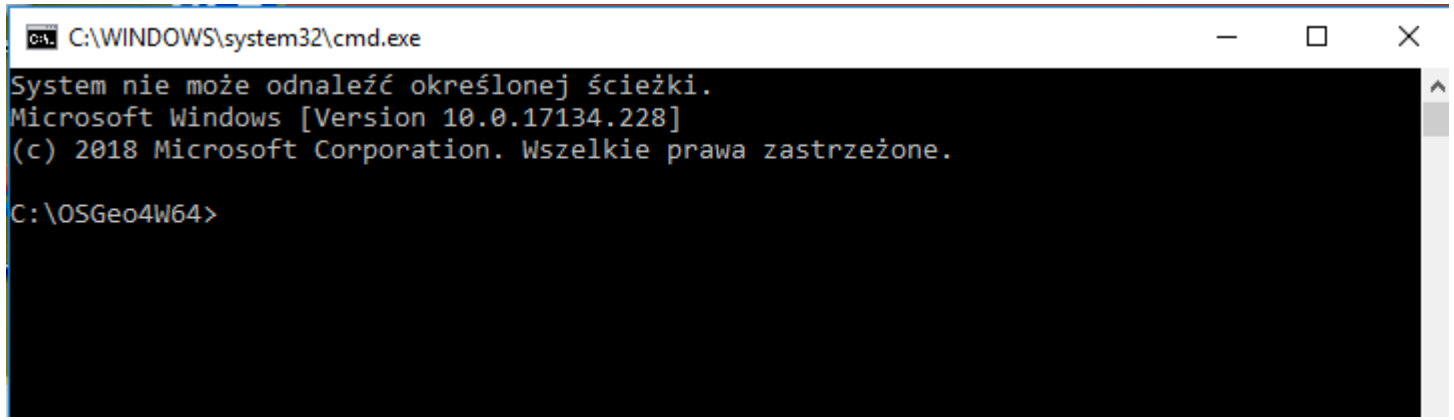


Windows (C:) › JACEK2 › QGISHEL18 › Hel18 › dzien5 › wtyczki › werteksylinii

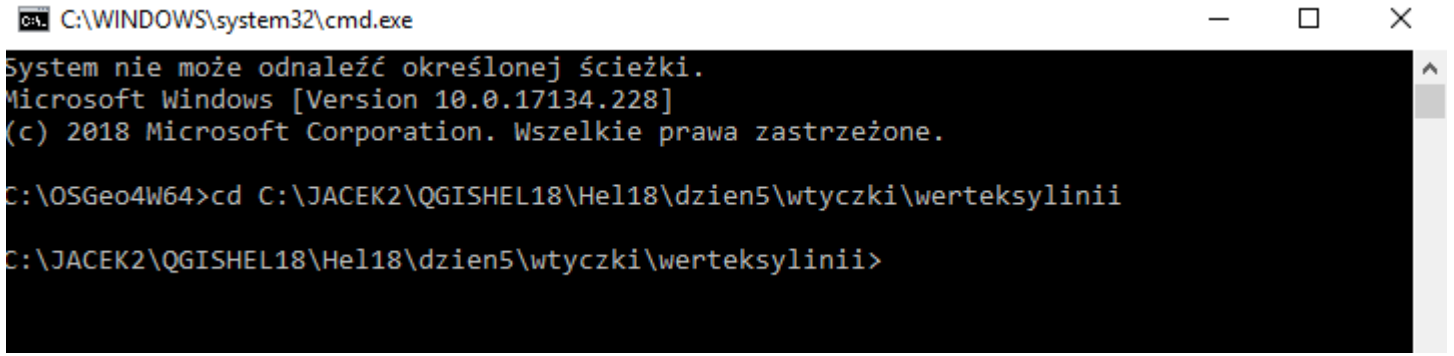
help	05.09.2018 10:03	Folder plików	
i18n	26.02.2018 17:47	Folder plików	
scripts	26.02.2018 17:47	Folder plików	
test	05.09.2018 10:03	Folder plików	
__init__.py	05.09.2018 10:03	Python File	2 KB
icon.png	26.02.2018 17:47	Plik PNG	2 KB
Makefile	05.09.2018 10:03	Plik	8 KB
metadata.txt	05.09.2018 10:03	Dokument tekstowy	1 KB
pb_tool.cfg	05.09.2018 10:03	Plik CFG	3 KB
plugin_upload.py	26.02.2018 17:47	Python File	4 KB
pylintrc	26.02.2018 17:47	Plik	9 KB
README.html	05.09.2018 10:03	Chrome HTML Do...	2 KB
README.txt	05.09.2018 10:03	Dokument tekstowy	1 KB
resources.qrc	05.09.2018 10:03	Plik QRC	1 KB
werteksy_linii.py	05.09.2018 10:03	Python File	7 KB
werteksy_linii_dialog.py	05.09.2018 10:03	Python File	2 KB
werteksy_linii_dialog_base.ui	05.09.2018 10:03	Plik UI	2 KB

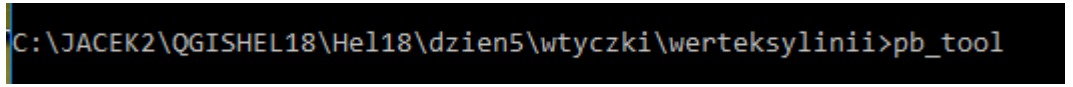
KROK 3





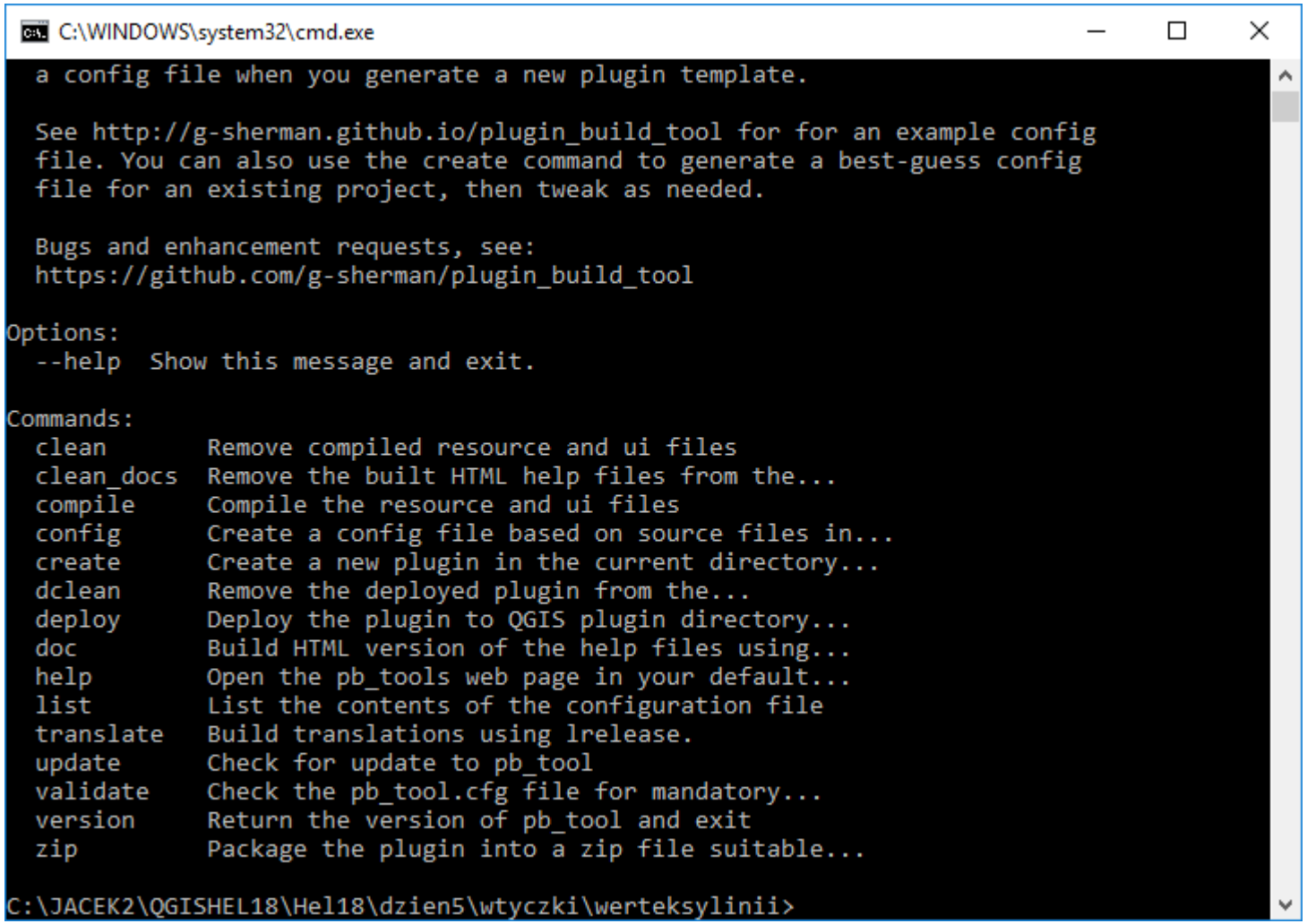
Idziemy do katalogu w którym mamy wtyczkę





