

Podstawy Programowania Programowanie Obiektowe

Michał Bujacz

bujaczm@p.lodz.pl

**B9 „Lodex” 207
godziny przyjęć: środy i
czwartki 10:00-11:00**

<http://www.eletel.p.lodz.pl/bujacz/>



Pytania powtarzające

$x \& y$?
$x y$?
$\sim x$?
$x \wedge y$?

- o Definicje:

bit, bajt, słowo, little/big endian

Pytania powtarzające - maskowanie



o Rejestr R

R7	R6	R5	R4	R3	R2	R1	R0
?	?	?	?	?	?	?	?

- o Jak sprawdzić co siedzi w R4 ?
- o Jak przestawić R4 na 1?
- o Jak przestawić R4 na 0?

Pytania powtarzające: złożone typy danych



Definicje:

- Tablice/Sekwencje
- Zbiory/słowniki
- Struktury/Rekordy
- Klasy/obiekty

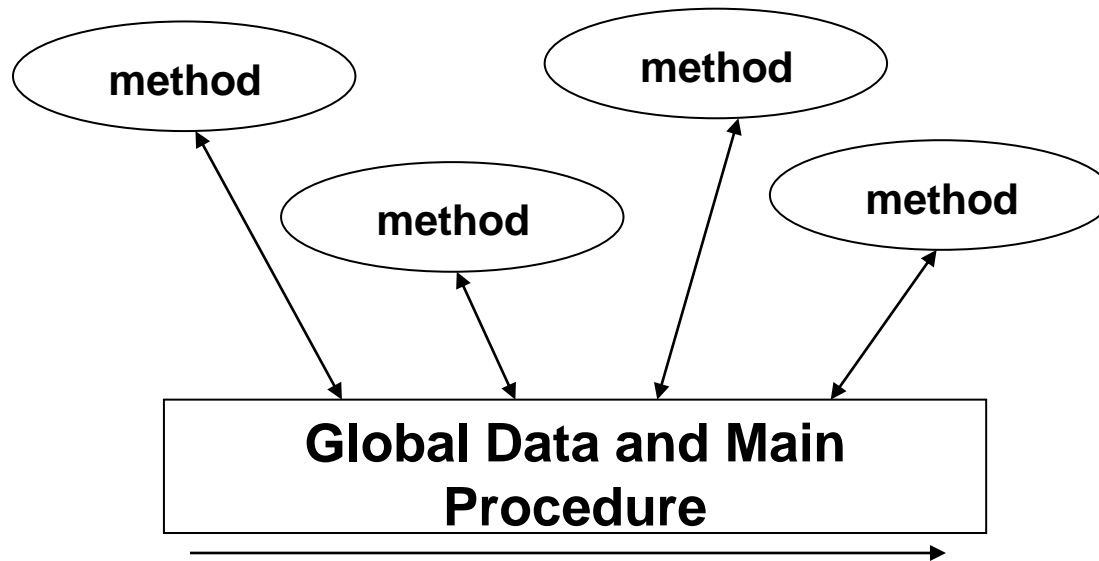
Programowanie Obiektowe



- Paradygmaty programowania
- Imperatywny (strukturalny/proceduralny)

