

Konverzije

- ❖ Eksplicitna konverzija operatorom *const_cast* dozvoljena je samo za ukidanje cv-kvalifikacije (ili dodavanje takve kvalifikacije, ali se to može raditi i implicitno), najčešće za pokazivače i reference:

```
const Clock* pcc = new Clock;  
Clock* pc = const_cast<Clock*>cc;
```

Ovde implicitna konverzija nije moguća

- ❖ Eksplicitna konverzija operatorom *static_cast* dozvoljena je za skoro sve ugrađene konverzije, svakako za one koje se mogu raditi implicitno, kao i za one koje se smatraju nebezbednim (ali imaju bilo kakvog smisla), s tim da *ne može* da ukida cv-kvalifikaciju; u svakom slučaju, za vreme izvršavanja ne vrši se nikakva provera korektnosti, npr. u smislu da li je objekat koji se krije iza pokazivača / reference zaista odredišnog tipa (npr. za *downcast* pokazivača i referenci):

```
Base* pb = new Derived;  
Derived* pd = static_cast<Derived*>pb;
```

Ovde implicitna konverzija nije moguća

- ❖ Eksplicitna konverzija operatorom *dynamic_cast* namenjena je za konverzije između tipova u hijerarhiji klasa, i to nagore, nadole ili bočno, s tim da ona u vreme izvršavanja proverava tip polimorfnog objekta (objekta klase sa bar jednom virtuelnom funkcijom), i u slučaju da izvorišna vrednost nije odredišnog tipa, vraća *null* vrednost za pokazivače ili baca izuzetak za reference:

```
Base* pb = ...;  
Derived* pd = dynamic_cast<Derived*>pb;
```

Statička konverzija ne bi garantovala da *pd* zaista ukazuje na objekat klase *Derived*

- ❖ Eksplicitna konverzija operatorom *reinterpret_cast* može da vrši konverzije skoro svih tipova, osim da ukida cv-kvalifikacije, s tim da se za nju ne generišu nikakve mašinske instrukcije; jednostavno se binarna vrednost operanda tumači na drugačiji način, u skladu sa odredišnim tipom; veoma je osetljiva i retko potrebna

Konverzije

- ❖ Kod eksplicitne konverzije operatorom *cast* nasleđenim iz jezika C

(new_type)expression

prevodilac pokušava redom sledeće konverzije izvorišnog u odredišni tip, i primenjuje prvu koja odgovara datim tipovima, čak i ako ona nije dozvoljena (u kom slučaju prijavljuje grešku):

a) *const_cast<new_type>(expression)*

b) *static_cast<new_type>(expression)*, sa tim proširenjem što pokazivač/referenca na izvedenu klasu može da se konvertuje u pokazivač/referencu na osnovnu klasu, čak i ako ta osnovna klasa nije dostupna (to znači, ne uzima se u obzir to što je osnovna klasa izvedena kao *private* ili *protected*)

c) *static_cast* (sa proširenjem), iza koje sledi *const_cast*

d) *reinterpret_cast<new_type>(expression)*

e) *reinterpret_cast*, iza koje sledi *const_cast*

- ❖ Prema tome, ovim operatorom mogu se vršiti sve konverzije navedenih operatora specifičnih konverzija, osim operatora *dynamic_cast*, ali se ipak preporučuje korišćenje upravo nekog od ovih operatora umesto operatora *cast* iz sledećih razloga:

- upotrebom nekog od navedenih specifičnih operatora naglašava se namera date konverzije, tj. njena svrha i cilj, pa program može biti razumljiviji
- ukoliko ta namera nije korektna, u smislu da zahtevana konverzija nije dozvoljena tim operatorom za date tipove, prevodilac će programeru ukazati na grešku (dok bi operator *cast* “ćutke” možda primenio neku drugu, nenameravanu konverziju)