

# Nowoczesne języki programowania obiektowego

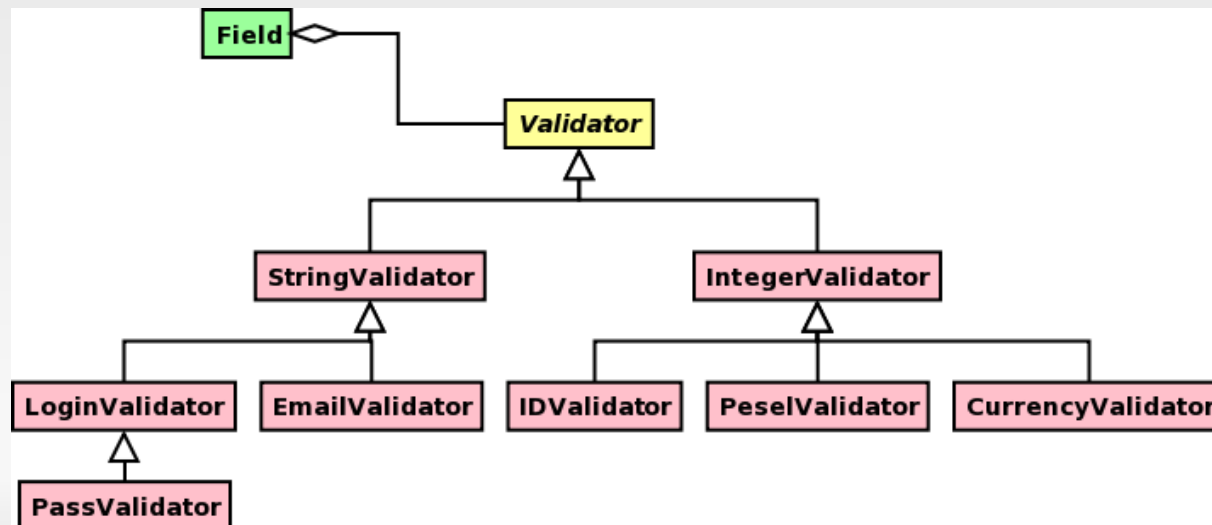
Wzorce projektowe: STRATEGIA

dr Aleksander Lamża

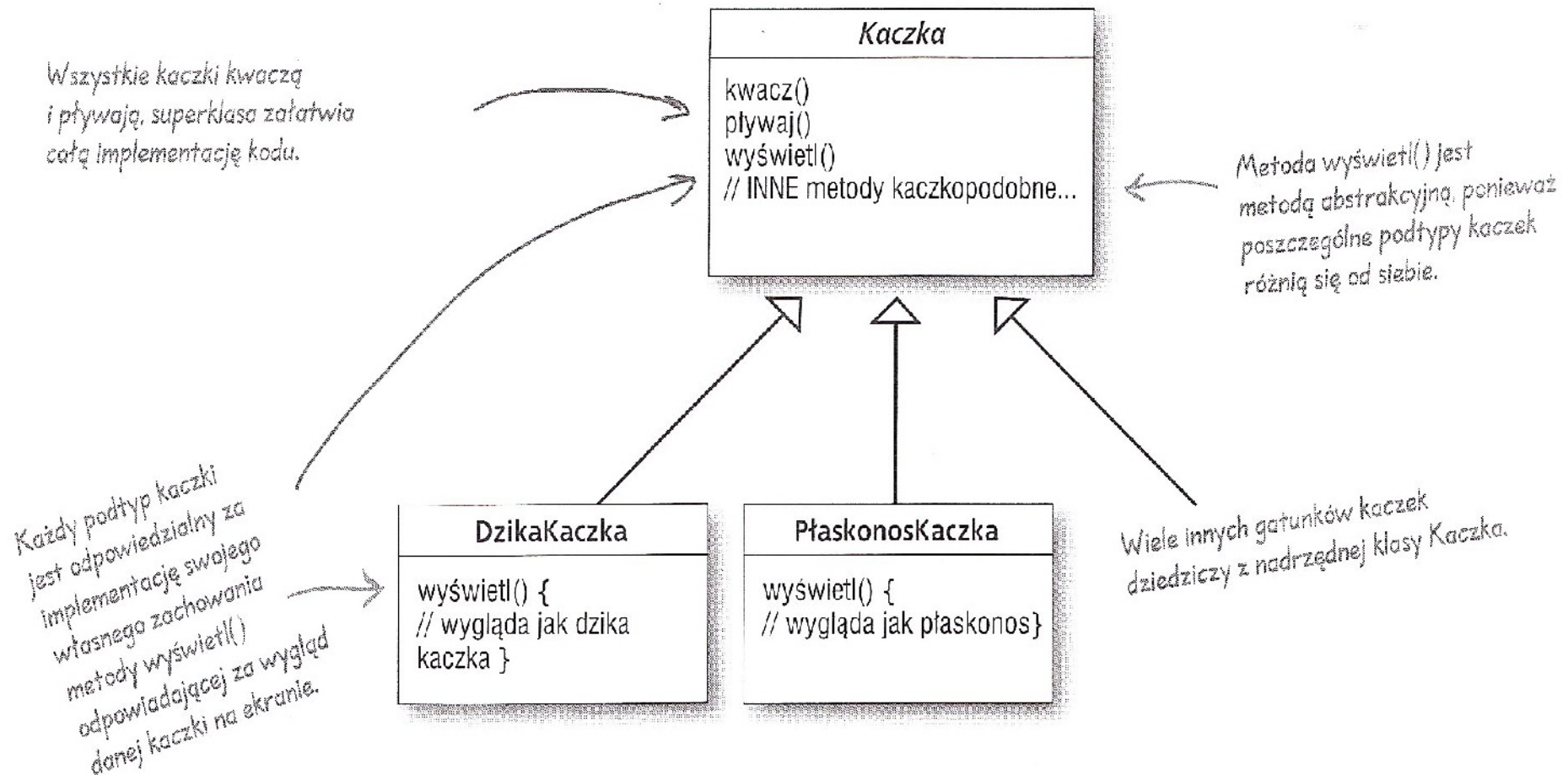
Wydział Informatyki i Nauki o Materiałach  
Instytut Informatyki, Uniwersytet Śląski

