

# Programowanie obiektowe

Wykład 1: Wprowadzenie do programowania  
obiektowego

Dr hab. inż. Stanisław Deniziak, prof.PK,  
p.101, 410  
[sdeniziak@pk.edu.pl](mailto:sdeniziak@pk.edu.pl)

# Informacje organizacyjne

---

- Wykład 30h
  - Zaliczenie: egzamin
- Laboratorium 30h
  - Mgr inż. Sławomir Bąk
  - Mgr inż. Anna Mroczek

<http://eclipse.elektron.pk.edu.pl/~sdeniziak/doku.php?id=ydaktyka:informatyka:po>

Hasło: pobj2013

# Literatura

---

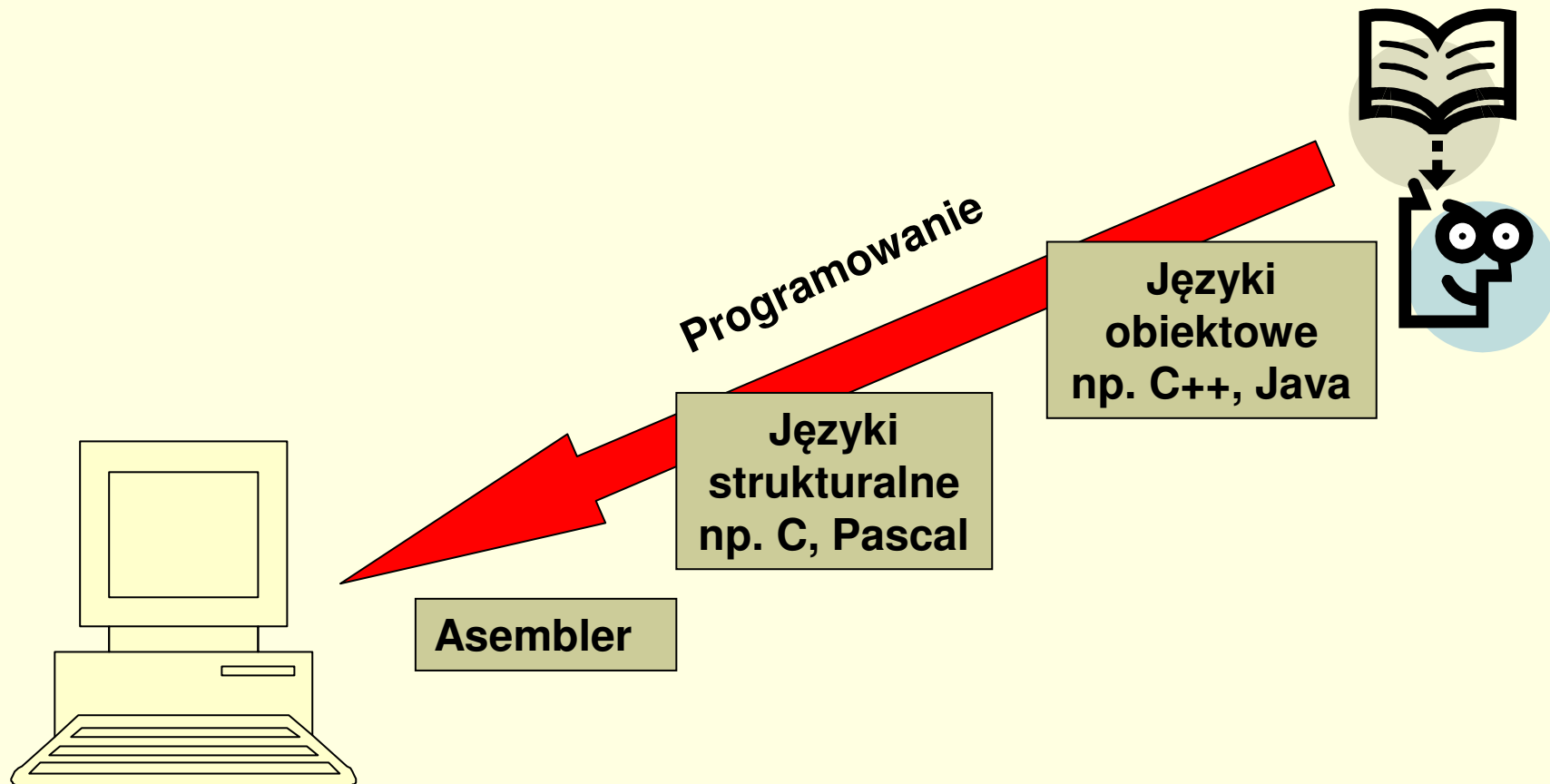
- Bertrand Meyer, *Programowanie zorientowane obiektowo*, Helion, 2005.
- Bruce Eckel, *Thinking in Java – edycja polska*, wydanie. 4, Helion, 2006.
  - [www.bruceeckel.com](http://www.bruceeckel.com) – przykłady, wydania książki w j. ang. (wyd.3, również część edycji 4!)
- Cay S. Horstmann, Gary Cornell, *Java 2. Podstawy*, Helion, 2003.
- <http://www.oracle.com/technetwork/java/index.html>:
  - Java Software Development Kit (J2SE, J2EE, J2ME),
  - Dokumentacje
  - *The Java Tutorial. A practical guide for programmers*