KROK 3

Uruchamiamy pb_tool i otrzymujemy komendy



KROK 3

```
>pb_tool deploy
```

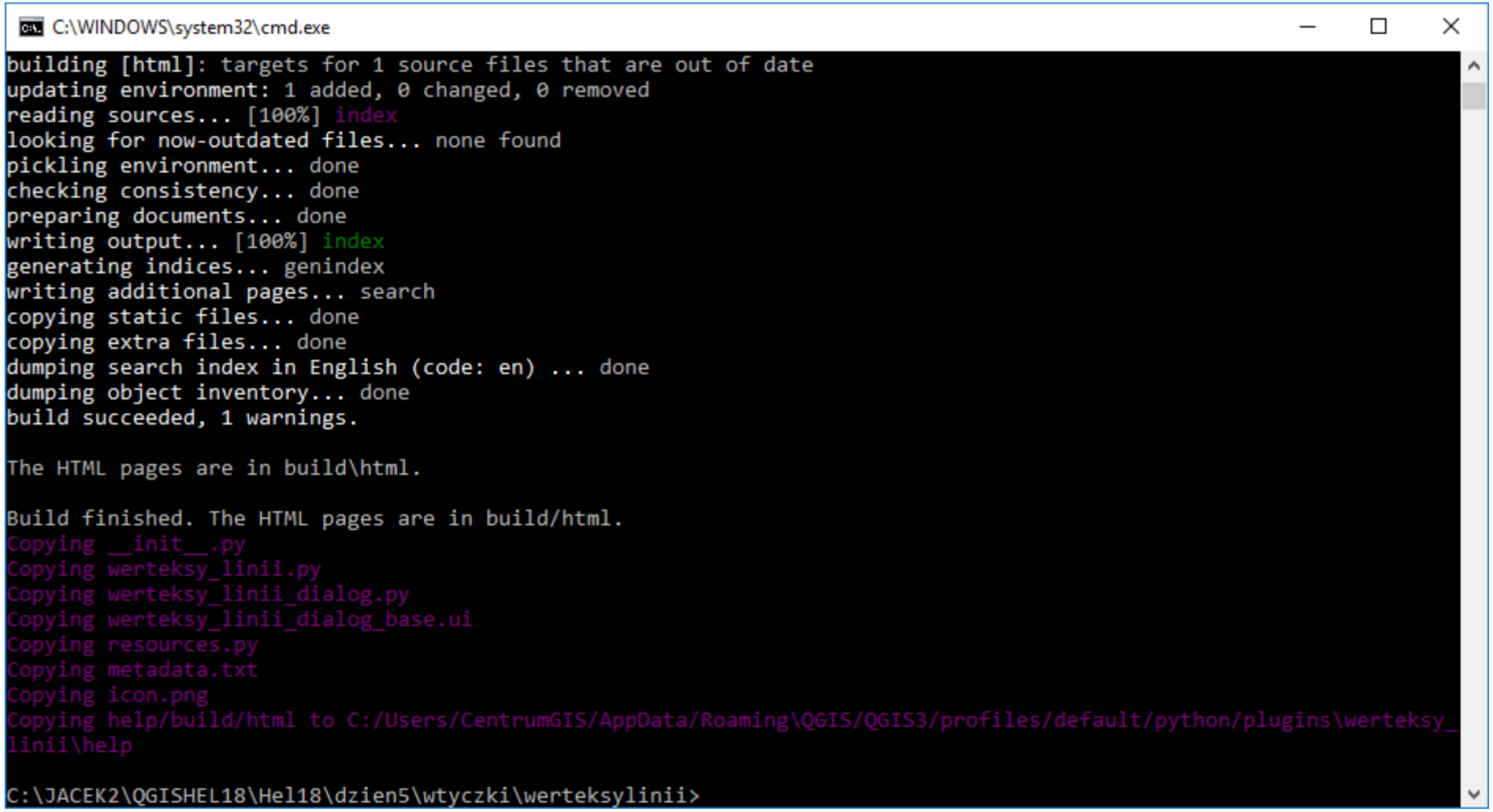
C:\WINDOWS\system32\cmd.exe - pb_tool deploy

```
Commands:
  clean       Remove compiled resource and ui files
  clean_docs  Remove the built HTML help files from the...
  compile     Compile the resource and ui files
  config      Create a config file based on source files in...
  create      Create a new plugin in the current directory...
  dclean      Remove the deployed plugin from the...
  deploy      Deploy the plugin to QGIS plugin directory...
  doc         Build HTML version of the help files using...
  help        Open the pb_tools web page in your default...
  list        List the contents of the configuration file
  translate   Build translations using lrelease.
  update      Check for update to pb_tool
  validate    Check the pb_tool.cfg file for mandatory...
  version     Return the version of pb_tool and exit
  zip         Package the plugin into a zip file suitable...

C:\JACEK2\QGISHEL18\Hel18\dzien5\wtyczki\werteksylinii>pb_tool deploy
QCoreApplication::applicationDirPath: Please instantiate the QApplication object first
QCoreApplication::applicationDirPath: Please instantiate the QApplication object first
Deploying to C:/Users/CentrumGIS/AppData/Roaming\QGIS/QGIS3/profiles/default/python/plugins\werteksy_linii
Deploying will:
               * Remove your currently deployed version
               * Compile the ui and resource files
               * Build the help docs
               * Copy everything to your C:/Users/CentrumGIS/AppData/Roaming\QGIS/QGIS3/profiles/default/python/pl
ugins\werteksy_linii directory

Proceed? [y/N]:
```

sprawdzamy

Your QGIS plugin directory is located at:
C:/Users/CentrumGIS/AppData/Roaming/QGIS/QGIS3/profiles/default/python/plugins

