

### Laboratoria 3. Polimorfizm i dziedziczenie

---

Polimorfizm prowadzi do różnego zachowania się obiektów dziedziczących po pewnym obiekcie w zależności od faktycznej ich klasy. Pozwala to uniknąć wielokrotnego stosowania instrukcji warunkowych, dzięki przerzuceniu ciężaru wyboru odpowiedniej metody, dopasowanej do klasy obiektu na program.

---

#### Zadanie 1.

Zdefiniuj klasę o nazwie `Figura`, oraz dziedziczące po niej klasy `Prostokąt`, `Trojkat`, `Elipsa`, a także dziedziczącą po klasie `Elipsa`, klasę `Kolo` i dziedziczącą po klasie `Prostokąt`, klasę `Kwadrat`. Obiekty tych klas powinny przechowywać informacje o przechowywanych figurach. Poza podstawowymi dla tych klas metodami, zdefiniuj dla nich także 2 metody wirtualne:

1) `void Drukuj()`

Drukującą informację o obiekcie (rodzaj figury, długości boków/promienia/półosi)

2) `double ObliczPole();`

Zwracającą pole figury przechowywanej w obiekcie

Napisz program pobierający od użytkownika następujące dane:

- 1) Liczba całkowita ( $n$ ), której wartość będzie równa podanej przez użytkownika liczbie stworzonych figur.
- 2) W  $n$ -krotnej pętli program pobiera od użytkownika rodzaj wybranej figury, oraz długości jej boków/promienia/półosi.

Program zwraca na ekran informacje o wybranych figurach, oraz ich pola.

#### **Uwaga:**

W programie należy zastosować dynamiczne tworzenie obiektów oraz skorzystać z polimorfizmu (za pośrednictwem metod `Drukuj()` i `ObliczPole()`), dla ujednoczenia działania programu.