# Paradygmaty programowania obiektowego

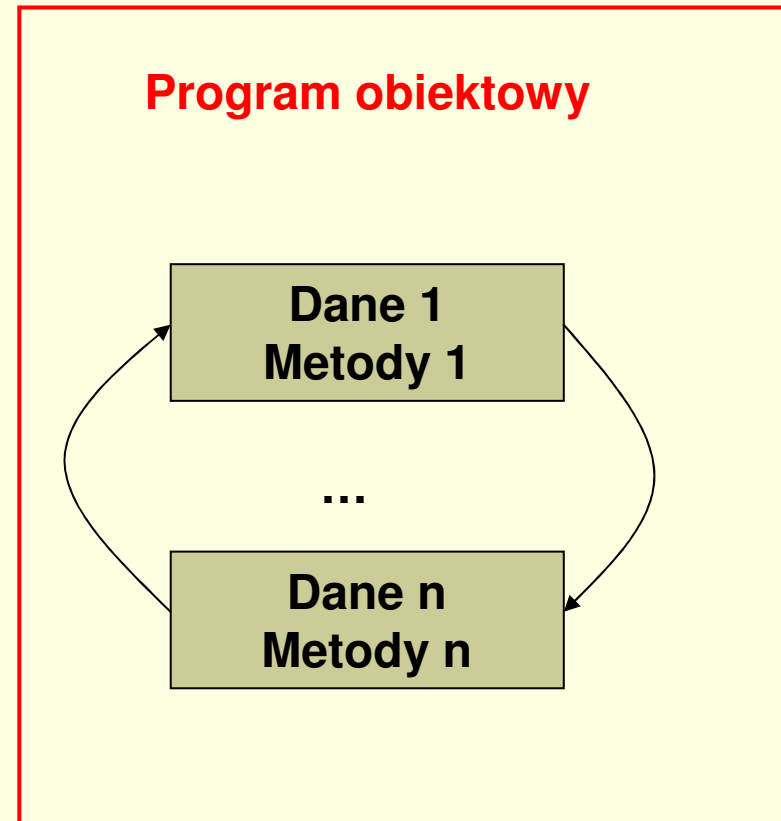
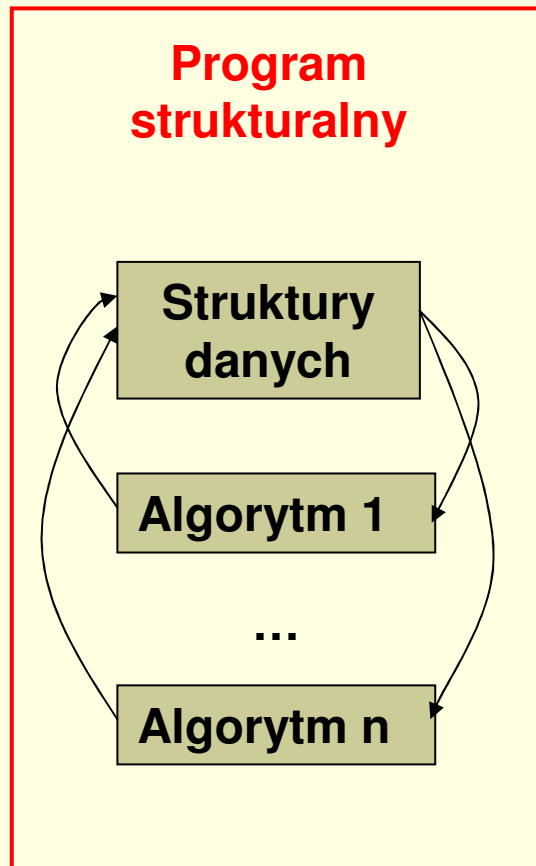
---

