

→ JEŚLI KLASA NIE MA ŻADNEGO KONSTRUKTORA, WOLICZAS KOMPILATOR SAM WYGENERUJE DLA TEJ KLASY KONSTRUKTOR DOMNIEMANY (public, inline)

→ C++11 • default : konstruktor() = default ; // „wymuszenie” wygenerowania konstruktora domniemanego
• delete : konstruktor() = delete ; // „zakazanie” konstruktora domniemanego

✓ MOŻE BYĆ PRZELADOWANY
(killuje konstruktory w klasie -
- różna lista argumentów)

✓ NIE MOŻNA POSŁUŻYĆ SIĘ JEGO
ADRESEM

✓ NICZEGO NIE ZWRACA
(nawet typu void)

✓ MOŻE BYĆ WYWOŁANY DLA
TWORZENIA OBIEKTÓW Z
przdomkami const i volatile -
- SAM NIE MOŻE BYĆ FUNKCJĄ
const/volatile

✓ NIE MOŻE BYĆ TYPU STATIC

- specjalna funkcja stadora (nazywa się tak samo jak klasa)
- w ciele konstruktora możemy zamieścić instrukcje nadające wartości początkowe zmiennym def. obiektu

KONSTRUKTOR (NIE BUDOWNICZY - dekorator wewnątrz)

- nie przydziela pamięci (sam) - inicjalizuje
- AUTOMATYCZNIE uruchamiany przy definiowaniu każdego obiektu tej klasy

- funkcja wywoływana samoczynnie (gdy obiekt klasy jest likwidowany)
- klasa NIE musi mieć dopiękoko destruktora

✓ np. ZWOLNIENIE PAMIĘCI REZERWOWANEJ
DYNAMICZNIE (umieszczenie instrukcji
delete)

✓ NIE MOŻE ZWRACAĆ ŻADNEJ WARTOŚCI
(nawet typu void)

✓ MA PUSTĄ LISTĘ ARGUMENTÓW

✓ NIE MOŻE BYĆ PRZELADOWANY

✓ NIE POWINNO SIĘ RZUCAĆ WYJĄTKU

DESTRUKTOR

- destruktor nie zwalnia obszaru pamięci, który obiekt zajmował
- destruktor nie likwiduje obiektu

LISTA INICJALIZACYJNA

- lista/sygn. mówiąca konstruktorowi, jak ma inicjalizować wybrane zmienniki klasy (niestatyczne, czyli anglice)
- zmienniki z przdomkiem const - inicjalizacja TYLKO lista in.