# Problem

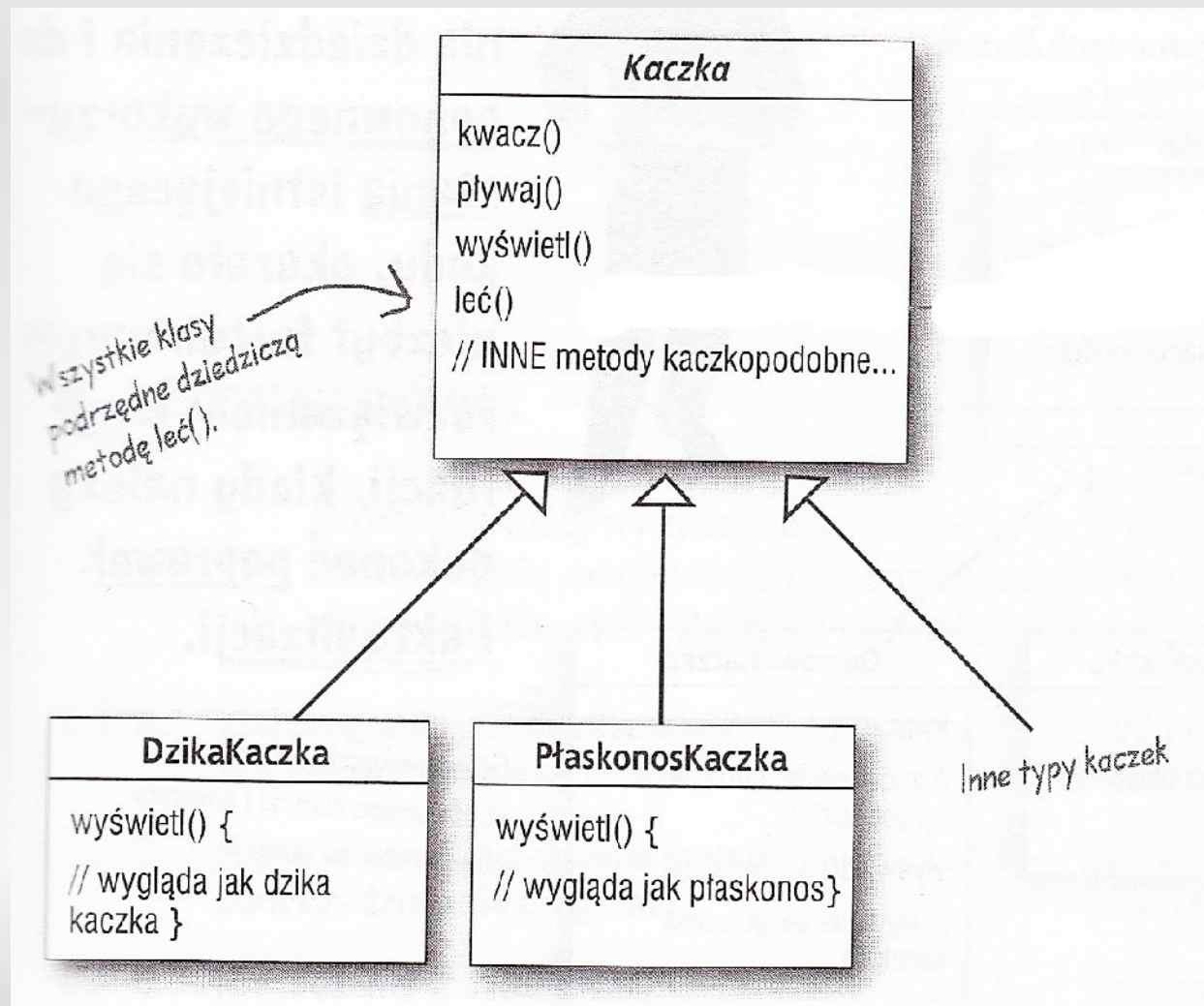
- Potrzeba wybrania jednego z kilku sposobów wykonania częściowego zadania.
- Przykład: walidacja pola w formularzu