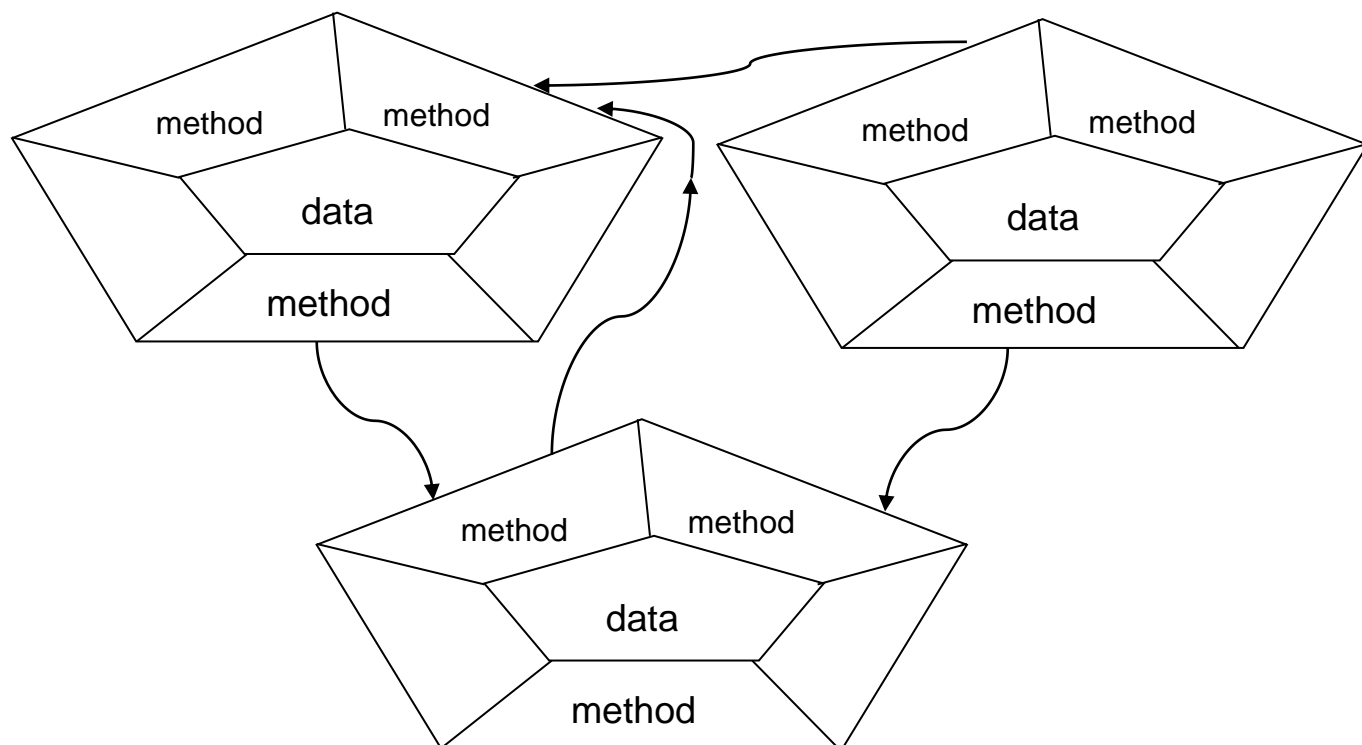
Programowanie Imperatywne

- skupione na instrukcjach
- dane przechowywane w głównej procedurze i przekazywane do i z metod/funkcji/procedur
- jasny rozdział „funkcjonalności” i „danych”



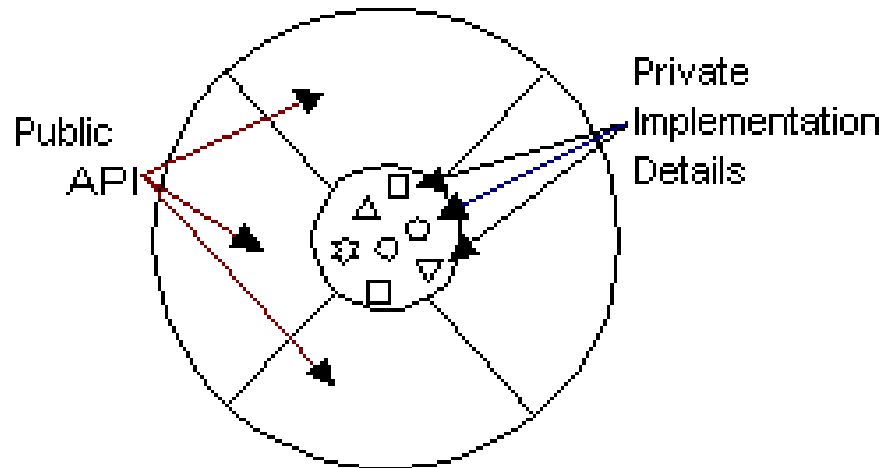
Programowanie obiektowe

- obiekt jest połączeniem danych (stan/atrtrybuty obiektu) i funkcjonalności (metod)
- program polega na tworzeniu obiektów i wysyłania między nimi komunikatów



