



# Wydział Geodezji i Kartografii

POLITECHNIKA WARSZAWSKA

Projekt nr 1  
Informatyka Geodezyjna  
sem. IV, ćwiczenia, rok akad. 2018-2019

Klaudia Żarnowiecka  
grupa 1, Numer Indeksu: 302634  
Warszawa, 2019

---

## Spis treści

1	Opis zadania	2
2	Obliczenie współrzędnych punktu przecięcia	2
3	Pobieranie danych wejściowych od użytkownika	2
4	Zapis danych do pliku	4
5	Czyszczenia pól zawierających początkowe współrzędne	5
6	Wykres	5
7	Funkcje	6
8	Instrukcja dla użytkownika	6

## 1.Opis zadania

Celem ćwiczenia jest stworzenie programu posiadającego graficzny interfejs użytkownika (GUI), który będzie realizował zadanie znalezienia przecięcia dwóch odcinków.

## 2.Obliczenie współrzędnych punktu przecięcia

Do obliczenia punktu przecięcia wykorzystana została funkcja if. Uwzględnionych zostaje 5 możliwości: przecięcie odcinków, przecięcie na przedłużeniu odcinka AB, przecięcie na przedłużeniu odcinka CD, przecięcie na przedłużeniu obu odcinków, brak punktu P-gdy współrzędne są równoległe

```
Fragment kodu
dXab = Xb - Xa
dYab = Yb - Ya
dXcd = Xd - Xc
dYcd = Yd - Yc
dXac = Xc - Xa
dYac = Yc - Ya
x = dXab*dYcd - dYab*dXcd

if x!= 0:
    t1 = (dXac*dYcd - dYac*dXcd)/x
    t2 = (dXac*dYab - dYac*dXab)/x
    if t1>=0 and t1<=1 and t2>=0 and t2<=1:
        xp = Xa + t1*dXab
        yp = Ya + t1*dYab
        a = "{0:.3f}".format(xp)
        b = "{0:.3f}".format(yp)
        self.xpEdit.setText(str(a))
        self.ypEdit.setText(str(b))
        self.polozenie.setText(str("Punkt P znajduje się na przecięciu odcinków"))
    elif 0 <= t1 <=1:
        xp = Xa + t1*dXab
        yp = Ya + t1*dYab
        c = "{0:.3f}".format(xp)
        d = "{0:.3f}".format(yp)
        self.xpEdit.setText(str(c))
        self.ypEdit.setText(str(d))
        self.polozenie.setText(str("Punkt P znajduje się na przedłużeniu odcinka ab"))
```

## 3.Pobieranie danych wejściowych od użytkownika

Pierwszym etapem ćwiczenia było zaimportowanie potrzebnych bibliotek oraz utworzenie okna dialogowego. Współrzędne punktów są pobierane od użytkownika, następnie program sprawdza czy wpisane wartości są liczbami. Korzystam tu z instrukcji warunkowej if, która pozwala na wykonywanie obliczeń w zależności od tego czy są prawdziwe czy nie. Dodatkowo przy sprawdzaniu prawdziwości podanych wartości pomijamy znak -,wykorzystujemy tu polecenie lstrip.

```
#import bibliotek
from PyQt5.QtWidgets import QWidget, QPushButton, QApplication, QLabel, QLineEdit, QGridLayout,
QColorDialog, QMessageBox
from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
import sys
import matplotlib.pyplot as plt

def checkValues(self, element):
    if element.text().lstrip('-').replace('.', '', 1).isdigit():
        return float(element.text())
    else:
        element.setFocus()
        return None
```