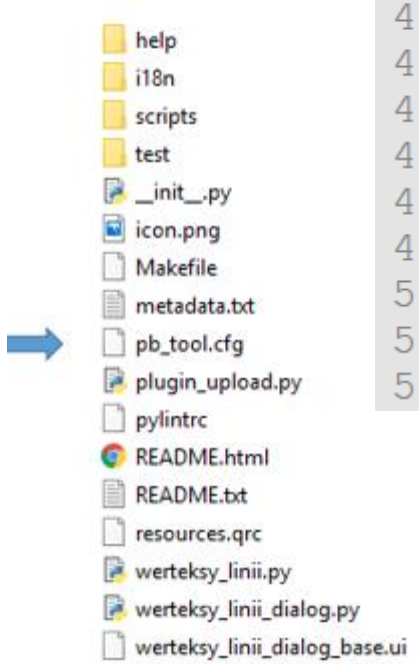
KROK 3



IF ERROR

IF ERROR

Otwieramy w Notpad++ pb_tool.cfg

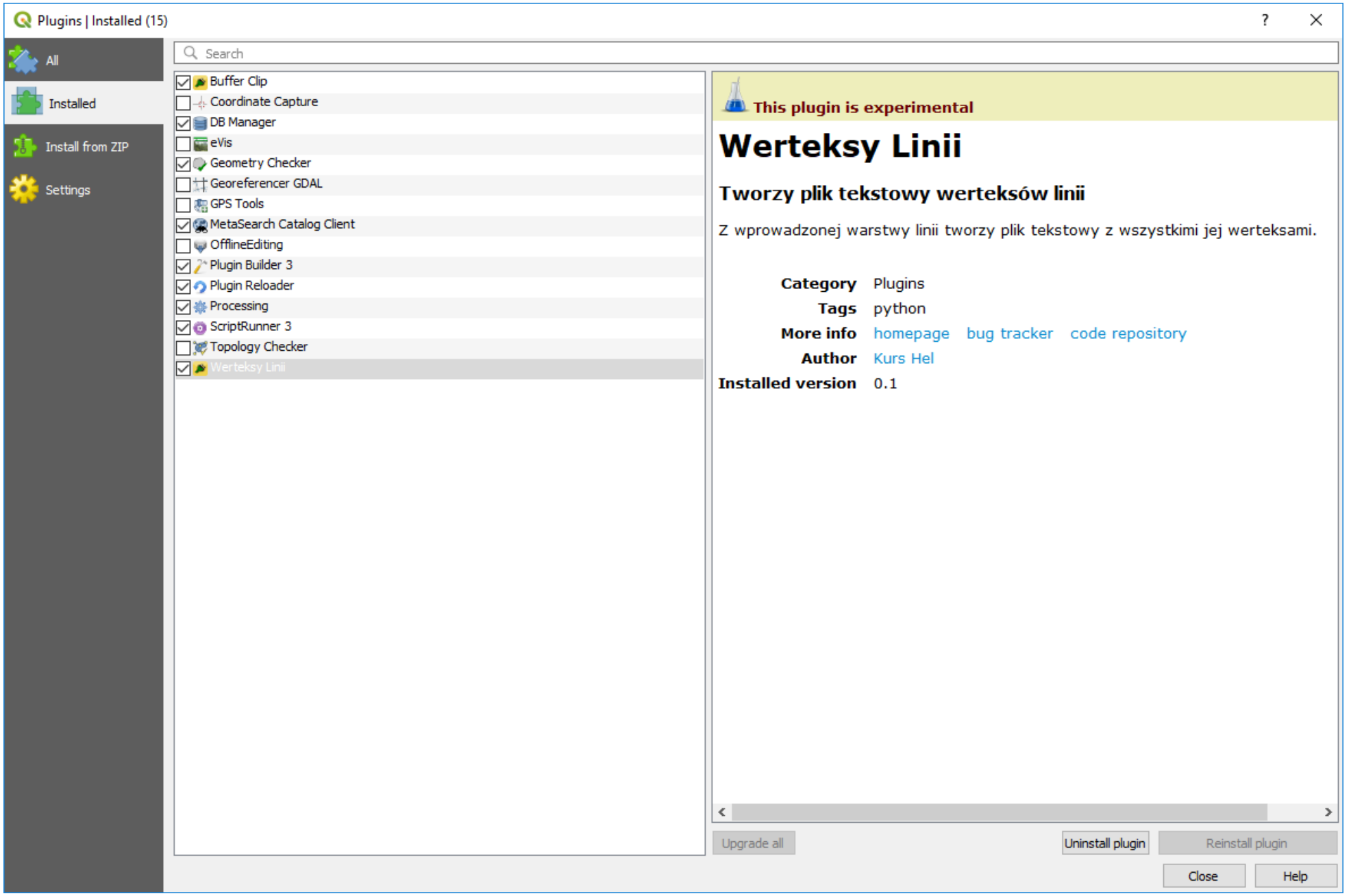


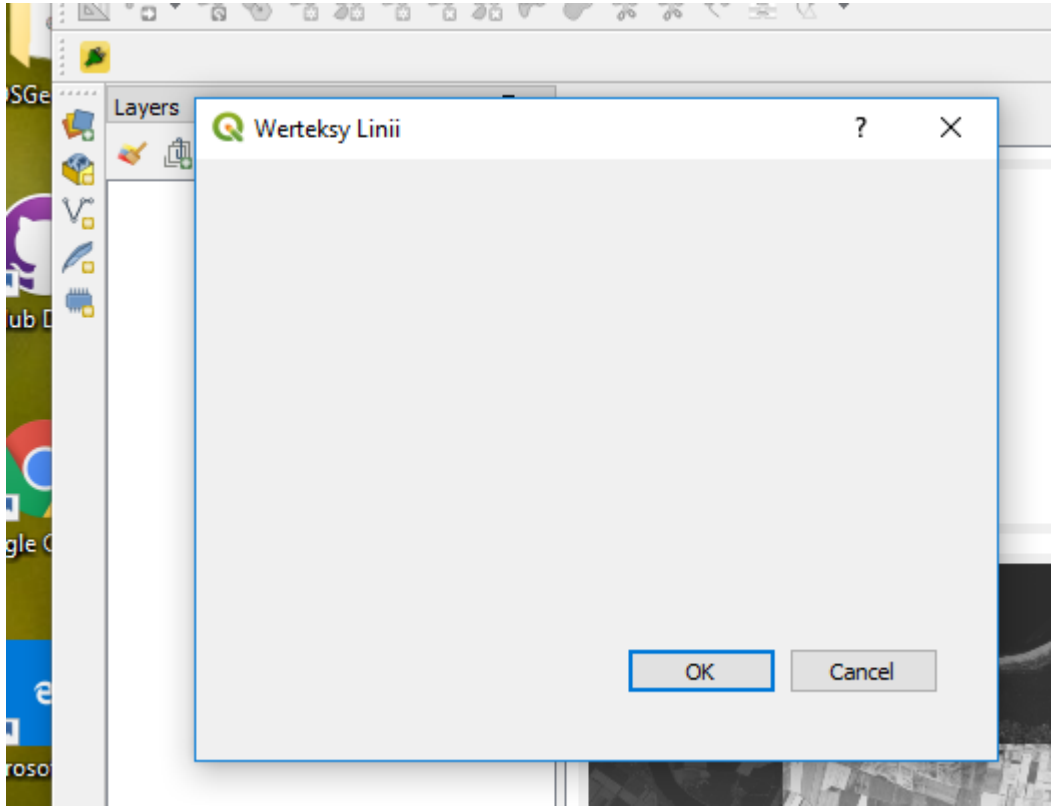

```
42  name: werteksy_linii
43
44  # Full path to where you want your plugin directory copied. If empty,
45  # the QGIS default path will be used. Don't include the plugin name in
46  # the path.
47  plugin_path:
48
49  [files]
50  # Python  fil   that should be deployed with the plugin
51  python_files:  __init__.py werteksy_linii.py werteksy_linii_dialog.py
52
```

C:/Users/CentrumGIS/AppData/Roaming/QGIS/QGIS3/profiles/default/python/plugins

KROK 4

Zamykamy i otwieramy QGISa, uaktywniamy wtyczkę





Wtyczka się uruchamia

base template

KROK 5 - **QT Designer (projektowanie GUI)**

Qt Designer

Plik Edycja Formularz Widok Ustawienia Okno Pomoc

Panel widżetów

Filtr

Layouts
- Vertical Layout
- Horizontal Layout
- Grid Layout
- Form Layout

Spacers
- Horizontal Spacer
- Vertical Spacer

Buttons
- Push Button
- Tool Button
- Radio Button
- Check Box
- Command Link Button
- Dialog Button Box

Item Views (Model-Base
- List View
- Tree View
- Table View
- Column View

Item Widgets (Item-Bas
- List Widget
- Tree Widget
- Table Widget

Containers
- Group Box
- Scroll Area
- Tool Box
- Tab Widget
- Stacked Widget
- Frame

Nowy formularz - Qt Designer

templates\forms
- Dialog with Buttons Bottom
- Dialog with Buttons Right
- Dialog without Buttons
- Main Window
- Widget

Widżety

Projekt dla urządzeń specjalizowanych

Urządzenie: Żadne

Rozmiar ekranu: Domyślny rozmiar

to okno przy uruchamianiu

Utwórz | Otwórz... | Ostatnie | Zamknij

Otwórz formularz

« dzien5 › wtyczki › werteksylinii

Przeszukaj: werteksylinii

Organizuj | Nowy folder

dzien4
GDS_Hel18
OneDrive
Ten komputer
Dokumenty
Muzyka
Obiekty 3D
Obrazy
Pobrane

Nazwa	Data modyfikacji	Typ
help	05.09.2018 10:03	Folder plików
i18n	26.02.2018 17:47	Folder plików
scripts	26.02.2018 17:47	Folder plików
test	05.09.2018 10:03	Folder plików
werteksy_linii_dialog_base.ui	05.09.2018 10:03	Plik UI

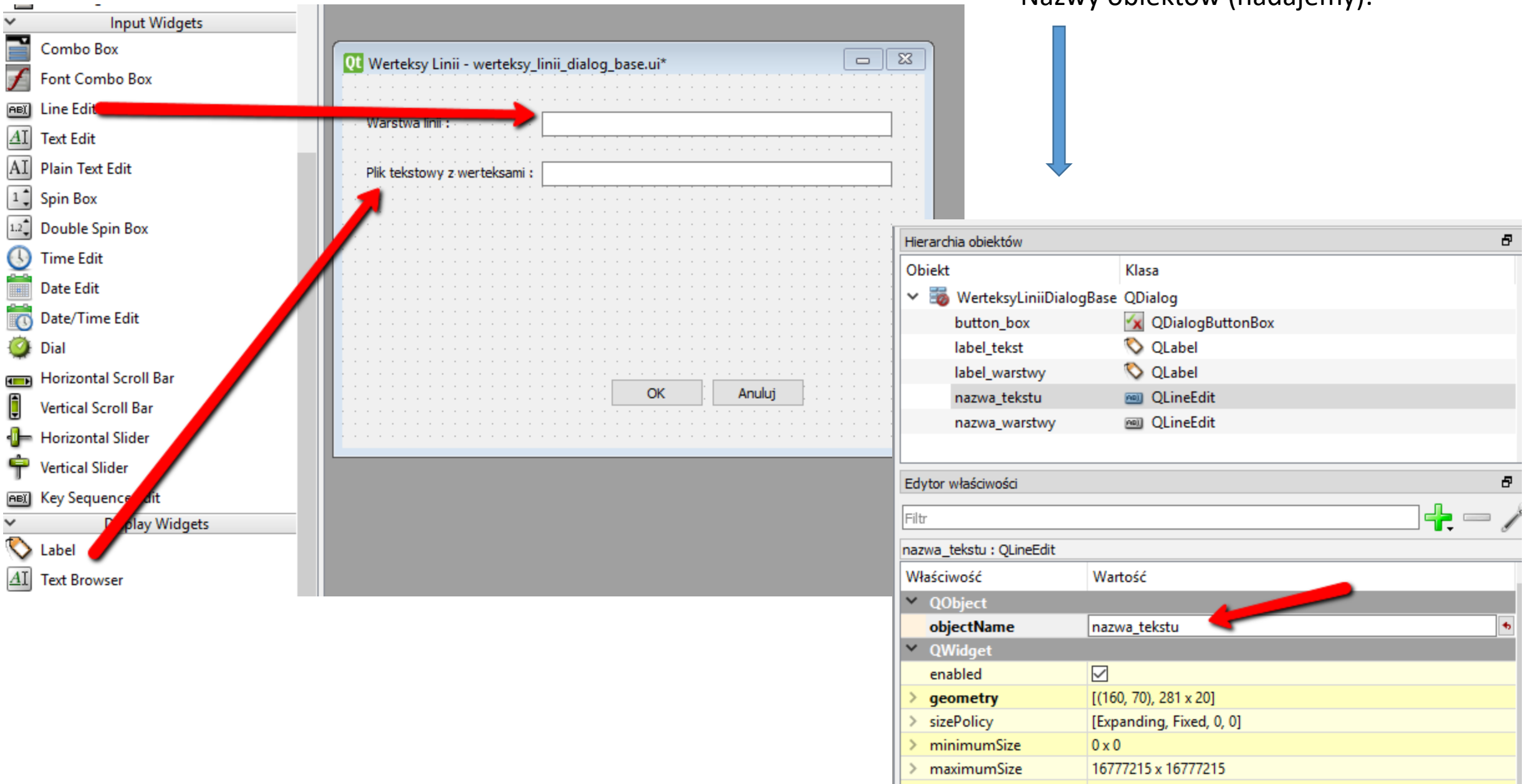
Nazwa pliku: werteksy_linii_dialog_base.ui