Klasa

Klasa – definicja typu zmiennej którym są obiekty stanowi opis/schemat (blueprint) jakie dane będzie przechowywał obiekt i jaką funkcjonalnością będzie dysponował (jakie ma metody)



Przykład – klasa „boxer”



Dane:

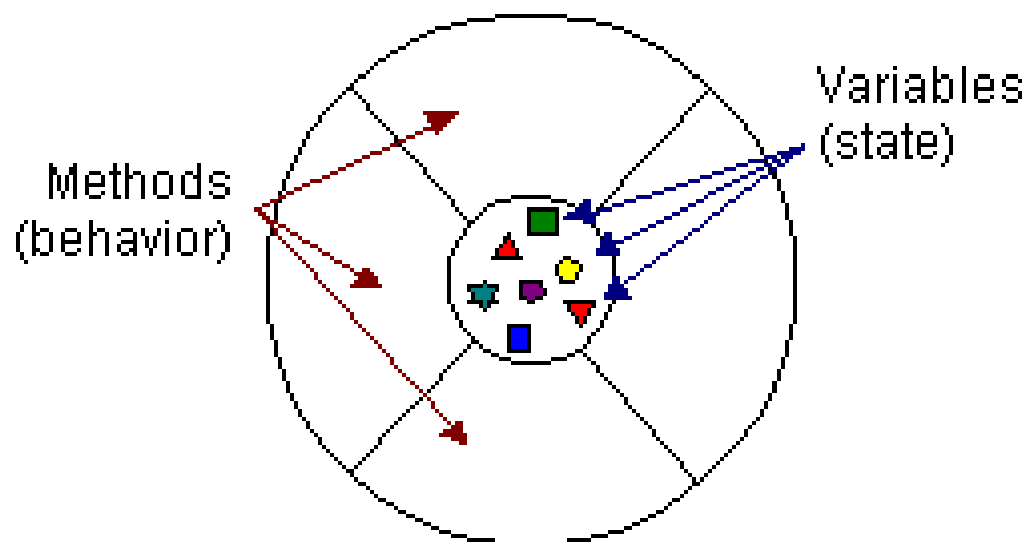
- imię
- zdrowie
- siła

Metody:

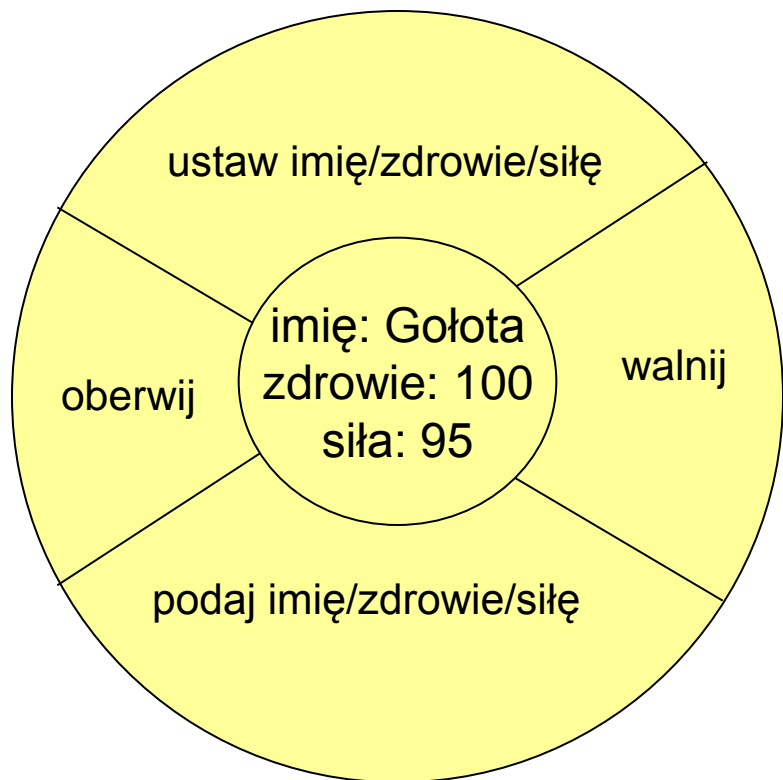
- walnij
- oberwij
- podaj imię/zdrowie/siłę (get)
- ustaw imię/zdrowie/siłę (set)