## 4.Zapis danych do pliku

Do zapisu współrzędnych punktów zostaje otwarty nowy plik .txt. Zapisane zostają z dokładnością 3 miejsc po przecinku, poszczególne współrzędne danego punktu oddzielone zostają znakiem |. Natomiast współrzędne kolejnych punktów zostają zapisane w kolejnych wierszach. Dodatkowo w przypadku gdy punkt P nie istnieje w miejscu współrzędnych wyświetla się napis „brak”.

```
plik = open('wyniki.txt','w')
plik.write("|{:^15}|{: ^15}|{: ^15}|\n".format("Nazwa punktu", "X [m]", "Y [m]" ))
plik.write("|{: ^15}|{:15.3f}|{:15.3f}|\n".format("A",Ya, Xa))
plik.write("|{: ^15}|{:15.3f}|{:15.3f}|\n".format("B",Yb, Xb))
plik.write("|{: ^15}|{:15.3f}|{:15.3f}|\n".format("C",Yc, Xc))
plik.write("|{: ^15}|{:15.3f}|{:15.3f}|\n".format("D",Yd, Xd))
if self.xaEdit == None:
    plik.write("|{: ^15.3f}|{: ^15.3f}|{: ^15.3f}|\n".format("P","brak", "brak"))
else:
    plik.write("|{: ^15}|{: ^15.3f}|{: ^15.3f}|\n".format("P",yp,xp))
plik.close()
```

## 5.Czyszczenia pól zawierających początkowe współrzędne

Do wyczyszczenia pól oraz wyczyszczenia informacji o przecięciu została stworzona funkcja połączona z przyciskiem.

```
self.button2=QPushButton('Wyczysc dane', self)
layout.addWidget(self.button2,15, 1, 1, 2)
self.button2.clicked.connect(self.czysc)
```

```
def czysc(self):
    self.xaEdit.clear()
    self.yaEdit.clear()
    self.xbEdit.clear()
    self.ybEdit.clear()
    self.xcEdit.clear()
    self.ycEdit.clear()
    self.xdEdit.clear()
    self.ydEdit.clear()
    self.xpEdit.clear()
    self.ypEdit.clear()
    self.polozenie.clear()
```

## 6.Wykres

Do narysowania wykresu została użyta biblioteka matplotlib. Końce odcinków zostały oznaczone „kółeczkiem”, dodane zostały również etykiety, oraz zmieniony został rodzaj linii w przypadku gdy punkt P leży na przedłużeniu odcinka. Dodatkowo użytkownik może zmieniać kolor odcinka.

```
#wykres przedstawiający odcinki i punkt przecięcia odcinków
self.figure.clear() #czyszczenie pozostawosci
ax = self.figure.add_subplot(111)
ax.plot([yp, Yb], [xp, Xb], 'go:')
ax.plot([Ya, yp], [Xa, xp], 'go:')
ax.plot([yp, Yd], [xp, Xd], 'go:')
ax.plot([Yc, yp], [Xc, xp], 'go:')
ax.plot([Ya, Yb], [Xa, Xb], 'bo-')
ax.plot([Yc, Yd], [Xc, Xd], color=col, marker='o')
ax.plot(yp, xp, color='blue', marker='o')

#dodanie etykiet na wykresie, określenie rozmiaru oraz koloru czcionki dla każdego punktu
ax.text(Ya, Xa, " A(" + str(round(Xa,3)) + ";" + str(round(Ya,3)) + ")", fontsize=9, color="black")
ax.text(Yb, Xb, " B(" + str(round(Xb,3)) + ";" + str(round(Yb,3)) + ")", fontsize=9, color="black")
ax.text(Yc, Xc, " C(" + str(round(Xc,3)) + ";" + str(round(Yc,3)) + ")", fontsize=9, color="black")
ax.text(Yd, Xd, " D(" + str(round(Xd,3)) + ";" + str(round(Yd,3)) + ")", fontsize=9, color="black")
ax.text(yp, xp, " P(" + str(round(xp,3)) + ";" + str(round(yp,3)) + ")", fontsize=9, color="black")
self.canvas.draw()
```

