



Programowanie Obiektowe

Grzegorz Hałdaś

Technologie Internetowe



Plan Wykładu

- Argumenty domyślne (dokończenie)
- Konstruktor
- Destruktor
- Dynamiczna alokacja pamięci



Wprowadzenie do konstruktora

```
int i;  
int j=1;  
int k(2); //int k=2;  
int m=int(3);  
  
// Definiujemy klasę INT, która ma mieć  
// cechy podobne do typu wbudowanego int.  
  
// Definiowanie obiektu klasy INT  
INT I;    // Bez konstruktora tej klasy  
// nie ma innej możliwości inicjowania
```



Konstruktor

- to metoda klasy nazywająca się tak samo jak klasa
- uruchamia się automatycznie na rzecz nowopowstałego obiektu
- nie konstruuje obiekt klasy tylko inicjalizuje składowe klasy
- nie zwraca żadnej wartości
- może być przeładowywany



Destruktor

- to metoda klasy nazywająca się tak samo jak klasa poprzedzona znakiem ~ (tyldy)
- uruchamia się automatycznie na rzecz obiektu tuż przed jego likwidacją
- nie likwiduje obiektu
- nie posiada żadnych argumentów
- nie zwraca żadnej wartości
- nie może być przeładowywany



Dynamiczna alokacja pamięci

- tablice zastępujemy wskaźnikami
- definicje operatorów:
 - przydział pamięci:

```
void* operator new(size_t s);
```

- zwalnianie pamięci:

```
void operator delete(void* p);
```



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