Pliki Designer UI (*.ui)

Otwórz | Anuluj

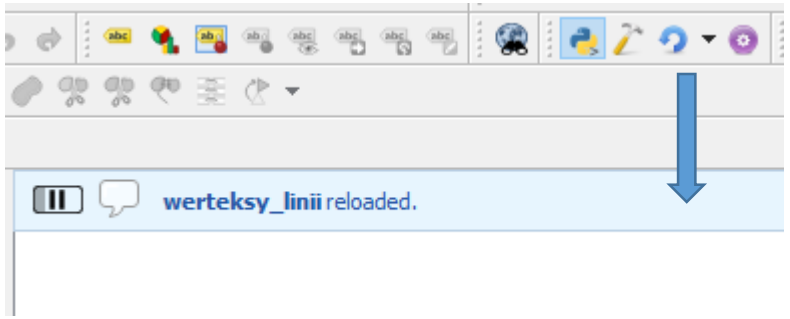
KROK 6 - **QT Designer**

Nazwy obiektów (nadajemy):

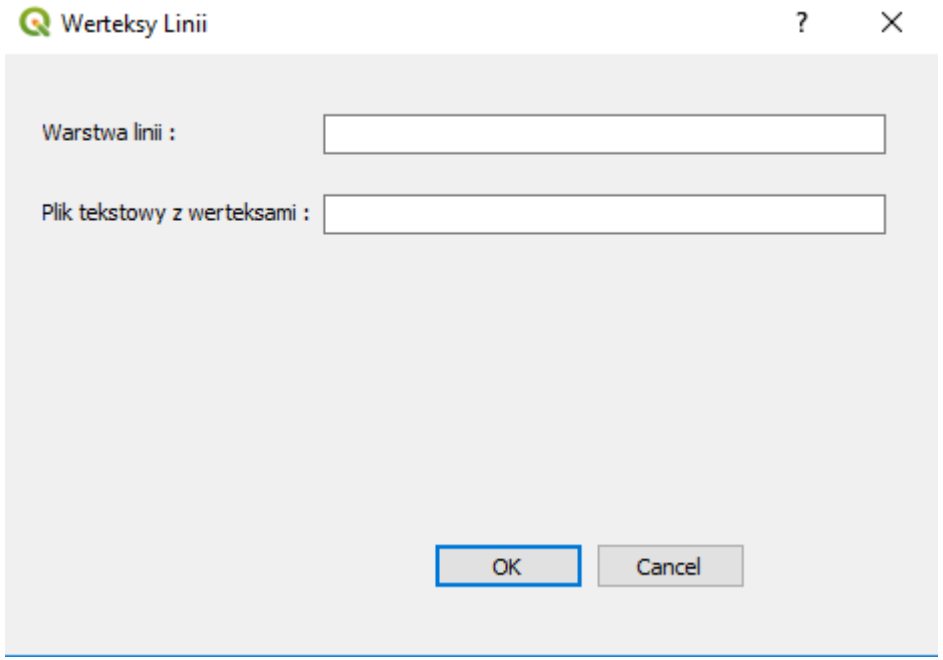


Po każdym użyciu Qt Designera (oraz zmianie kodu) i zapisaniu rezultatów należy:

```
C:\JACEK2\QGISHEL18\Hel18\dzien5\wtyczki\werteksylinii>pb_tool deploy
```

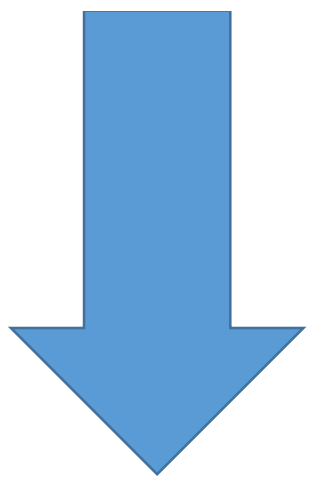


Używamy Reload Plugin po konfiguracji



KROK 6 - **Uzupełnianie kodu w** **nazwa_plugin.py**

```
def run(self):
    """Run method that performs all the real work"""
    # show the dialog
    self.dlg.show()
    # Run the dialog event loop
    result = self.dlg.exec_()
    # See if OK was pressed
    if result:
        # Do something useful here - delete the line containing pass and
        # substitute with your code.
        pass
```



KROK 6 -

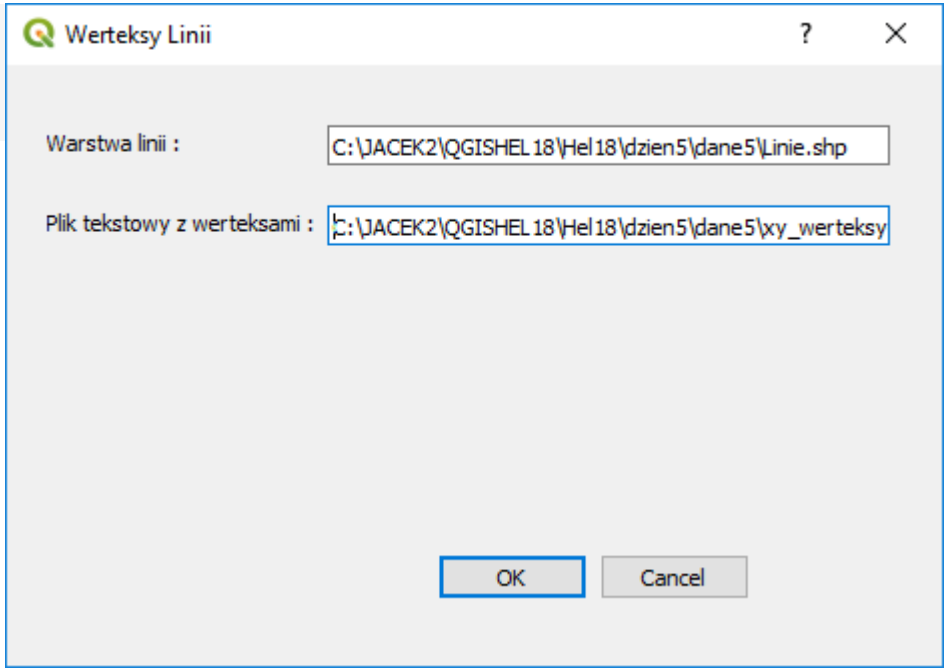
Uzupełnianie kodu

nazwa_plugin.py


```
32  import os.path
33  from osgeo import ogr
```

```
if result:
    # Do something useful here - delete the line containing pass and
    # substitute with your code.

    driver=ogr.GetDriverByName('ESRI shapefile')

    # parametry
    #ff=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\dane5\Linie.shp"

    #p_text=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\dane5\xy_werteksy.txt"
    ff=self.dlg.nazwa_warstwy.text()
    p_text=self.dlg.nazwa_tekstu.text()

    dataSource=driver.Open(ff,0)  # czytanie i pisanie
```

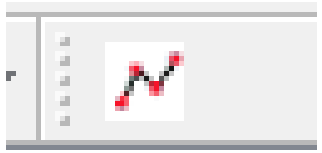
KROK 7 - **pb_tool Deploy** i **Reload plugin** 



```
0,0,439709.85,706167.63
0,1,439572.82,706163.21
0,2,439491.05,706154.37
0,3,439438.01,706132.27
0,4,439400.44,706074.81
0,5,439369.5,706021.77
0,6,439329.72,705981.99
0,7,439307.62,705940.0
0,8,439278.89,705926.74
0,9,439221.43,705895.8
0,10,439163.97,705878.12
0,11,439110.92,705844.97
0,12,439071.14,705833.92
0,13,439007.05,705814.03
0,14,438883.29,705787.51
0,15,438812.57,705685.84
1,0,439654.6,706435.05
1,1,439594.92,706410.74
1,2,439590.5,706328.97
1,3,439541.88,706315.71
1,4,439506.52,706267.09
1,5,439504.31,706225.09
1,6,439491.05,706154.37
2,0,438752.9,706236.15
2,1,438805.94,706127.85
```

KROK DODATKOWY **ZMIANA IKONKI** 24 x 24 piksele

Nowa ikonka do katalogu



Modyfikacja **resources.qrc**

```
<RCC>
    <qresource prefix="/plugins/LiniaWerteksy" >
        <file>icon.png</file>
    </qresource>
</RCC>
```

```
<RCC>
    <qresource prefix="/plugins/LiniaWerteksy" >
        <file>xy.png</file>
    </qresource>
</RCC>
```

Modyfikacja metadata.txt

```
homepage=""
category=Plugins
icon=icon.png
# experimental flag
experimental=True
```

```
homepage=""
category=Plugins
icon=xy.png
# experimental flag
```

Modyfikacja **nazwa_plugin.py (werteksy_linii.py)**

```
def initGui(self):
    """Create the menu entries and toolbar icons ir

    icon_path = ':/plugins/LiniaWerteksy/icon.png'
    self.add_action(
        icon_path,
        text=self.tr(u''),
        callback=self.run,
        parent=self.iface.mainWindow())
```