- Abstrakcja
- Enkapsulacja
- Dziedziczenie
- Polimorfizm

# Programowanie obiektowe – OOP(Object Oriented Programming)



# Abstrakcja: programowanie obiektowe a programowanie strukturalne



# Enkapsulacja (hermetyzacja, ochrona implementacji)

---

■ **Obiekt** = dane + operacje



■ **Klasa** = „typ” obiektu

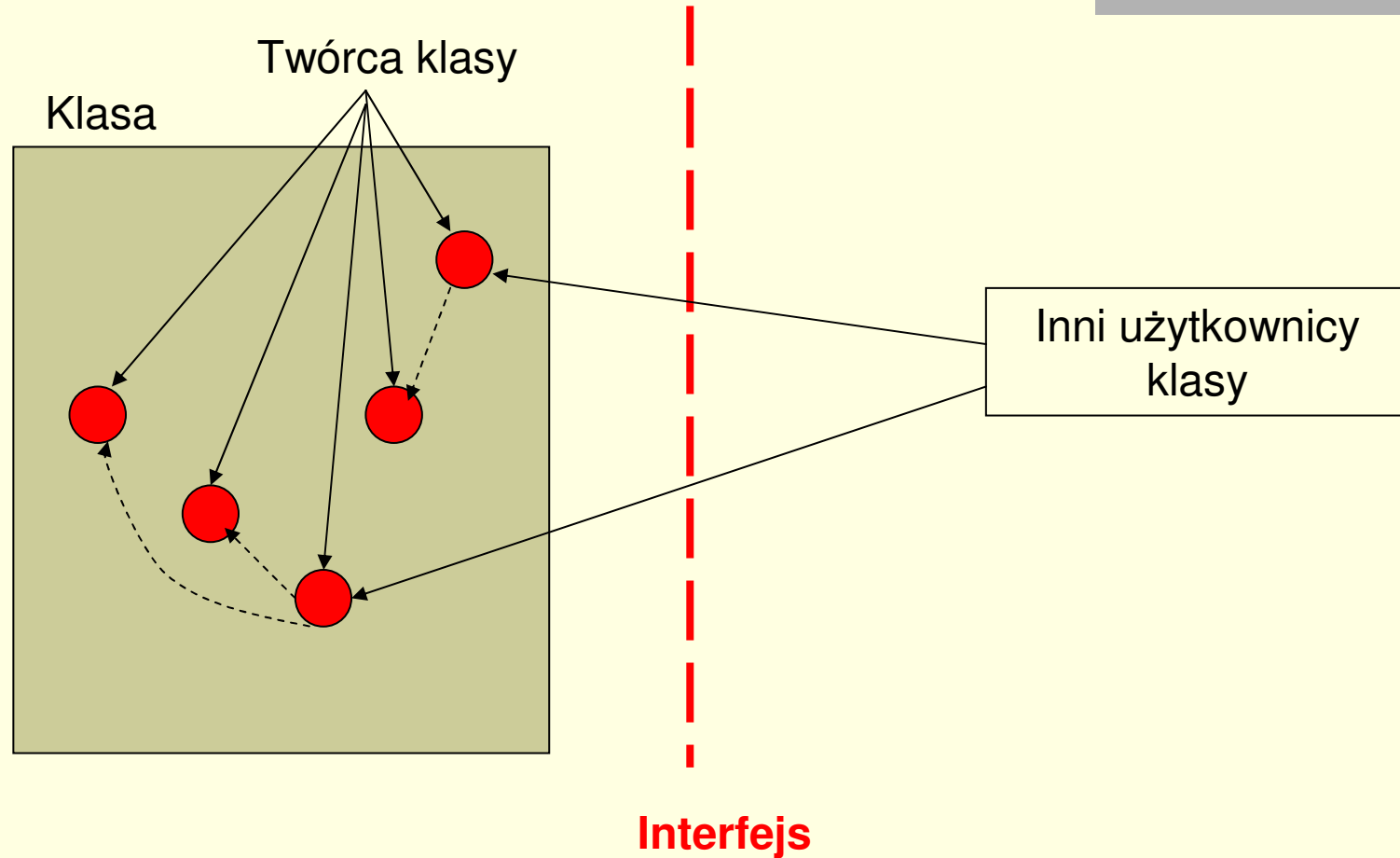
■ **Operacja** = interfejs + implementacja

