

Operatori *new* i *delete*

- ❖ Jedna ideja za alokaciju prostora za objekte klase *X* koja nema problem fragmentacije, jer objekte smešta u niz slotova veličine tipa *X*, pri čemu slobodne slotove ulančava u listu (pa su operacije alokacije i dealokacije kompleksnosti $O(1)$):

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

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

```
    void* alloc () { Slot* p=head; if (p) head=p->next; return p?p->slot:nullptr; }
```

```
    void free (void* addr) { head = new (addr) Slot(head); }
```

```
private:
```

```
    struct Slot {
```

```
        Slot () : next(this+1) {}
```

```
        Slot (Slot* nxt) : next(nxt) {}
```

```
        union {
```

```
            Slot* next;
```

```
            char slot[sizeof(T)];
```

```
        };
```

```
    };
```

```
    Slot* head;
```

```
    Slot slots[size];
```

```
};
```

Poziva se ugrađena alokatorska funkcija za *placement new* i klasu *Slot*

Operatori *new* i *delete*

- ❖ Sada se ova pomoćna klasa za alokaciju može koristiti na sledeći način:

```
class X {
public:
    ...
    void* operator new (size_t) {
        void* addr = storage.alloc();
        if (!addr) throw std::bad_alloc;
        else return addr;
    }
    void operator delete (void* addr) { storage.free(addr); }
private:
    static Storage<X,2000> storage;
};
Storage<X,2000> X::storage;
```

- ❖ Nedostatak ovog rešenja jeste to što se unapred, statički mora zadati veličina skladišta; potrebnu veličinu je često teško odrediti, jer zavisi od dinamičke prirode programa, a čak i ako se odredi, može biti premala (da ne može da zadovolji trenutne potrebe, iako slobodne memorije i dalje ima) ili prevelika (pa se memorija nepotrebno rezerviše i drži zauzetom za skladište, iako nije potrebna za objekte). Ova veličina može se zadati i dinamički, ali pri njenoj inicijalizaciji (unaprediti ovo rešenje na taj način)
- ❖ Postoje i naprednija rešenja koja nemaju ovo ograničenje; zadatak: osmisлити i implementirati neko takvo rešenje