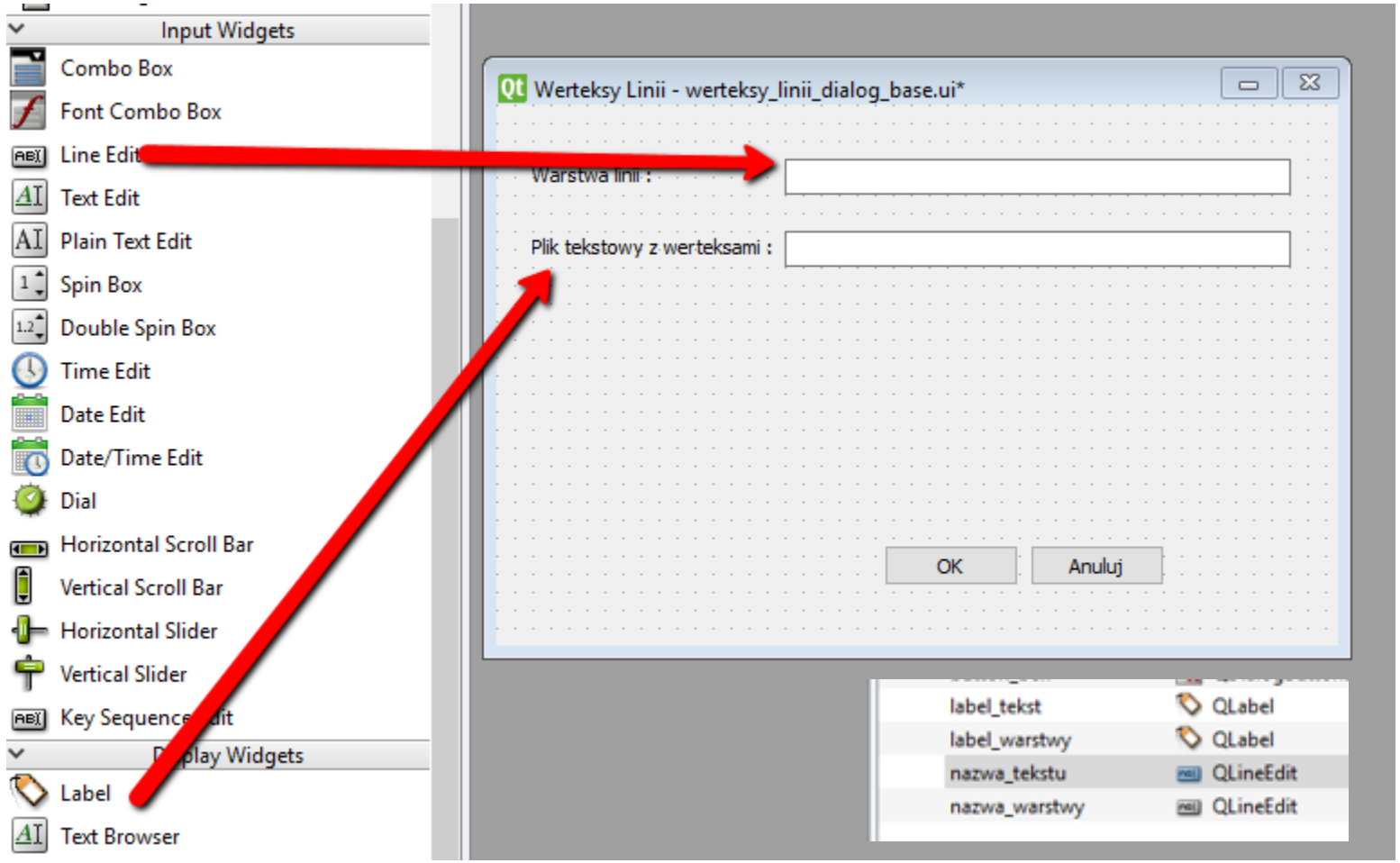
```
def initGui(self):
    """Create the menu entries and toolbar icons insi

    icon_path = ':/plugins/LiniaWerteksy/xy.png'
    self.add_action(
        icon_path,
        text=self.tr(u'Werteksy linii'),
        callback=self.run,
        parent=self.iface.mainWindow())
```

pb_tool deploy i Reload

PROJEKTOWANIE GUI

GUI 1

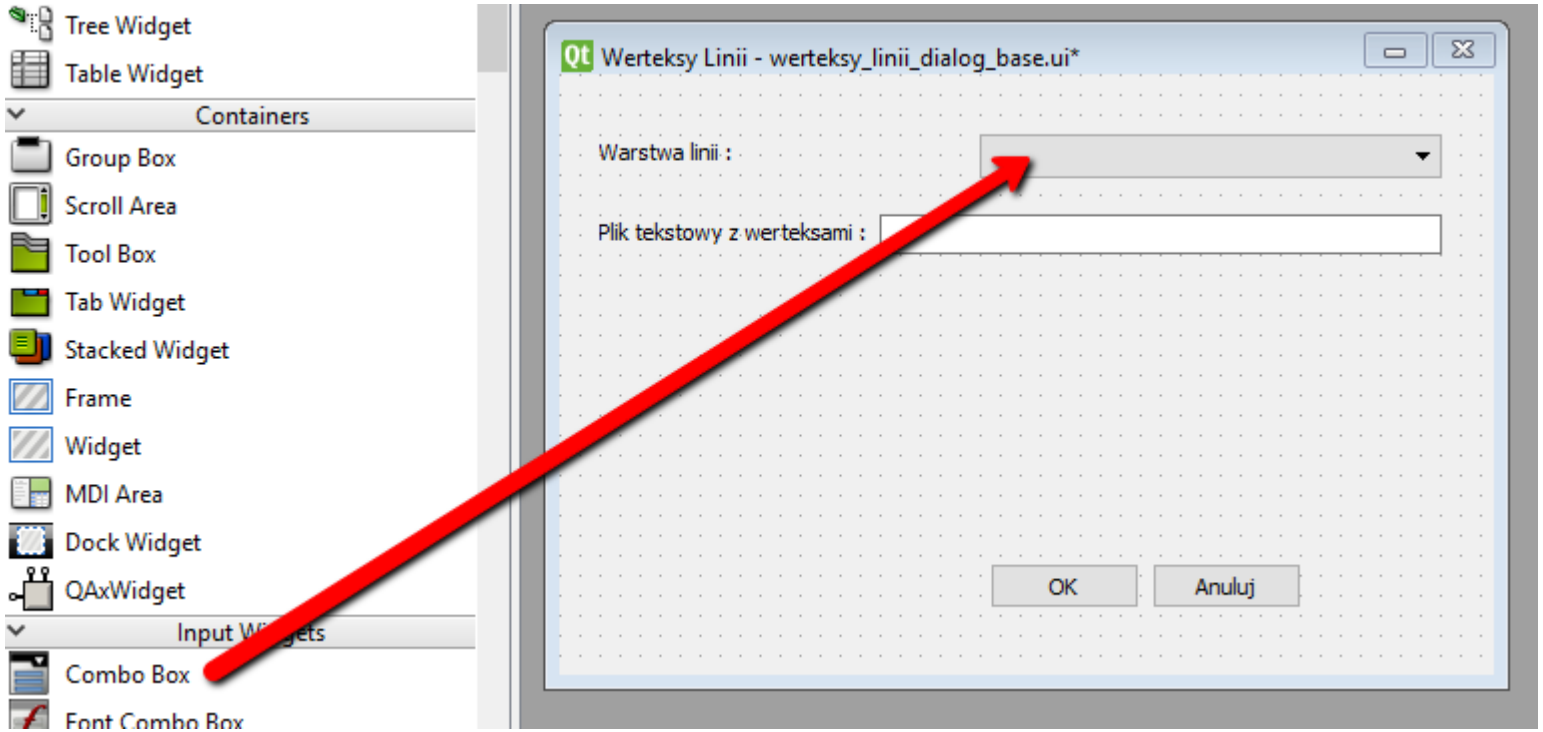
Wprowadzanie stringu




```
# parametry
#ff=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\dane5\Linie.shp"

#p_text=r"C:\JACEK2\QGISHEL18\Hel18\dzien5\dane5\xy_werteksy.txt"
ff=self.dlg.nazwa_warstwy.text()
p_text=self.dlg.nazwa_tekstu.text()
```