OBIEKT

Obiekt to „byt programistyczny” posiadający zmienne i metody opisane w klasie



Przykładowy obiekt: Gołota klasy Bokser



Obiekt Gołota:

Dane:

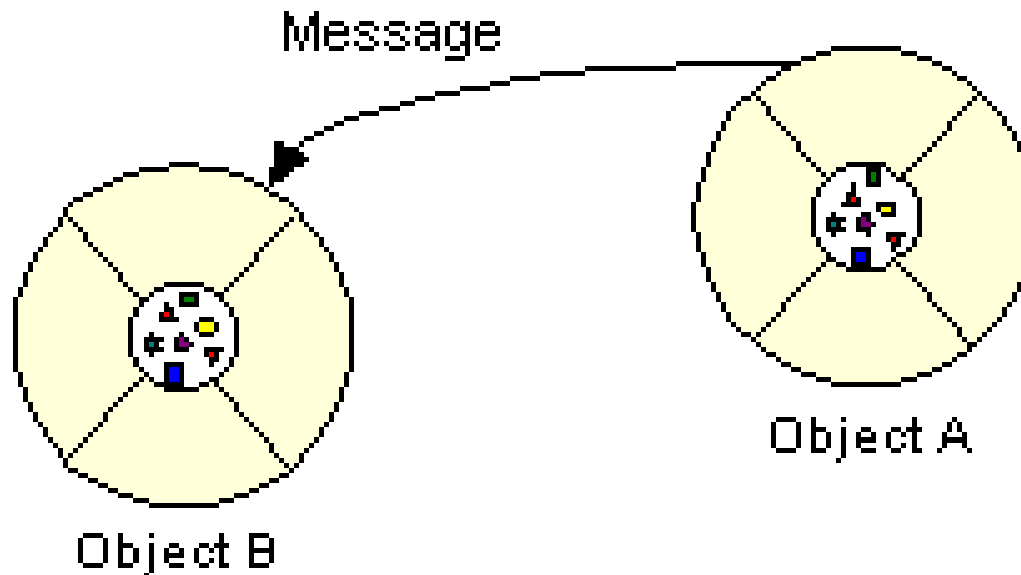
- imię = „Gołota”
- zdrowie = 100
- siła = 95

Metody:

- walnij
- oberwij
- podaj imię/zdrowie/siłę (get)
- ustaw imię/zdrowie/siłę (set)

Komunikaty

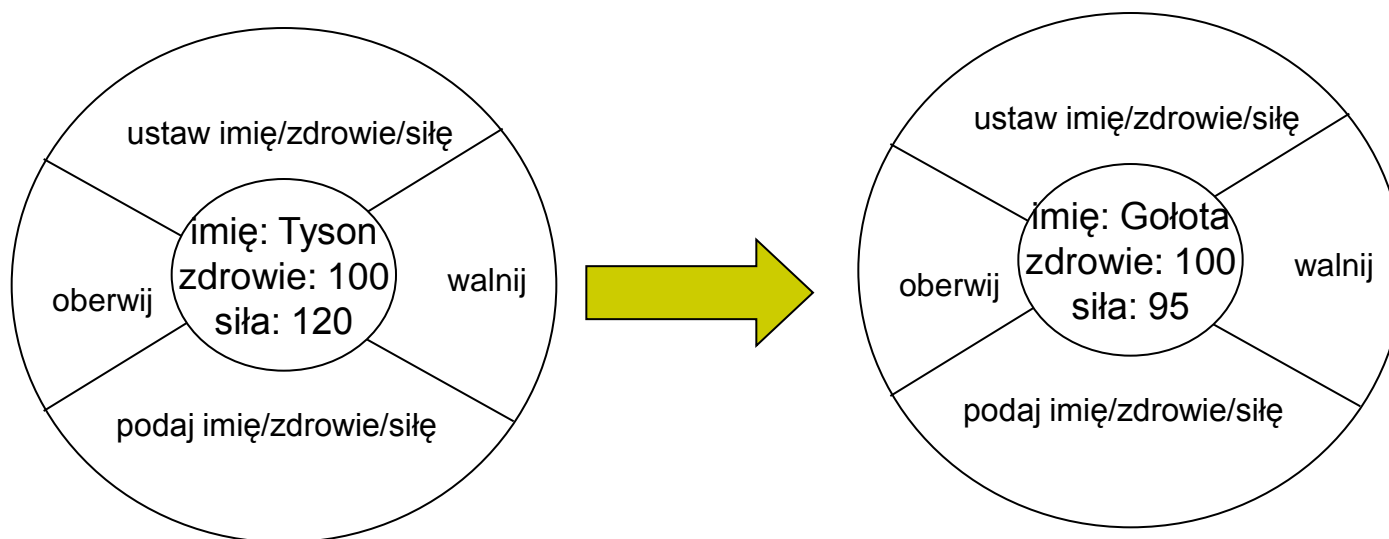
W programowaniu obiektowym instrukcje polegają na wysyłaniu przez obiekty komunikatów (message).