# Interfejs a implementacja





# Ochrona implementacji



# Wielokrotne wykorzystanie implementacji

---

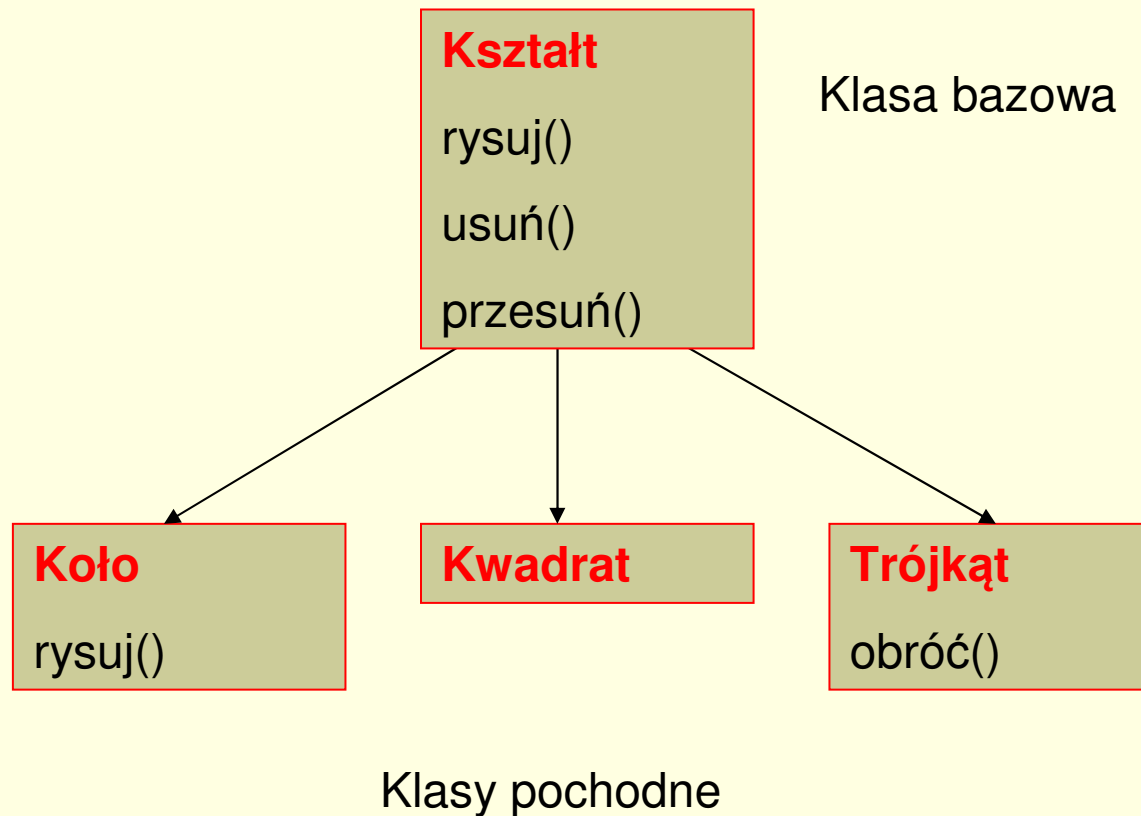
## ■ Składanie

- Wielokrotne wykorzystanie wcześniej napisanego kodu (np. funkcje, deklaracje struktur danych, moduły, klasy itp.)
- Powszechnie stosowane w programowaniu