GUI 2

Wybór warstwy z dostępnych



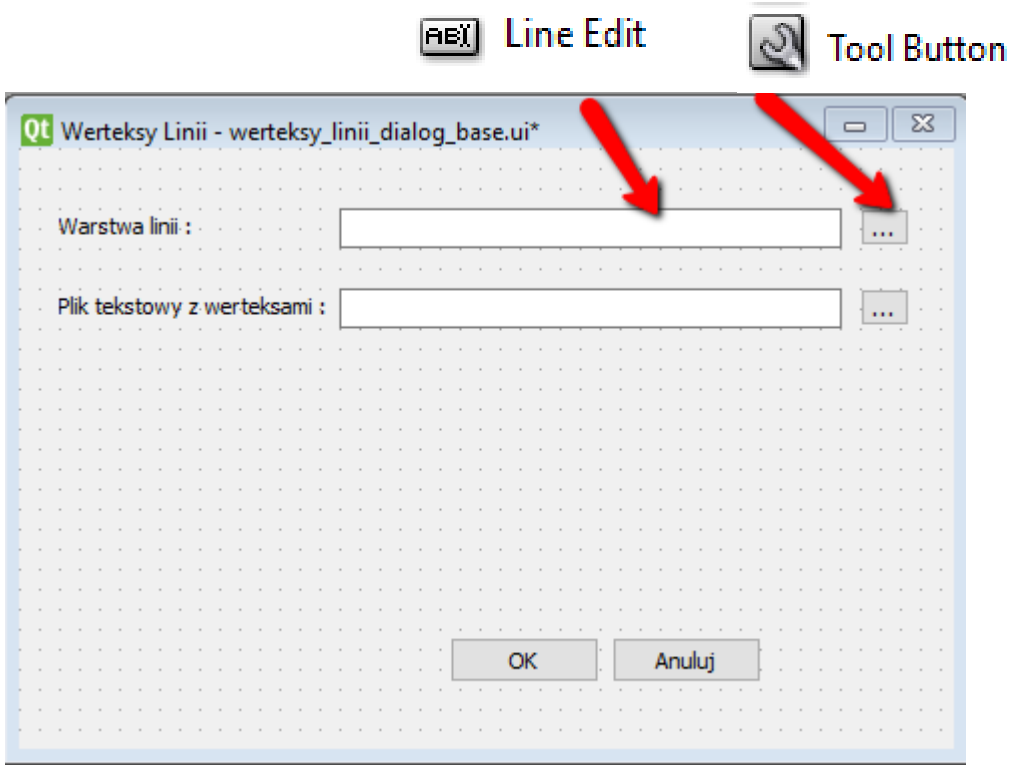
comboBox	QComboBox
label_tekst	QLabel
label_warstwy	QLabel
nazwa_tekstu	QLineEdit

```
from PyQt5.QtCore import QSettings, QTranslator, qVersion, QCoreApplication
from PyQt5.QtGui import QIcon
from PyQt5.QtWidgets import QAction
from qgis.core import *
```

```
def run(self):
    """Run method that performs all the real work"""
    # wypelnienie Combo Box
    layers=QgsProject.instance().mapLayers().values()
    layer_listV=[]
    layer_listVn=[]
    self.dlg.comboBox.clear()
    for layer in layers:
        if layer.type()==0: # vector
            layer_listV.append(layer.source())
            layer_listVn.append(layer.name())
    self.dlg.comboBox.addItems(layer_listVn)
    # show the dialog
    self.dlg.show()
    # Run the dialog event loop
    result = self.dlg.exec_()
    # See if OK was pressed

    if result:
        # Do something useful here - delete the line containing pass and
        # substitute with your code.
        # ComboBox
        selLayerIndexV=self.dlg.comboBox.currentIndex()
        ff=layer_listV[selLayerIndexV]
```

GUI 3

Input and Output dialog – okno czytania i zapisywania warstwy otwierane za pomocą Tool Button



label_tekst	QLabel
label_warstwy	QLabel
nazwa_tekstu	QLineEdit
nazwa_warstwy	QLineEdit
toolButton_in	QToolButton
toolButton_out	QToolButton

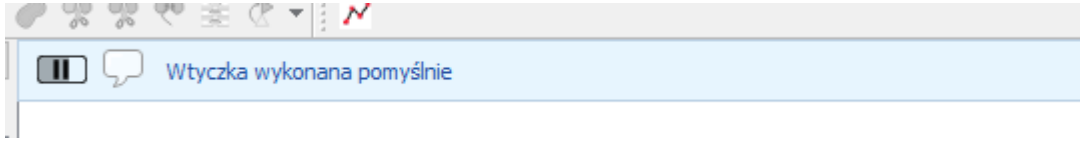
```
24  from PyQt5.QtCore import QSettings, QTranslator, qVersion, QCoreApplication
25  from PyQt5.QtGui import QIcon
26  from PyQt5.QtWidgets import QAction, QFileDialog
27  from qgis.core import *
28  # Initialize Qt resources from file resources.py
29  from .resources import *
30  # Import the code for the dialog
31  from .werteksy_linii_dialog import WerteksyLiniiDialog
32  import os.path
33  from osgeo import ogr
34
35  ppath=''
```

```
76          self.dlg.nazwa_warstwy.clear()
77          self.dlg.toolButton_in.clicked.connect(self.select_open_file)
78          self.dlg.nazwa_tekstu.clear()
79          self.dlg.toolButton_out.clicked.connect(self.select_output_file)
80      # noinspection PyMethodMayBeStatic
81      def tr(self, message):
82          """Get the translation for a string using Qt translation API.
83
```

```
189
190     def select_open_file(self):
191         global ppath
192         filenamet=QFileDialog.getOpenFileName(self.dlg,"Wczytaj warstwe linii",ppath,'*.shp')
193         filename=str(filenamet[0])
194         if filename!='':
195             ppath=os.path.dirname(filename)
196         self.dlg.nazwa_warstwy.setText(filename)
197
198     def select_output_file(self):
199         global ppath
200         filenamet2=QFileDialog.getSaveFileName(self.dlg,"Wynikowy plik tekstowy",ppath,'*.txt')
201         filename2=str(filenamet2[0])
202         if filename2!='':
203             ppath=os.path.dirname(filename2)
204         self.dlg.nazwa_tekstu.setText(filename2)
205
206     def run(self):
207         """Run method that performs all the real work"""
```

```
215         if result:
216             # Do something useful here - delete the line containing pass and
217             # substitute with your code.
218
219             driver=ogr.GetDriverByName('ESRI shapefile')
220             # parametry
221
222             ff=self.dlg.nazwa_warstwy.text()
223             p_text=self.dlg.nazwa_tekstu.text()
224
225             dataSource=driver.Open(ff,0)  # czytanie i pisanie
```

GUI 4



Messages
+
Pozostanie
wtyczki

```
24  from PyQt5.QtCore import QSettings, QTranslator, qVersion, QCoreApplication
25  from PyQt5.QtGui import QIcon
26  from PyQt5.QtWidgets import QAction, QFileDialog
27  from qgis.core import *
28  from qgis.utils import iface
29  # Initialize Qt resources from file resources.py
30  from .resources import *
31  # Import the code for the dialog
32  from .werteksy_linii_dialog import WerteksyLiniiDialog
33  import os.path
34  from osgeo import ogr
35  from qgis.gui import QgsMessageBar
```

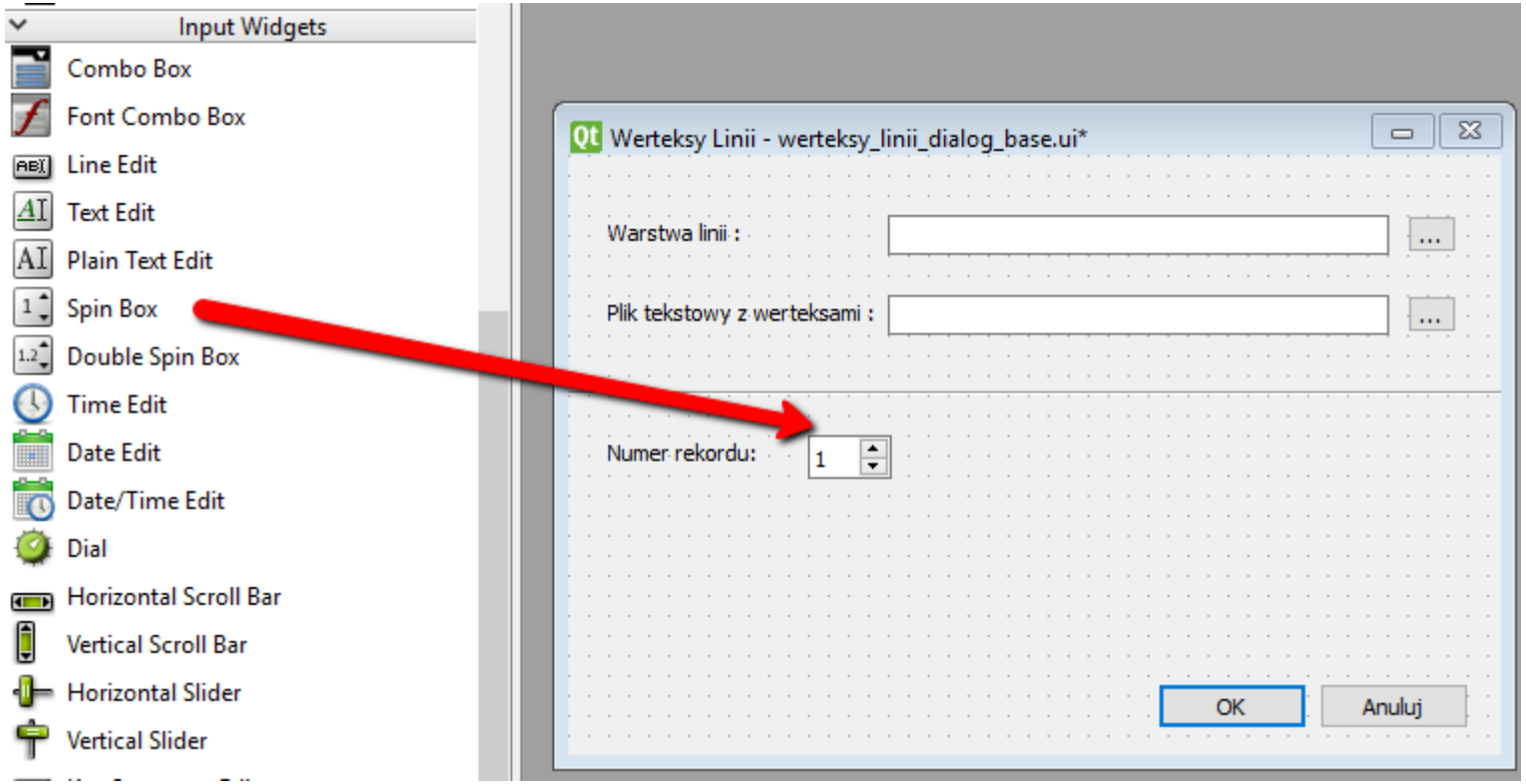
```
250             feature.Destroy()
251             iface.messageBar().pushMessage("Wtyczka wykonana pomyślnie")
252             result=self.dlg.exec_()
253
```

