



# Programowanie Obiektowe

---

Grzegorz Hałdaś

Technologie Internetowe



# Plan Wykładu

---

- Dynamiczna alokacja pamięci
- Konstruktor kopiujący
- Przeciążanie (przeładowywanie) funkcji operatorowych
- Dziedziczenie



# Dynamiczna alokacja pamięci

- przydział pamięci

`Typ * pt;`

1.

`pt = new Typ;`

2.

`pt = new Typ(...);`

3.

`pt = new Typ[l_elem];`

- zwalnianie pamięci

1.

`delete pt;`

2.

`delete pt;`

3.

`delete[] pt;`



# Dynamiczna alokacja pamięci

```
Data *pd;  
pd = new Data;  
delete pd;
```

```
pd = new Data(21, 3, 2008);  
delete pd;
```

```
pd = new Data[10];  
delete [] pd;
```



# Konstruktor kopiujący

---

Miejsca, w których mamy do czynienia z kopiowaniem obiektów:

- w miejscu deklaracji obiektu inicjowanie go innym istniejącym obiektem
- przy przekazywaniu obiektu przez wartość do funkcji
- przy zwracaniu z funkcji obiektu przez wartość



# Przeciążanie operatorów

Zasady definiowania funkcji operatorowych:

- definiujemy je za pomocą słowa kluczowego **operator**
- zachowana jest liczba argumentów danego operatora
- nie zmienia się priorytet
- operatory 1) `.` 2) `.*` 3) `::` 4) `?:` nie można przeładowywać
- nie można definiować nowych operatorów



# Przeciążanie operatorów

Zasady definiowania funkcji operatorowych (c.d.):

- operatory predefiniowane (tzn. takie które mogą być używane bez konieczności wcześniejszego ich definiowania) – są to:  
1) `=`, 2) `&` (pobranie adresu obiektu) , 3) `new`, 4) `delete`, 5) ,  
(przecinek).
- operatory 1) `=`, 2) `->`, 3) `()`, 4) `[]` muszą być definiowane jako funkcje składowe klasy
- deklaracja operatora będąca funkcją składową posiada o jeden argument mniej (na rzecz niejawnego argumentu `this`) niż taka sama funkcja operatorowa definiowana jako funkcja zwykła