# Przykład abstrakcyjny: kaczki

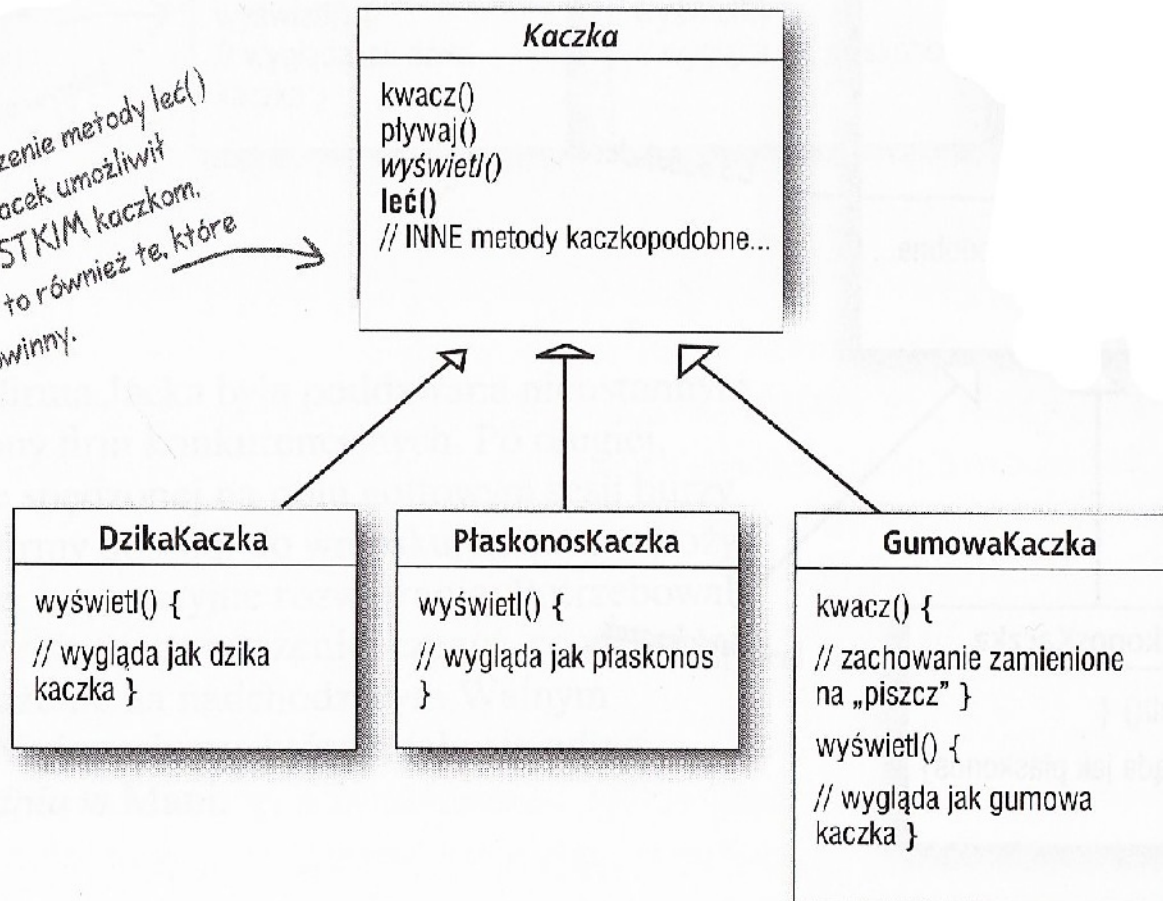


# Teoretycznie dobry pomysł



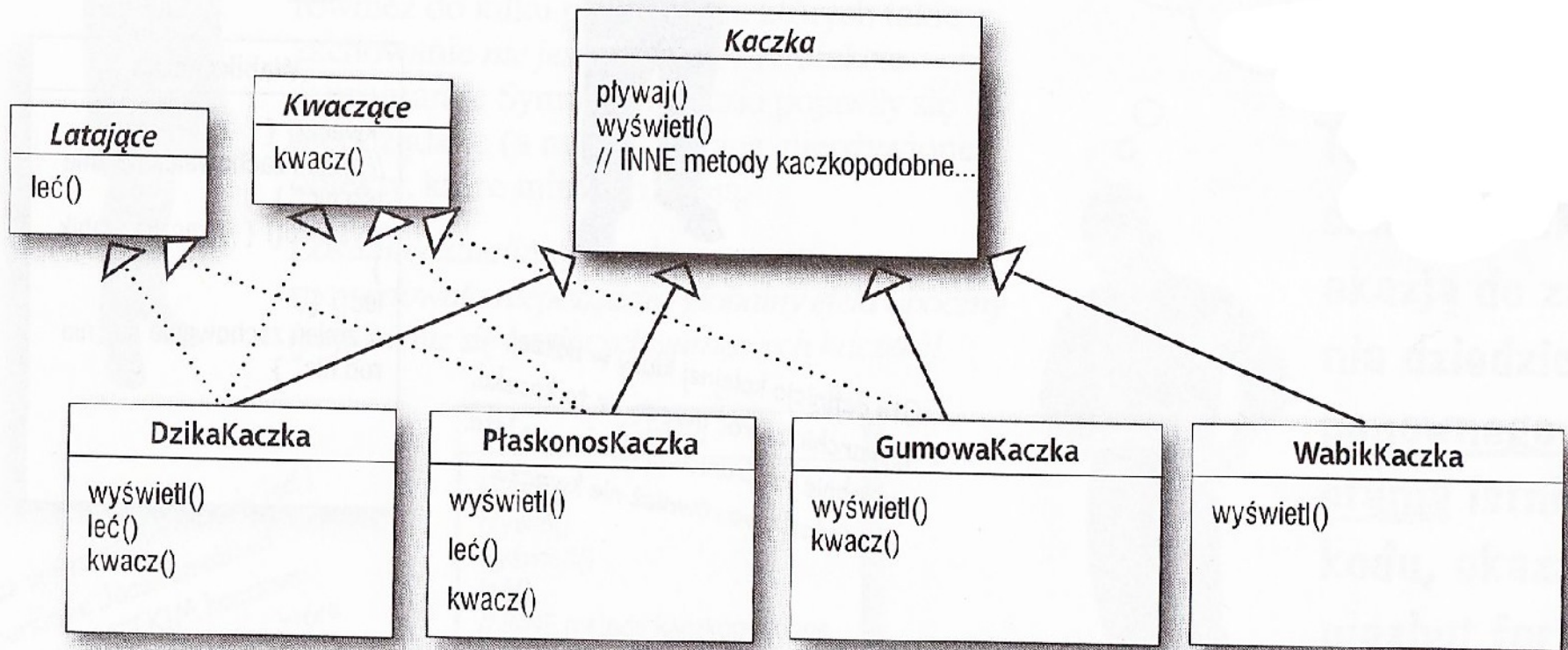
# Problemy z dziedziczeniem

Poprzez umieszczenie metody `leć()` w superklasie Jacek umożliwił latanie **WSZYSTKIM** kaczkom, włączając w to również te, które latać nie powinny.



Gumowe kaczki nie kwaczą, stąd metoda `kwacz()` zmieniła zachowanie na „piszcz”.

# Wprowadzenie interfejsów





# Jedna ze złotych zasad

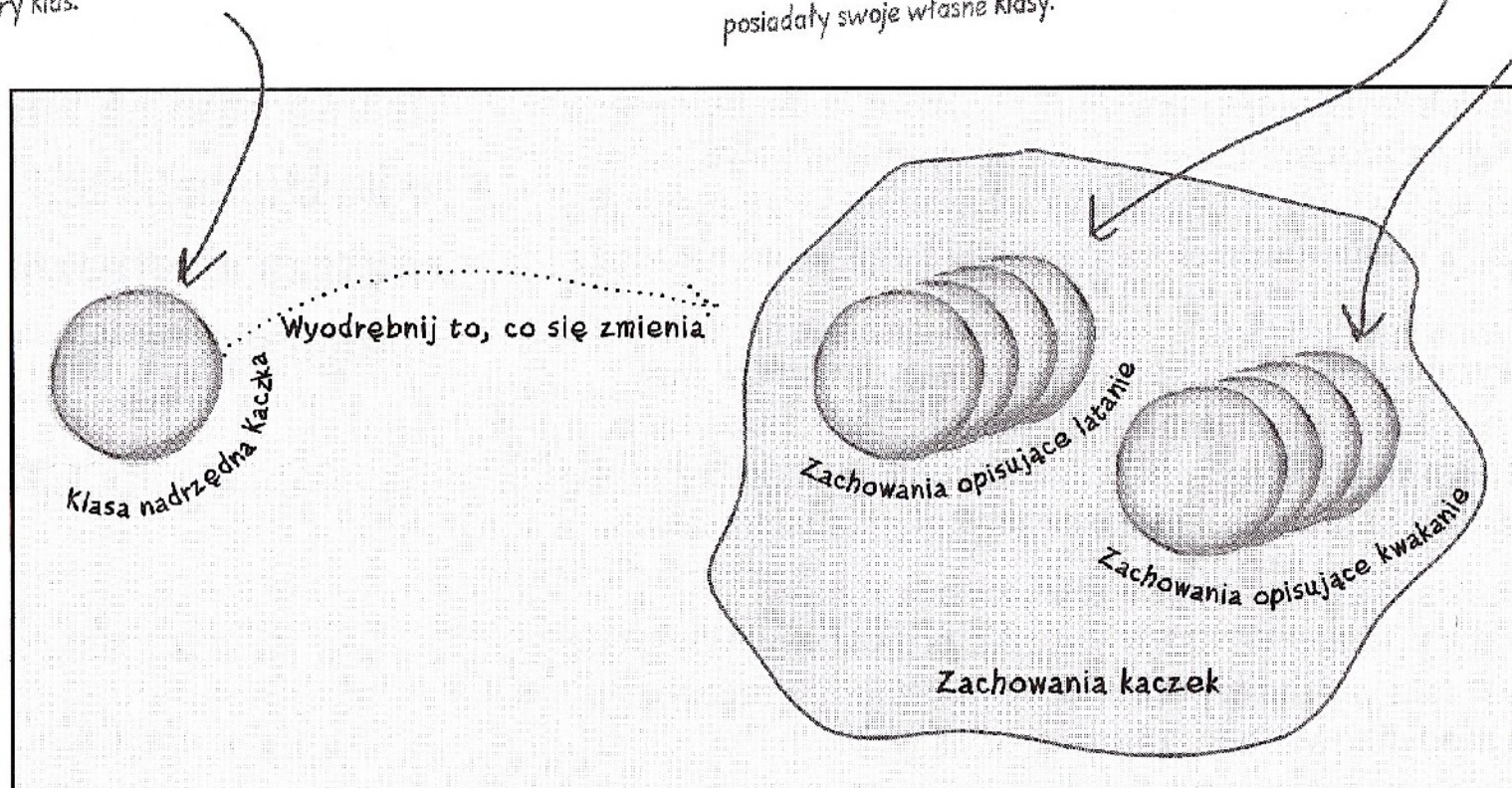
Zidentyfikuj fragmenty aplikacji, które się *zmieniają* i **oddziel** je od tych, które pozostają *stałe*.