GUI 5

Wprowadzanie liczb całkowitych i rzeczywistych

Spin Box

Double Spin Box



Double Spin Box działa podobnie ale pozwala wybrać liczbę rzeczywistą

button_box	QDialogButtonBox
label_rek	QLabel
label_tekst	QLabel
label_warstwy	QLabel
line	Line
nazwa_tekstu	QLineEdit
nazwa_warstwy	QLineEdit
spinBox	QSpinBox
toolButton_in	QToolButton
toolButton_out	QToolButton

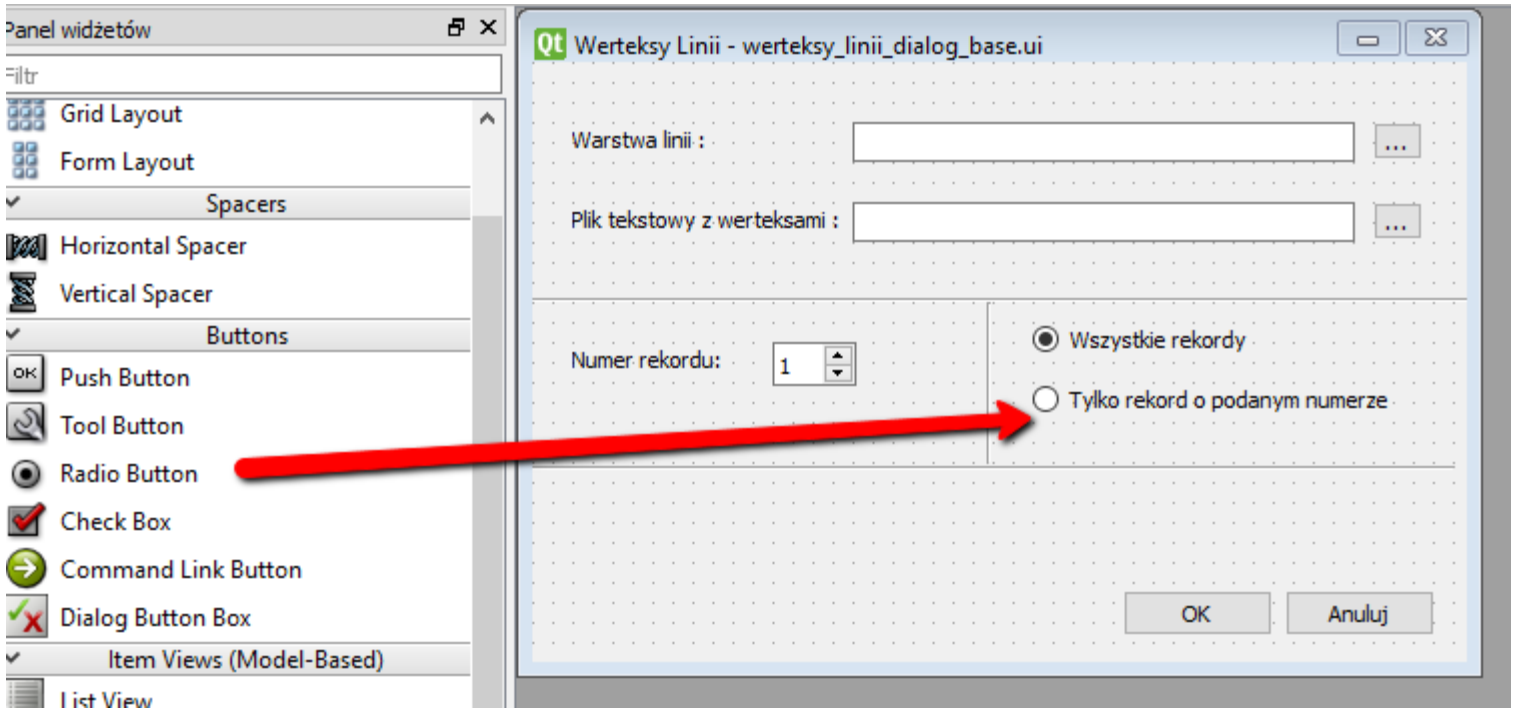
```
ff=self.dlg.nazwa_warstwy.text()
p_text=self.dlg.nazwa_tekstu.text()
nr_rek=self.dlg.spinBox.value()

dataSource=driver.Open(ff,0)  # czytanie i pisanie

vlayer=dataSource.GetLayer()
#print vlayer.GetGeomType()
numFeature=vlayer.GetFeatureCount()
#print numFeature
file=open(p_text,"w")
for i in range (0, numFeature):
    feature=vlayer.GetFeature(i)

    geometry=feature.GetGeometryRef()
    nwtx=geometry.GetPointCount()

    for j in range(0,nwtx):
        x=geometry.GetX(j)
        y=geometry.GetY(j)

        ss= str(i)+','+str( j)+','+str( round(x,2))+','+str(round(y,2))+'\n'
        file.write(ss)
ss='Wybrano rekord : '+str(nr_rek)+'\n'
file.write(ss)
file.close()
feature.Destroy()
iface.messageBar().pushMessage("Wtyczka wykonana pomyślnie")
result=self.dlg.exec_()
```

```
3,4,439475.58,705659.32
3,5,439489.51,705689.59
Wybrano rekord : 2
```

GUI 6

Wprowadzanie opcji za pomocą Radio Button

WerteksyLiniiDialogBase	QDialog
button_box	QDialogButtonBox
label_rek	QLabel
label_tekst	QLabel
label_warstwy	QLabel
line	Line
line_2	Line
line_3	Line
nazwa_tekstu	QLineEdit
nazwa_warstwy	QLineEdit
radioButton_1	QRadioButton
radioButton_2	QRadioButton
spinBox	QSpinBox
toolButton_in	QToolButton
toolButton_out	QToolButton



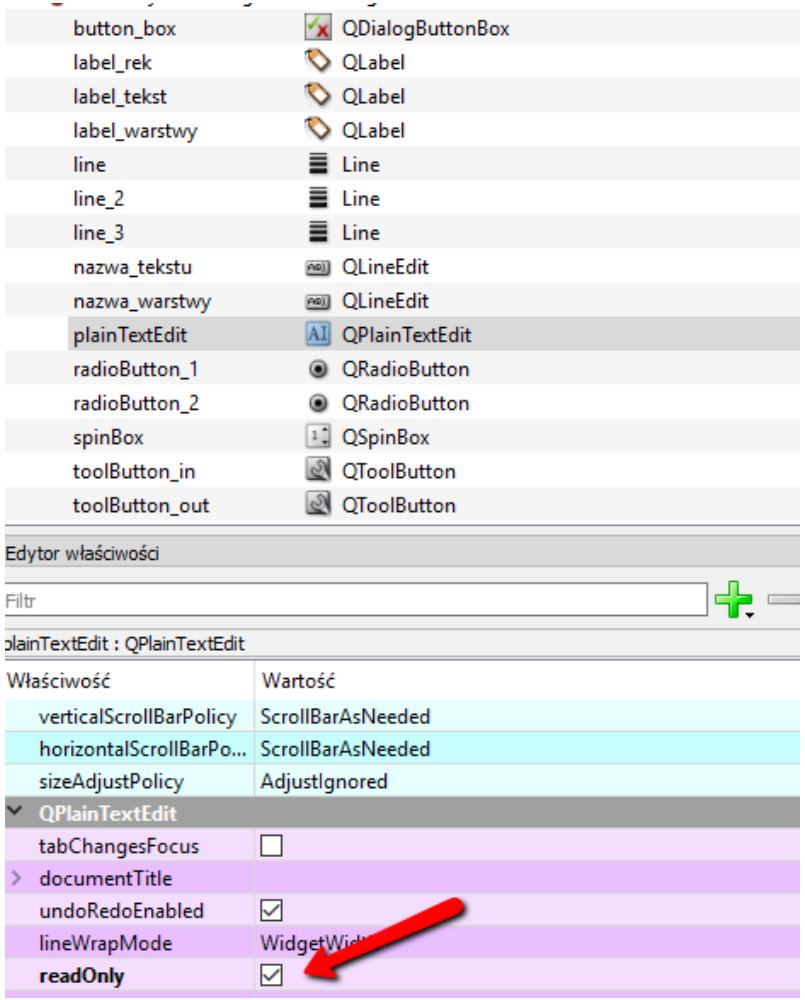
Plik Edycja Format Widok Pomoc

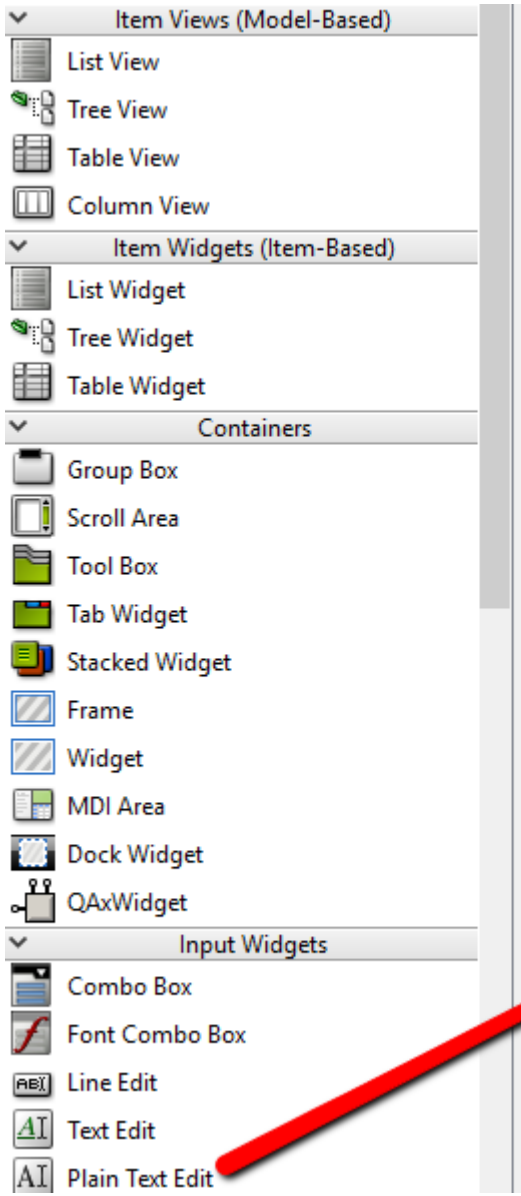
```
1,0,439654.6,706435.05
1,1,439594.92,706410.74
1,2,439590.5,706328.97
1,3,439541.88,706315.71
1,4,439506.52,706267.09
1,5,439504.31,706225.09
1,6,439491.05,706154.37
Wybrano rekord : 2
```