Elementy komunikatu

nazwa obiektu . nazwa metody (parametry metody)

Gołota . Oberwuj (Tyson . Walnij ())



Konstruktor



Metoda uruchamiana w momencie stworzenia obiektu.

Zazwyczaj wykorzystywana do ustawienia wartości początkowych zmiennych.

Czasami uruchamia też statyczne metody (np. licznik obiektów).

Dynamiczne vs. Statyczne



Zmienne i metody dynamiczne – należą do danego obiektu. Każdy obiekt ma własne.

Zmienne i metody statyczne – należą do klasy. Są współdzielone przez obiekty. Mogą być uruchamiane/modyfikowane bez tworzenia obiektu.

Publiczne vs prywatne

- Prywatne dane/metody – mogą być modyfikowane/uruchamiane tylko przez obiekty tej samej klasy
- Publiczne dane/metody – dostępne dla wszystkich innych klas
- Chronione (protected) – dostępne dla danej klasy i klas z niej dziedziczących

Programowanie proceduralne vs programowanie obiektowe



Proceduralne

- dane i funkcjonalność rozdzielne
- program to ciąg instrukcji operujących na danych lub wywołujących metody

Obiektowe

- połączenie danych i funkcjonalności
- każdy obiekt przechowuje jakieś dane (stan/attributy) i posiada funkcjonalność
- klasy opisują obiekty
- program to tworzenie obiektów i wywoływanie ich metod

Wady i zalety

Proceduralne (imperatywne)

+ bardziej zbliżone do maszynowego (więc szybsze i wydajniejsze)

+ dobre to tworzenia krótkich programów przez jednego programistę

- gorsze do tworzenia złożonych programów przez wielu programistów

- trudne w modyfikacji i rozbudowie

Obiektowe

+ dobre dla tworzenia rozbudowanych programów przez wielu programistów

+ naturalnie modeluje interfejsy graficzne

+ łatwe w modyfikacji, rozbudowie

- mniej wydajne

- dużo zbędnego kodu dla małych programów

Nowe słownictwo OOP



Abstraction (abstrakcja)

Polymorphism (polimorfizm)

Inheritance (dziedziczenie)

Encapsulation (enkapsulacja/hermetyzacja)



Abstrakcja



Bardziej „abstrakcyjne” myślenie o programie:

mamy obiekty (agentów) wykonujących swoje zadania i wymieniających komunikaty.

nie musimy skupiać się na wewnętrznym działaniu/strukturze tych obiektów (chyba że to my definiujemy klasę)

Klasy i metody abstrakcyjne – to takie których obiektów nie można tworzyć (służą tylko za schemat do dziedziczenia)