## ■ Dziedziczenie

- Tylko w programowaniu obiektowym
- Tworzenie nowych klas na bazie klas istniejących
- Klasy pochodne są „kompatybilne w górę” z klasami bazowymi

# Dziedziczenie

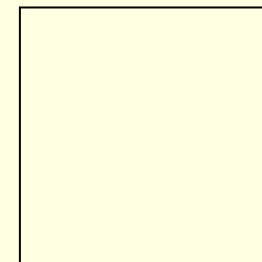
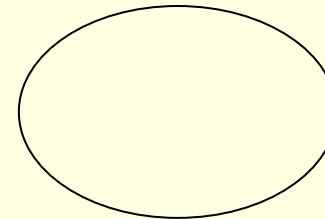


# Polimorfizm

```
Edytor graficzny:  
rysowanie(Kształt k) {  
  ...  
  rysuj (k)  
  ...  
}  
...  
rysowanie(koło)  
...  
rysowanie(kwadrat)
```

*Koło.rysuj()*

*Kwadrat.rysuj() =  
Kształt.rysuj()*



# Trochę historii

---

- **Simula 67** - Ole-Johan Dahl i Kristen Nygaard z *Norsk Regnesentral* w Oslo
  - Pojęcie klasy i obiektu
- **SmallTalk** - Alan Kay, Xerox Palo Alto Research Center
  - dziedziczenie
- **C++** - Bjarne Stroustrup
  - Język wieloparadygmatowy
- **Java** — James Gosling, Sun Microsystems
  - Język „czysto” obiektowy

