# Projekt: Zbiór punktów przecięcia odcinków oraz Otoczka wypukła

# Skład zespołu nr 5,

Bartosz Heliński

180748,

# Lista wykonanych zadań w projekcie

*Zarządzanie zespołem, projektowanie i tworzenie gui.*

Kamil Nabożny

180772,

# Lista wykonanych zadań w projekcie

*Dodawanie nowych funkcjonalności, testowanie aplikacji.*

Przemek Bednarz

180732,

# Lista wykonanych zadań w projekcie

*Implementacja algorytmu punktów przecięcia odcinków, naprawianie błędów.*

Dawid Flis

180742,

# Lista wykonanych zadań w projekcie

*Implementacja algorytmu otoczki wypukłej, naprawianie błędów.*

# Manual

## Zadanie, które program ma realizować

1. Program wyznacza przecięcie dwóch odcinków na płaszczyźnie:

* Informuje, czy odcinki przecinają się (TAK/NIE).
* Jeżeli odcinki przecinają się, to program określa zbiór, na którym następuje przecięcie:
  + punkt i jego współrzędne, albo
  + odcinek i współrzędne jego końców.

Dane wejściowe: współrzędne końców dwóch odcinków.

2. Program wyznacza otoczkę wypukłą czterech punktów na płaszczyźnie:

* informuje, jakim zborem jest otoczka wypukła (czworokąt, trójkąt, odcinek, punkt...),
* program wypisuje współrzędne kolejnych wierzchołków otoczki wypukłej.

Dane wejściowe: współrzędne punktów na płaszczyźnie.

## Lista opcji do wyboru z krótkim opisem każdej z nich

**Menu główne:**

**Nowy projekt** – utwórz nowy projekt:

* **Przecięcie odcinków** – podaj 4 punkty (P1–P4), aby sprawdzić przecięcie dwóch odcinków.
* **Otoczka wypukła** – podaj punkty (min. 1), aby wyznaczyć otoczkę wypukłą.

**Wczytaj projekt** – załaduj zapisane dane z pliku CSV.

**Opcje** – zmień język, motyw (jasny/ciemny) oraz rozmiar okna.

**Informacje** – wyświetla informacje o aplikacji.

**Zakończ** – zamyka program.

**Przecięcie odcinków:**

* Pola: x/y dla punktów P1–P4
* Dozwolone wartości: liczby zmiennoprzecinkowe
* Obsługa błędów: brak danych lub tekst → komunikat
* Interaktywność: wybór punktu, klikanie i przesuwanie wykresu, wynik graficzny i tekstowy
* Możliwość wygenerowania wyniku, zapisu do pliku CSV, wyczyszczenia widoku oraz powrotu do menu

**Otoczka wypukła:**

1. Pola: x/y dla dowolnej liczby punktów (min. 1)

* Dozwolone wartości: liczby zmiennoprzecinkowe
* Obsługa błędów: puste/niepoprawne pola → komunikat
* Interaktywność: dodawanie punktów, klikanie i przesuwanie wykresu, obrys otoczki i komunikat
* Możliwość wygenerowania wyniku, zapisu do pliku CSV, wyczyszczenia widoku oraz powrotu do menu

## Niezgodności z założeniami przekazanymi w treści zadania

W ramach realizacji zadania wykonano obydwa **projekty** zamiast jednego, jak przewidywała treść zadania. Zamiast wyboru tylko jednej opcji (przecięcie odcinków lub otoczka wypukła), zdecydowano się zaimplementować **oba tryby**. Dzięki temu aplikacja oferuje użytkownikowi możliwość skorzystania z obu narzędzi geometrycznych w jednym spójnym interfejsie.

# Opis kodu

## Lista plików z kodem źródłowym wchodzących w skład programu

main.py - plik startowy uruchamiający aplikację

gui.py - interfejs graficzny aplikacji (Tkinter)

intersection.py - logika obliczania przecięcia odcinków

convex\_hull.py - logika wyznaczania otoczki wypukłej

translations.py - obsługa tłumaczeń interfejsu (PL/EN)

## Schemat algorytmu

**Przecięcie odcinków:**

Opis algorytmu

Wykorzystano klasyczny **Orientation Test (CCW Test)** oraz **iloczyn wektorowy**, aby sprawdzić przecięcie dwóch odcinków.

Schemat działania

1. **Sprawdź orientację punktów**:
   * + Dla każdej z czterech kombinacji trzech punktów oblicz orientację.
2. **Zastosuj reguły przecięcia**:
   * + Jeśli orientacje są różne → odcinki się przecinają (przypadek ogólny)
     + Jeśli punkty współliniowe → sprawdź, czy leżą na odcinku (przypadki szczególne)
3. **Wyznacz punkt przecięcia (jeśli istnieje)**:
   * + Użyj wzorów parametrycznych dla równań prostych
4. **Zwróć wynik (komunikat i punkt/przedział)**

**Otoczka wypukła (Convex Hull):**

Opis algorytmu

Zastosowano algorytm **Andrew's Monotone Chain** (wariant algorytmu Grahama) – efektywna metoda wyznaczania otoczki wypukłej punktów na płaszczyźnie w czasie O(n log n).

**Schemat działania**

1. **Usuń duplikaty punktów**
2. **Obsłuż przypadki szczególne**:
   * Brak punktów → komunikat
   * Jeden punkt → to właśnie otoczka
   * Dwa punkty lub wszystkie punkty współliniowe → odcinek
3. **Sortuj punkty leksykograficznie (x, potem y)**
4. **Zbuduj dolną część otoczki**:
   * Dla każdego punktu:
     + Jeśli tworzy skręt w prawo z dwoma poprzednimi, usuń ostatni punkt z listy
5. **Zbuduj górną część otoczki**:
   * Podobnie jak dolna, ale w odwrotnej kolejności
6. **Połącz górną i dolną część**
7. **Zwróć otoczkę oraz opis tekstowy**