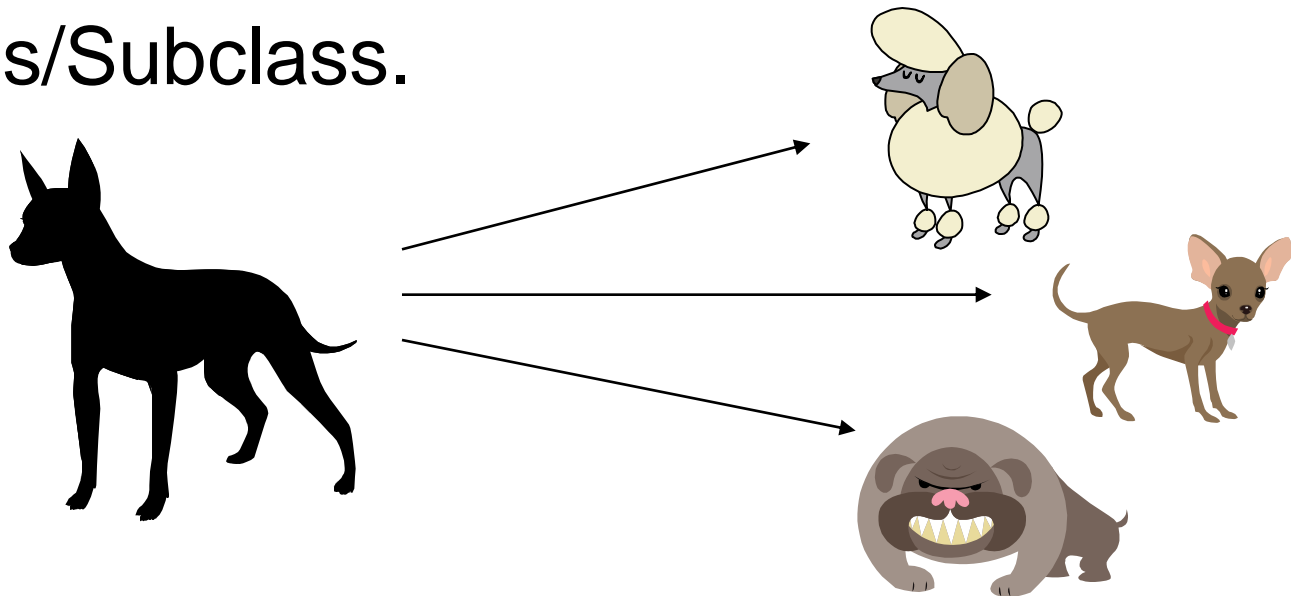


Dziedziczenie

Klasy mogą dziedziczyć z innych. Przejmują wtedy ich dane i metody. Mogą wtedy dodawać swoje nowe dane i metody lub „przeciążać” (nadpisać) stare.

Rodzic/dziecko.

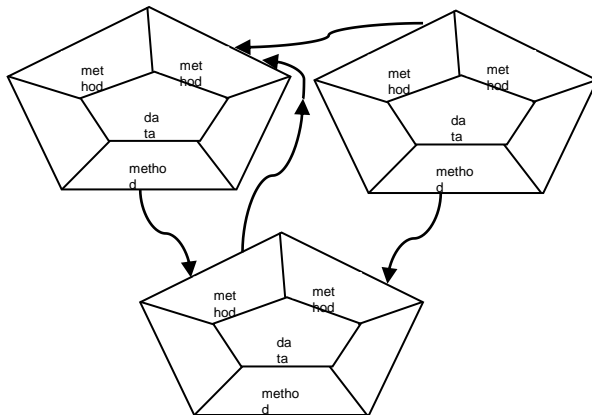
Superclass/Subclass.



Hermetyzacja

Dane ustawiane jako prywatne, metody jako publiczne. Pozwala to uniknąć przypadkowego edytowania danych lub błędów dostępu do pamięci.

Interfejs – typ klasy składającej się tylko z publicznych metod, bez danych.



Polimorfizm

Możliwość używania metod/zmiennych na kilka różnych sposobów. Np. ta sama metoda zadziała inaczej w zależności od typu danych który otrzyma, lub obiektu z którego została uruchomiona.

override (zastąpić) – dziedzicząca klasa tworzy własną wersję jakiejś metody

overload (przeciążyć) tworzymy inną metodę o tej samej nazwie ale przyjmującą inne parametry

