

Konstantna inicijalizacija

❖ Primer: struktura za alokaciju prostora za objekte tipa T , bez fragmentacije (ostataka slobodnog prostora):

```
template <class T, int size>
class Storage {
public:
```

Konstruktor je *constexpr*, pa se može koristiti za konstantnu inicijalizaciju statičkih objekata u toku prevođenja

```
constexpr Storage () : head(slots) { slots[size-1].next = nullptr; }

void* alloc () { Slot* p=head; if (p) head=p->next; return p->slot; }
void free (void* addr) { head = new (addr) Slot(head); }
```

```
private:
```

Članovi moraju imati inicijalizatore koji su konstantni izrazi

```
struct Slot {
    constexpr Slot () : next(this+1) {}
    Slot (Slot* nxt) : next(nxt) {}
```

Konstruktor strukture *Slot* je *constexpr*, pa se može pozvati iz konstruktora *Storage* koji je *constexpr*

```
    union {
        Slot* next;
        char slot[sizeof(T)];
    };
};

Slot* head;
Slot slots[size];

Storage<Clock, 50> clockStorage;
```

Ovaj statički objekat bi mogao da se inicijalizuje konstantnom inicijalizacijom u toku prevođenja, ali nažalost, izgleda da još uvek nema načina da se ovo garantuje na prenosiv način u opštem slučaju, pa će inicijalizacija ipak po pravilu biti dinamička. Ako bi se ovaj objekat deklariseo kao *constexpr*, njegova inicijalizacija bi bila konstantna, ali bi i on bio konstantan i ne bi mogao da se menja

Zadatak:

Funkcija *Store::free* nije pouzdana ako joj se dostavi pokazivač koji ne ukazuje na neki od elemenata niza *slots*. Napraviti je tako da bude otporna na ovakve greške.

Inicijalizacija nulom

❖ *Inizijalizacija nulom (zero initialization)* se obavlja u sledećim slučajevima:

- Za svaku imenovanu varijablu sa statičkim trajanjem skladišta ili trajanjem skladišta vezanim za nit (*thread local*), pod uslovom da nije inicijalizovana konstantnom inicijalizacijom, a pre svake druge inicijalizacije:

```
static T t;
```

- Kao deo postupka inicijalizacije vrednošću, za neklasne tipove i za članove klasnih tipova koji su inicijalizovani vrednošću i koji nemaju konstruktore, uključujući inicijalizaciju vrednošću agregata za koje nisu zadati inicijalizatori:

```
T();
```

```
T t = {};
```

```
T{};
```

- Kada se niz znakova inicijalizuje string literalom koji je kraći od tog niza, ostatak niza se inicijalizuje nulama:

```
char a[n] = "";
```