



PRZYPOMNIENIE: KLASY, OBIEKTY I KONSTRUKTORY



Co to jest klasa?

Klasa to szablon do tworzenia obiektów.

Zawiera:

- pola (dane/atrybuty),
- metody (zachowanie),
- konstruktory.



Co to jest obiekt?

Obiekt to instancja klasy, czyli faktyczny twór w pamięci, który powstaje na podstawie klasy.



Konstruktory

Konstruktor to specjalna metoda, której zadaniem jest inicjalizacja obiektu w momencie jego utworzenia.

Cechy konstruktora:

- ma taką samą nazwę jak klasa,
- nie ma typu zwracanego (nawet `void`),
- może przyjmować parametry,
- można mieć wiele konstrktorów (przeciążanie).



Domyślny konstruktor

Jeśli nie napiszesz konstruktora, Java i tak stworzy konstruktor domyślny:

```
public Samochod() {}
```

Działa tylko wtedy, gdy nie napiszesz własnego.



Konstruktor

- Konstruktor to specjalna metoda, która ustawia początkowe wartości pól.
- Można tworzyć wiele konstruktorów z przeciążaniem, np. z różnymi parametrami.



Słowo kluczowe this

- Odnosi się do bieżącego obiektu.
- Przydaje się, gdy nazwa pola i nazwa parametru są takie same.



POLIMORFIZM



Co to jest polimorfizm?

Polimorfizm odnosi się do sytuacji, gdy obiekty różnych klas mogą być traktowane tak samo, jeśli mają wspólną klasę bazową.

W praktyce oznacza to:

Jeden typ → wiele zachowań.



O co w tym chodzi

Wyobraź sobie, że:

- masz różne obiekty,
- każdy z nich potrafi wykonać tę samą czynność,
- ale każdy robi to na swój sposób.

Program:

- nie musi wiedzieć, z jakim dokładnie obiektem pracuje,
- wystarczy, że wie, co obiekt potrafi zrobić.



Jak to działa w Javie

W Javie:

- obiekty mogą być traktowane jako bardziej ogólne, ale w środku nadal są konkretne.

Gdy wywołujesz metodę:

- Java sprawdza, jakiego typu jest obiekt i wybiera odpowiednią wersję działania.



Dlaczego polimorfizm jest potrzebny?

Polimorfizm:

- upraszcza kod,
- pozwala pisać programy z łatwą możliwością rozwoju w przyszłości, sprawia, że kod jest bardziej elastyczny.

Dzięki niemu:

- nie trzeba sprawdzać typu obiektu,
- nie trzeba pisać wielu warunków,
- łatwiej dodawać nowe elementy do programu.



Co daje polimorfizm programiście?

- Mniej powtarzającego się kodu
- Lepszą czytelność
- Łatwiejszą rozbudowę programu



Czego polimorfizm NIE robi?

- Nie miesza obiektów
- Nie zmienia typów
- Nie zgaduje
- Nie losuje