# Technika obiektowa

---

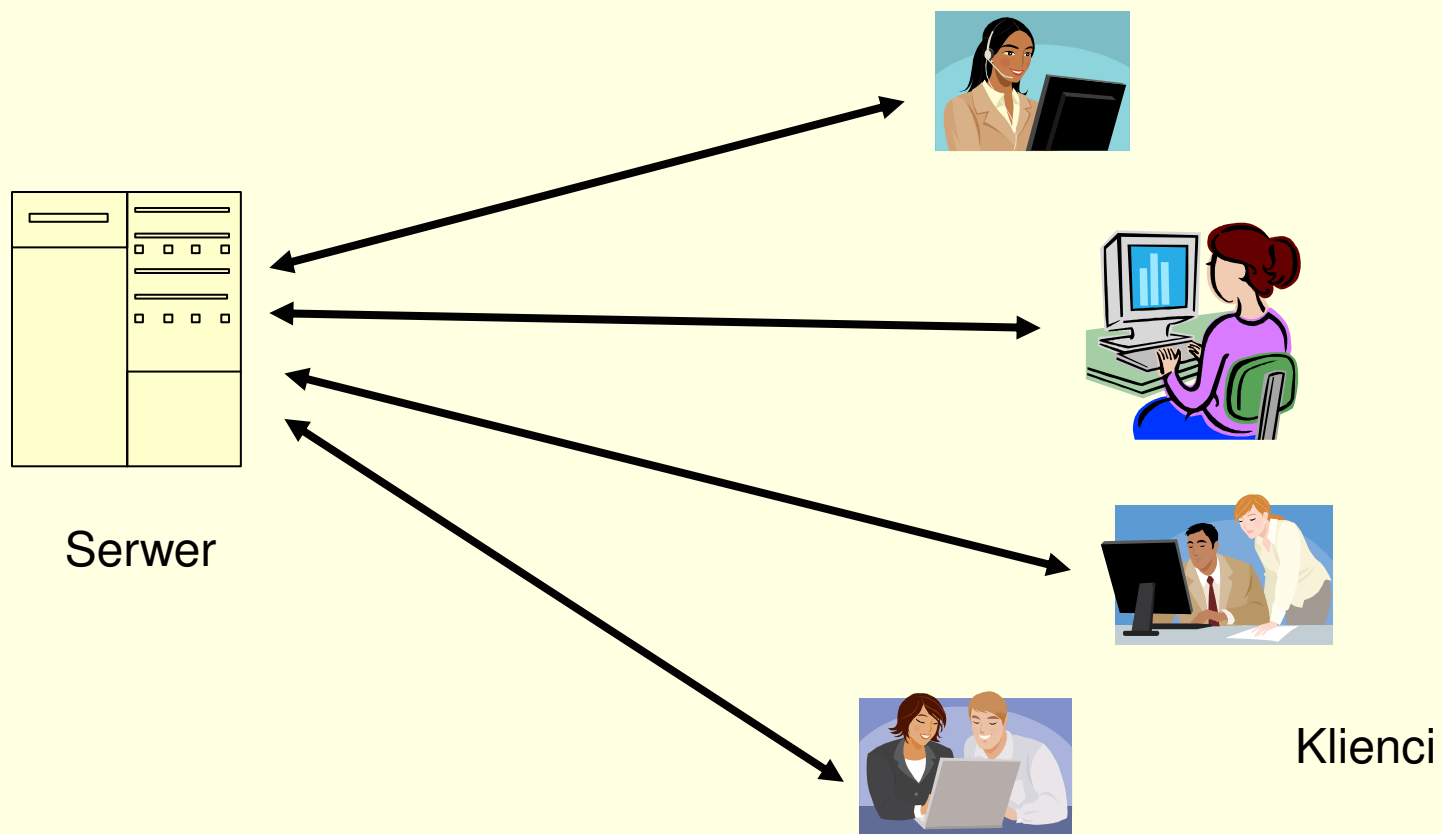
## ■ Analiza

- Badanie wymagań
- Klasyfikacja obiektów w świecie rzeczywistym (obiekty pojęciowe)

## ■ Projektowanie

- Opracowanie rozwiązania spełniającego zadane wymagania
- Projektowanie obiektowe – projektowanie obiektów programowych
- Wyznaczenie odpowiedzialności
- Określenie zasad współpracy pomiędzy obiektami

# Przykład: Czat internetowy (1)



# Funkcje serwera (2)

---

- Obsługa połączeń z aplikacjami klienckimi
- Zarządzanie użytkownikami
  - Logowanie
  - Rejestracja
  - Hasła
  - Baza danych zarejestrowanych użytkowników
  - Baza danych stałych pokoi
- Obsługa komunikatów
  - Wiadomości prywatne
  - Wiadomości publiczne
  - Informacje o statusie użytkowników
- Przetwarzanie poleceń
  - Odebranie polecenia od aplikacji klienckiej
  - Przetworzenie polecenia
  - Wysłanie odpowiedzi



# Funkcje aplikacji klienckiej (3)

---

- Obsługa użytkownika
  - Użytkownicy stali (wymagana rejestracja i hasło)
  - Użytkownicy tymczasowi (dowolna unikalna nazwa)
- Interfejs GUI
  - Okno komunikatów (MessageWindow)
  - Okno poleceń (CommandWindow)
  - Okno z listą użytkowników (UsersWindow)
  - Okno z listą pokoi (RoomWindow)
- Komunikacja z serwerem
  - Wysyłanie komend do serwera
  - Odbiór odpowiedzi
  - Odbiór i przetwarzanie komend z serwera

# Zalety programowania obiektowego

---

- Łatwiejsze tworzenie dużych programów
- Łatwiejsze programowanie zespołowe
- Ułatwia stosowanie techniki programowania przyrostowego
- Łatwiejsze wykorzystanie istniejącego kodu

# Podsumowanie

---

1. Klasa a obiekt
2. Enkapsulacja, hermetyzacja, ukrywanie implementacji
3. Dziedziczenie
4. Polimorfizm
5. Abstrakcja
6. Najistotniejsze różnice programowania obiektowego i programowania strukturalnego/proceduralnego
7. Identyfikacja klas w problemie



**Koniec**