

PRZYKŁAD: W którym miejscu poniższego kodu można zaobserwować polimorfizm?

```
class Figura {
    float f, g;
    Figura (float a, float b) {
        f=a;
        g=b;
    }
    float pole() {
        System.out.println("Niezdefiniowane");
        return 0;
    }
}

class Prostokat extends Figura {
    Prostokat (float a, float b) {
        super(a,b);
    }
    float pole() {
        System.out.println("Prostokat");
        return f*g;
    }
}

class Trojkat extends Figura {
    Trojkat (float a, float b) {
        super(a,b);
    }
    float pole() {
        System.out.println("Trojkat");
        return f*g/2;
    }
}

class Test {
    public static void main(String args[]){
        Figura fig = new Figura(1,4);
        Prostokat p = new Prostokat(5,10);
        Trojkat t = new Trojkat (6,9);
        Figura a;
        a = p; System.out.println(a.pole());
        a = t; System.out.println(a.pole());
        a = fig; System.out.println(a.pole());
    }
}
```

Wstęp: Klasy abstrakcyjne i interfejsy

W oparciu o wykład (pliki na edux.pjwstk.edu.pl) odpowiedz na pytania:

1. Co oznacza i jaką ma budowę metoda deklarowana ze słowem kluczowym **abstract**?
2. Kiedy klasa jest abstrakcyjna?
3. Czy można utworzyć obiekt klasy abstrakcyjnej?
4. Co wymusza zadeklarowanie metody jako abstrakcyjnej? (Co klasa abstrakcyjna wymusza na swoich klasach potomnych?)
5. Czy klasa ostateczna (**final**) może być jednocześnie abstrakcyjna?
6. Jaką klasę nazywamy *interfejsem*?
7. Jak traktowane są pola deklarowane w interfejsie?
8. Kiedy projektować klasy abstrakcyjne a kiedy interfejsy?
9. Czy wiele klas może implementować ten sam interfejs?
10. Czy jedna klasa może implementować wiele interfejsów?

Zadanie 10

1. Otwórz plik **Osoba.java** (plik na edux.pjwstk.edu.pl) i wyjaśnij powód deklaracji dwóch metod jako abstrakcyjnych.
2. Uzupełnij luki w kodzie klasy **Naukowiec** i **Student**.
3. Które z poniższych poleceń są nieprawidłowe?
 - *Naukowiec n1 = new Naukowiec("Kowalski", "Jan", 5, 180);*
 - *Student s1 = new Student("Nowak", "Piotr", 9, 25);*
 - *Osoba o1 = new Osoba("Wiśniewski", "Adam", 7);*
4. Napisz klasę testową, w której zastosujesz kilka wywołań polimorficznych.
5. Zmodyfikuj klasę testową tak, aby referencje do obiektów typu **Naukowiec** i **Student** przechowywane były w tablicy (jakiego będzie typu?).