# Rodziny algorytmów

Klasa Kaczka jest nadal klasą nadrzędną (superklasą) dla wszystkich kaczek, ale wyciągamy z niej metody opisujące latanie i kwakanie, a następnie wkładamy je do innej struktury klas.

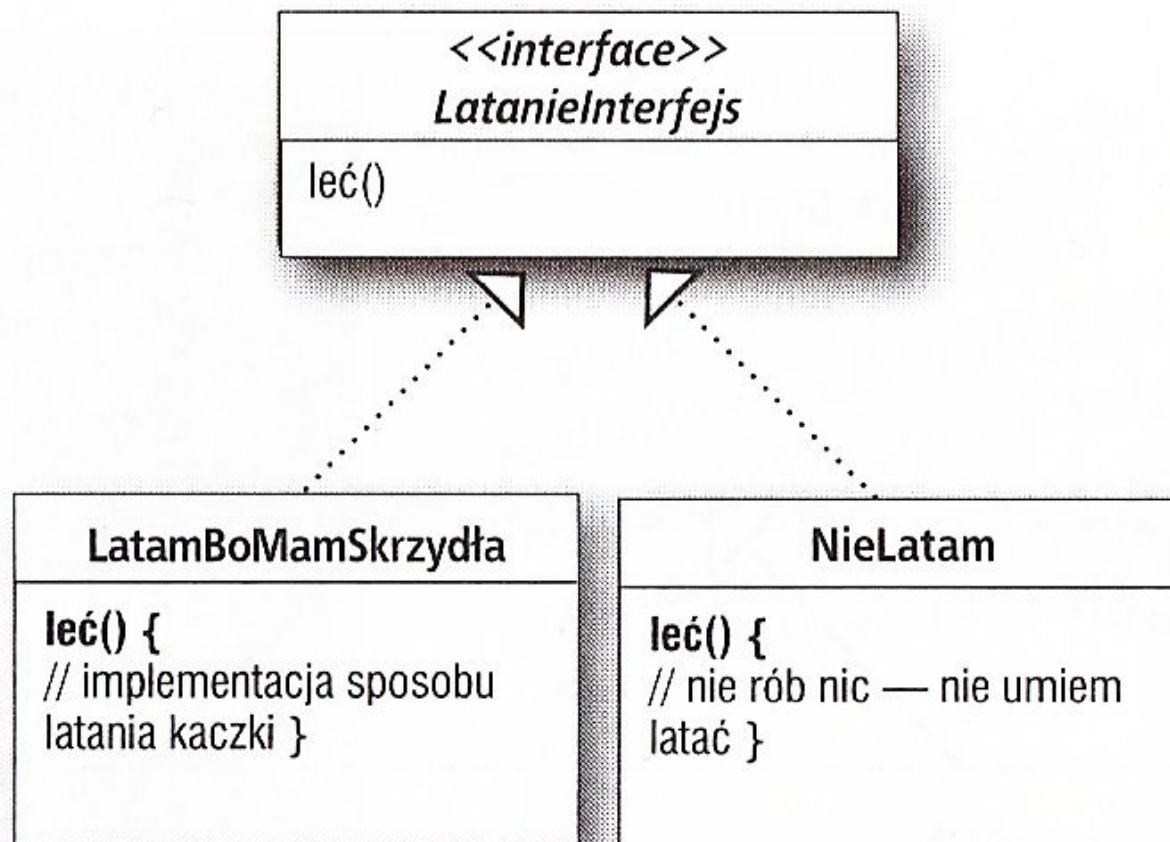
W chwili obecnej latanie i kwakanie będą posiadały swoje własne klasy.