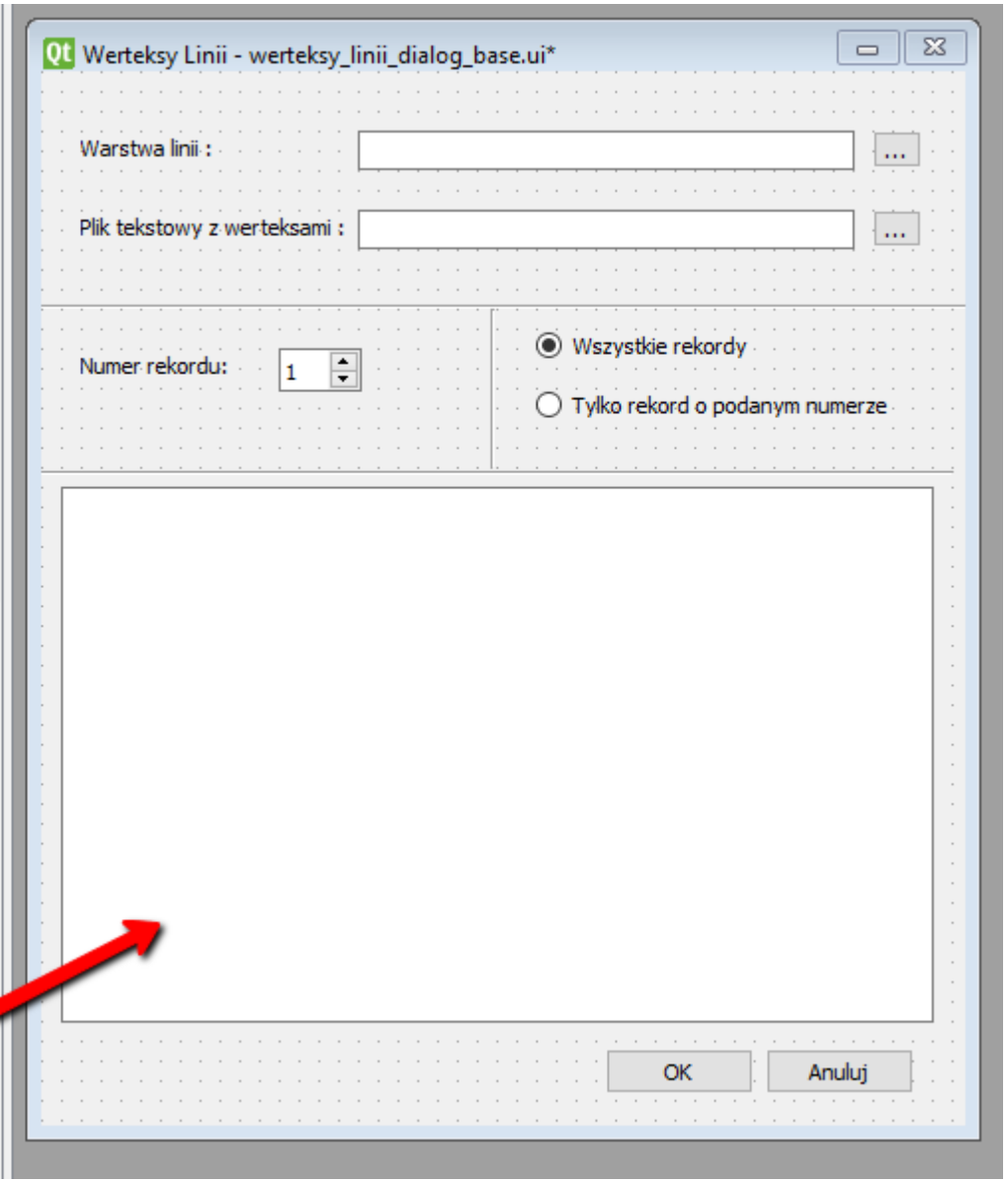
```
227             ff=self.dlg.nazwa_warstwy.text()
228             p_text=self.dlg.nazwa_tekstu.text()
229             nr_rek=self.dlg.spinBox.value()
230
231             B1=self.dlg.radioButton_1.isChecked()
232             B2=self.dlg.radioButton_2.isChecked()
233             dataSource=driver.Open(ff,0)  # czytanie i pisanie
234
239              file=open(p_text,"w")
240             if B1==1:
241                 for i in range (0, numFeature):
242                     feature=vlayer.GetFeature(i)
243
244                     geometry=feature.GetGeometryRef()
245                     nwtx=geometry.GetPointCount()
246
247                     for j in range(0,nwtx):
248                         x=geometry.GetX(j)
249                         y=geometry.GetY(j)
250
251                         ss= str(i)+','+str( j)+','+str( round(x,2))+','+str(round(y,2))+'\n'
252                         file.write(ss)
253             if B2==1:
254                 i=nr_rek-1
255                 feature=vlayer.GetFeature(i)
256
257                 geometry=feature.GetGeometryRef()
258                 nwtx=geometry.GetPointCount()
259
260                 for j in range(0,nwtx):
261                     x=geometry.GetX(j)
262                     y=geometry.GetY(j)
263
264                     ss= str(i)+','+str( j)+','+str( round(x,2))+','+str(round(y,2))+'\n'
265                     file.write(ss)
266             ss='Wybrano rekord : '+str(nr_rek)+'\n'
```

GUI 7

Wyprowadzenie tekstu w GUI







```
215         # See if OK was pressed
216         self.dlg.plainTextEdit.clear()
217         errr=0
218         if result:
```

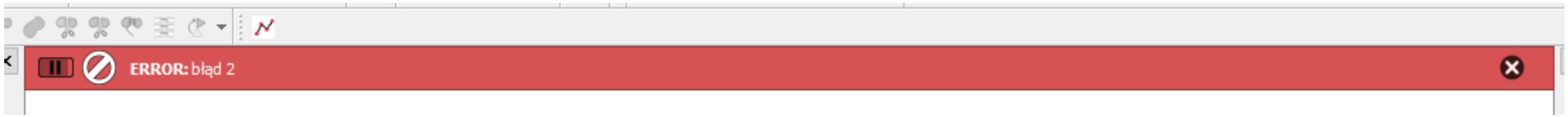
```
ss= str(i)+','+str( j)+','+str( round(x,2))+','+str(round(y,2))+'\n'
self.dlg.plainTextEdit.appendPlainText(ss)
file.write(ss)
```

Po zakończeniu testowania wprowadzamy konstrukcję :

try:

except:

i dodajemy messageBar o błędzie programu (ewentualnie z numerem błędu)

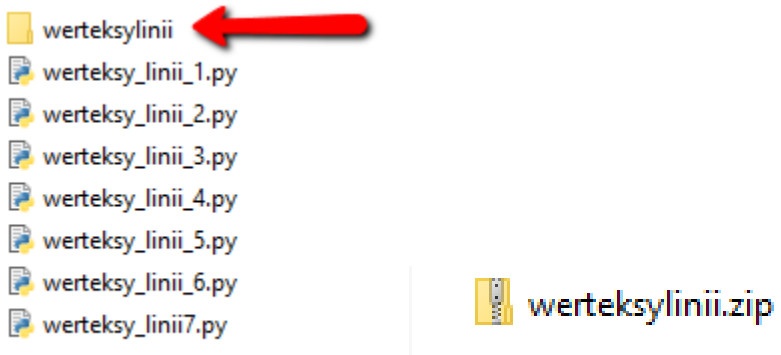


```
215         # See if OK was pressed
216         self.dlg.plainTextEdit.clear()
217         errr=0
218         if result:
219             # Do something useful here - delete the line containing pass and
220             # substitute with your code.
221             try:
222 
223 
224                 driver=ogr.GetDriverByName('ESRI shapefile')
225 
226                 # parametry
227 
228                 ff=self.dlg.nazwa_warstwy.text()
229                 p_text=self.dlg.nazwa_tekstu.text()
230                 nr_rek=self.dlg.spinBox.value()
231                 if ff=="":
232                     errr=1
233                 if p_text=="":
234                     errr=2
235                 B1=self.dlg.radioButton_1.isChecked()
```

```
276                 feature.Destroy()
277                 iface.messageBar().pushMessage("Wtyczka wykonana pomyślnie")
278                 result=self.dlg.exec_()
279             except:
280                 errrb="błąd "+str(errr)
281                 iface.messageBar().pushMessage("ERROR",errrb,level=Qgis.Critical)
```

Rozprowadzanie programu:

1. ZIP katalogu z wtyczką



2. Załadowanie wtyczki