## 7.Funkcje

**W aplikacji wykorzystane zostały poniższe funkcje:**

**checkValues**-funkcja sprawdzająca czy podane wartości są liczbami

**clrChoose**-funkcja pozwalająca zmienić kolor jednego z odcinków

**handleButton**-funkcja odpowiedzialna za działanie przycisku do Obliczania współrzędnych oraz wyświetlanie ich w konkretnym oknie, została podłączona pod inną funkcję

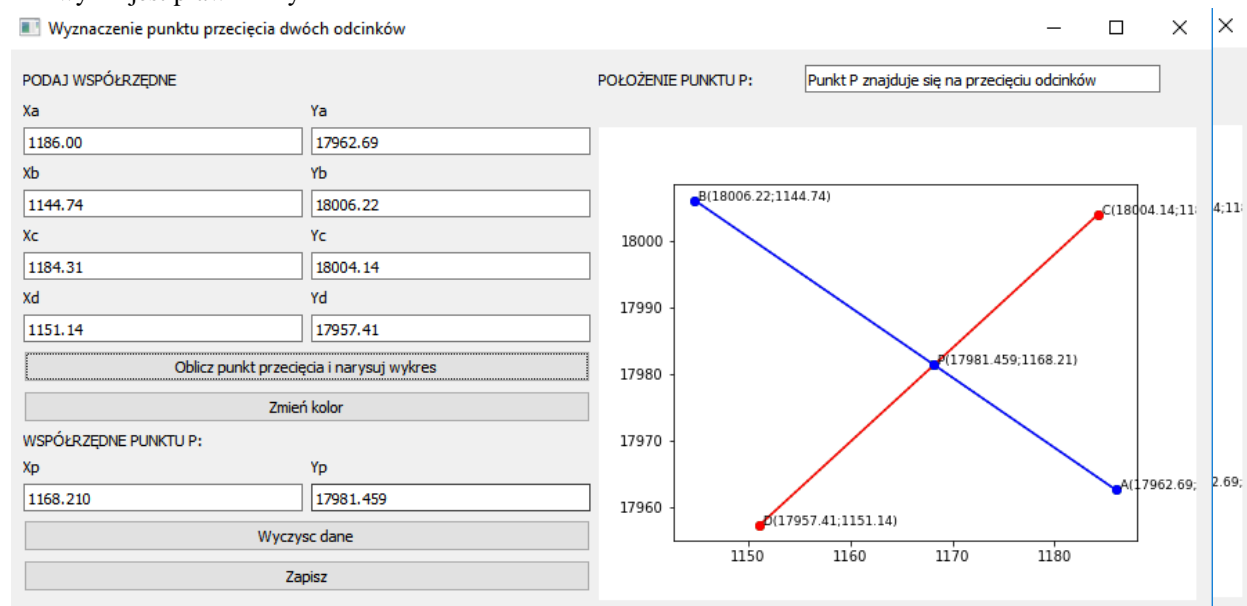
**zapisz**- funkcja zapisująca dane i wyniki do pliku

**czyść**-pozwalająca wyczyścić podane dane

**rysuj**-funkcja odpowiadająca za rysowanie wykresu, w niej wykonywane jest większość obliczeń

## 8.Instrukcja dla użytkownika

Poniżej przedstawiono okno aplikacji w którym zostały wprowadzone dane do przetestowania, jak widać, wynik jest prawidłowy.



Zadaniem użytkownika jest wpisanie współrzędnych punktów A, B, C i D. W przypadku gdy któraś z nich zostanie wprowadzona błędnie na ekranie pojawi się komunikat, który nas o tym poinformuje wtedy użytkownik będzie możliwość poprawy danych. Równocześnie z obliczeniem punktu przecięcia i narysowaniem wykresu program pokaże lokalizację punktu P, następnie użytkownik może zmieniać kolor jednego z odcinków.

W dowolnym momencie użytkownik może wyczyścić dane lub zapisać je do pliku.