Implementacje poszczególnych zachowań będą przechowywane tutaj.





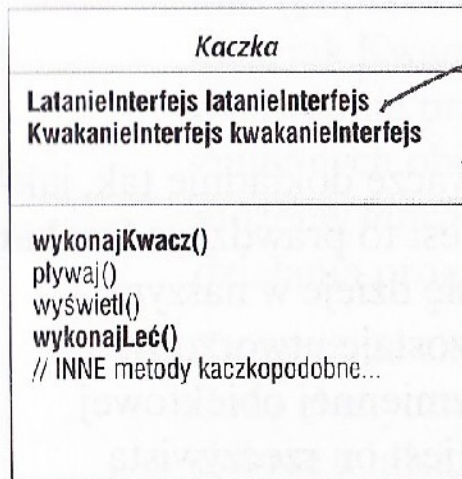
# Rodziny algorytmów



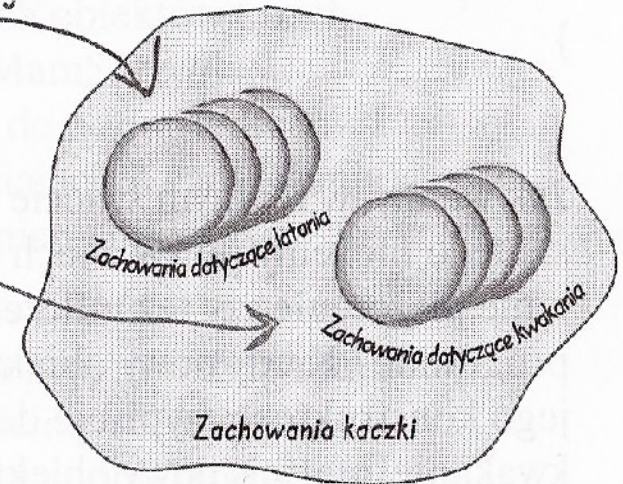
# Kompozycja!

Zmienne opisujące zachowania  
zostały zadeklarowane jako  
INTERFEJSY.

Te metody zastępują  
leć() i kwacz().



