

# Algorytmy Zaawansowane - POLE

## Dokumentacja końcowa

Piotr Izert, Łukasz Dragan

1 czerwca 2016

## Spis treści

<b>1</b>	<b>Opis zmian</b>	<b>3</b>
<b>2</b>	<b>Instrukcja użytkownika</b>	<b>3</b>
2.1	Rysowanie . . . . .	3
2.2	Operacje na plikach . . . . .	3
2.3	Obliczanie wyniku . . . . .	3
<b>3</b>	<b>Opis testów</b>	<b>3</b>
<b>4</b>	<b>Podział prac</b>	<b>4</b>
4.1	Łukasz Dragan . . . . .	4
4.2	Piotr Izert . . . . .	4
<b>5</b>	<b>Opis wejścia/wyjścia</b>	<b>4</b>
5.1	Wejście . . . . .	4
5.2	Wyjście . . . . .	4

## 1 Opis zmian

Nie zostały wprowadzone żadne zmiany względem pierwotnych założeń działania algorytmów i programu.

## 2 Instrukcja użytkownika

Dane wejściowe programu mogą zostać wprowadzone poprzez plik tekstowy lub za pomocą interfejsu graficznego.

Interfejs graficzny składa się z części przeznaczonej do rysowania oraz trzech przycisków zwykłych i dwóch przycisków typu „radio”.

### 2.1 Rysowanie

Ekran rysowania ma wymiary 500x500 a środek układu współrzędnych znajduje się w lewym górnym rogu. W celu narysowania wielokąta należy wybrać opcję „polygon” i klikając na ekranie rysowania tworzyć kolejne punkty wielokąta. Podwójne wciśnięcie przycisku myszy kończy rysowanie. W celu narysowania nowego wielokąta należy wykonać opisaną operację ponownie, a stary wielokąt zniknie samoczynnie.

W celu narysowania punktu należy przy wybranej opcji „point” kliknąć na ekran rysowania. Zostanie wybrany punkt. Aby wybrać nowy punkt należy kliknąć w inne miejsce ekranu. Wówczas stary punkt zniknie. Interfejs zawiera również przycisk „Clear” służący do czyszczenia ekranu rysowania i pamięci operacyjnej programu.

### 2.2 Operacje na plikach

Po wybraniu przycisku „read input” program sprawdza, czy w jego lokalizacji istnieje plik „in.txt”. Jeżeli tak, to wczytuje z niego dane i prezentuje w formie graficznej. W przeciwnym wypadku ekran rysowania jest pusty.

### 2.3 Obliczanie wyniku

Przy wprowadzonych danych widocznych w części graficznej możliwe jest dokonanie obliczeń: pola wielokąta, sprawdzenia zawierania punktu w wielokącie oraz sprawdzenia, czy wielokąt jest prosty. W tym celu należy wybrać przycisk „Calculate”. Wyświetlone zostanie wówczas okno z wartościami obliczonych danych oraz dane te zostaną zapisane do pliku „out.txt”.

## 3 Opis testów

Do kodu programu zostały dołączone testy jednostkowe, które pokrywają wszystkie zastosowane algorytmy rozważając różne przypadki testowe. Do dokumentacji końcowej dołączamy również zestaw plików testowych, który znajduje się

w folderze /test. Pliki są w postaci: plik wejściowy „in.txt” i odpowiadający mu plik wyjściowy „out.txt”.

## 4 Podział prac

### 4.1 Łukasz Dragan

- Implementacja interfejsu graficznego i obsługi plików,
- Implementacja i dokumentacja algorytmu liczenia pola wielokąta.

### 4.2 Piotr Izert

- Implementacja i dokumentacja algorytmu sprawdzania, czy wielokąt jest prosty,
- Implementacja i dokumentacja algorytmu sprawdzania, czy punkt znajduje się wewnątrz wielokąta.

## 5 Opis wejścia/wyjścia

### 5.1 Wejście

Program domyślnie jako wejście przyjmuje zawartość pliku „in.txt”, który powinien zawierać w kolejnych liniach:

1. Dane postaci  $x_1 y_1 \dots x_n y_n$ , gdzie  $(x_i, y_i) \in \mathbb{R}^2$  dla  $i = 1, 2, \dots, n$  to współrzędne kolejnych punktów a  $n$  to liczba wierzchołków wielokąta.
2. Dane postaci  $x y$ , gdzie  $(x, y) \in \mathbb{R}^2$  będące współrzędnymi punktu, którego zawieranie w wielokącie ma zostać sprawdzone.

#### Przykładowe wejście

```
344,8 91,2 68,8 121,6 352,8 218,4 448 114,4
288,8 136
```

### 5.2 Wyjście

Rezultat działania programu zapisywany jest w pliku „out.txt” w postaci  $S Ans$  gdzie  $S$  to pole powierzchni wielokąta a  $Ans \in \{„TAK”, „NIE”\}$  to odpowiedź na pytanie, czy dany punkt jest zawarty w wielokącie. W przypadku, gdy dany wielokąt nie jest prosty, rezultatem działania programu jest NOT SIMPLE. Jeżeli dane podane na wejściu są niepoprawne, program zapisze do pliku BAD INPUT.

#### Przykładowe wyjście

```
1243,33 TAK
```