
Konverzije

- ❖ Implicitna konverzija se vrši kad god se izraz jednog tipa $T1$ koristi u kontekstu u kom se ne prihvata taj tip, nego se prihvata neki drugi tip $T2$, a to su sledeće situacije:
 - kada se izraz tipa $T1$ koristi kao stvarni argument poziva funkcije čiji je odgovarajući formalni parametar deklarisan sa tipom $T2$
 - kada u funkciji koja ima deklarisan povratni tip $T2$ naredba *return* vraća izraz tipa $T1$
 - kada se izraz tipa $T1$ koristi kao operand operatora koji očekuje taj operand tipa $T2$
 - kada se inicijalizuje novi objekat tipa $T2$ izrazom tipa $T1$
 - kada je izraz upotrebljen u naredbi *switch* tipa $T1$ koji nije integralan tip ($T2$ je integralan tip)
 - kada je izraz upotrebljen u naredbi *if* ili u naredbi petlje (naredbe *for*, *do*, *while*) tipa $T1$ koji nije tip *bool* ($T2$ je tip *bool*)
- ❖ Implicitna konverzija može se vršiti i tranzitivno, tj. prevodilac može izvršiti i čitav niz sukcesivnih implicitnih konverzija, ali taj niz ne može sadržati više od jedne korisnički definisane konverzije — sve ostale moraju biti tzv. *standardne konverzije* (*standard conversion*)

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;
```

- ❖ 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;
```

- ❖ 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;
```

- ❖ 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