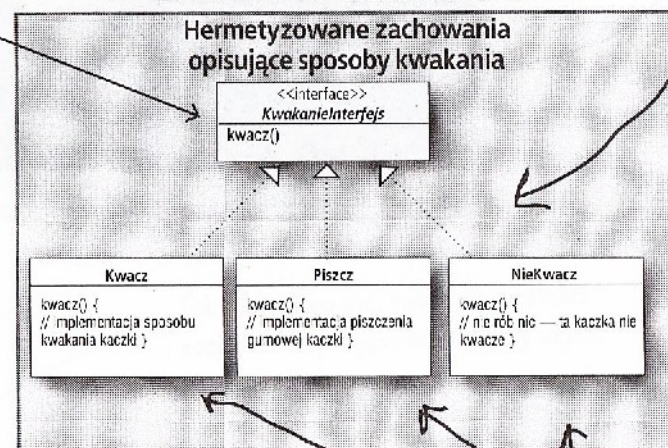
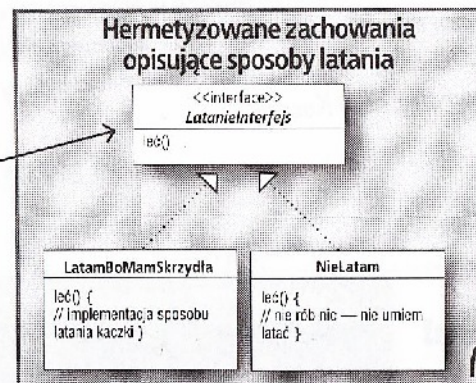
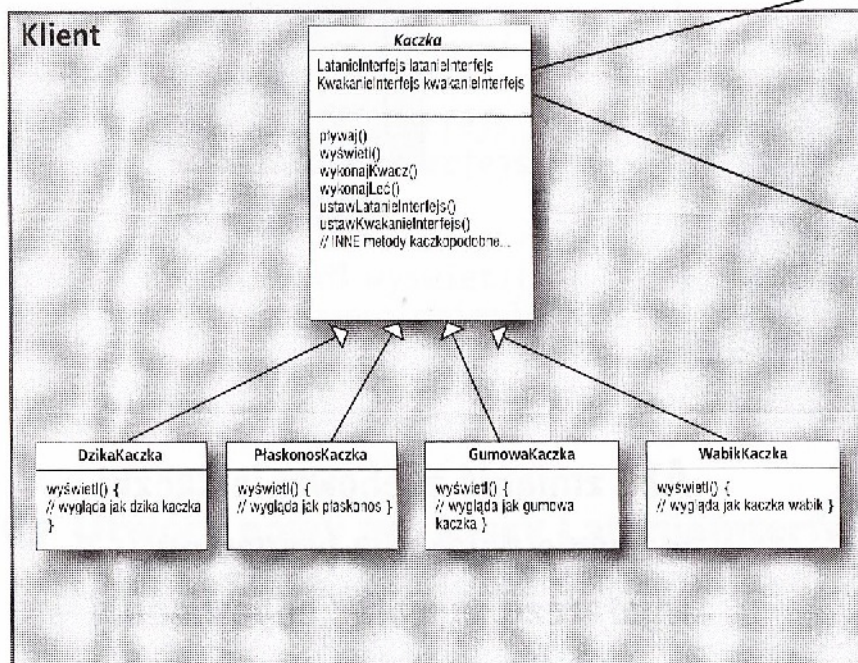
Zmienne obiektowe przechowują odwołania  
do specyficznych zachowań, modyfikowane  
w trakcie działania programu.





# Poprawiony projekt

Klient wykorzystuje hermetyzowaną rodzinę algorytmów do obsługi zarówno latania, jak i kwakania.



Te zachowania  
„algorytmny” są w pełni  
wymienne.

# Podsumowanie

- Zastosowany został wzorzec **Strategia** (ang. *Strategy*).
- Wzorzec strategia definiuje **rodzinę algorytmów**, pakuje je jako **osobne klasy** i powoduje, że są w pełni **wymienne**.
- Zastosowanie tego wzorca pozwala na to, aby zmiany w implementacji algorytmów przetwarzania były całkowicie niezależne od strony klienta, który z nich korzysta.