

Dinamički životni vek

❖ Izraz (operator) *new* uvek radi sledeće stvari, ovim redom:

1. alokira prostor za smeštanje jednog objekta datog tipa ili niza objekata datog tipa, ako se pravi niz
2. inicijalizuje jedan objekat ili svaki objekat u nizu (ako se pravi niz objekata) zadatim inicijalizatorom
3. vraća vrednost tipa pokazivača na dati tip koji ukazuje na napravljeni objekat, ili na prvi element napravljenog niza (ako se pravi niz objekata datog tipa)

❖ Alokacija prostora (prvi korak) vrši se pozivom neke od (preklopljenih) operatorskih funkcija koje su standardno definisane jezikom (postoje i ovakve operatorske funkcije sa još nekim parametrima):

void* operator new (std::size_t count);

Poziva se ako se pravi jedan objekat

void* operator new[] (std::size_t count);

Poziva se ako se pravi niz objekata

❖ Tip *size_t* je neoznačeni celobrojni tip deklarisan u nekoliko zaglavlja standardne biblioteke koji se koristi za izražavanje veličina tipova; ovo je tip rezultata operatora *sizeof*

❖ Parametar ovih funkcija tipa *size_t* prenosi veličinu prostora koji treba alocirati (u jedinicama *sizeof(char)*)

❖ Podrazumevana implementacija ovih funkcija upravlja slobodnom memorijom (*heap, free store*), odnosno radi dinamičku alokaciju i dealokaciju u za to predviđenom delu memorije programa

❖ Programer može promeniti način alokacije prostora na neki od sledećih načina:

- zameniti implementaciju neke od ovih funkcija svojom implementacijom, čime će promeniti način alokacije dinamičke memorije (preusmeriti na svoj alokator); dovoljno je samo negde u programu definisati takvu funkciju, bez posebne deklaracije na nekom drugom mestu
- definisati ove funkcije za svoje klasne tipove, čime će promeniti način alokacije prostora samo za objekte te klase

Dinamički životni vek

- ❖ Izraz (operator) *new* ima redom sledeće elemente iza ključne reči *new*:
 - opcione argumente za tzv. *parametre smeštanja* (*placement parameters*)
 - specifikator objektnog tipa, kao *type-id*, za objekat koji se kreira
 - opcioni inicijalizator, kao inače pri inicijalizaciji varijabli na drugim mestima

```
new (std::nothrow) Clock (9,0,0)
```

- ❖ Specifikator tipa je *type-id* koji mora biti objektni tip (ne tip reference), i može biti tip koji predstavlja jedan objekat ili niz objekata nekog tipa:

```
new (Clock*) (nullptr)
```

```
new (Clock*[n]) {}
```

- ❖ Ukoliko je specifikovan tip niza, sve dimenzije tog niza osim prve moraju biti date kao konstatni izrazi sa pozitivnom celobrojnomo vrednošću; prva dimenzija može biti izraz koji se izračunava u trenutku izvršavanja izraza *new* - tako se jedino i mogu praviti dinamički dimenzionisani nizovi (sa dimenzijom poznatom tek za vreme izvršavanja):

```
const int Max = 100;  
extern int n;  
new (Clock[2*n][Max]) (nullptr)  
new (Clock[2*n][n]) {}
```

- ❖ U specifikatoru tipa (*type-id*) može da se koristi i *auto*, s tim da je inicijalizator tada obavezan, jer se na osnovu njegovog tipa određuje tip objekta koji se kreira