
Konverzije

- ❖ Konverzija je pretvaranje vrednosti (*value*) jednog tipa u vrednost drugog tipa
- ❖ Konverzija može, a ne mora imati bilo kakve efekte u vreme izvršavanja
- ❖ Konverzija može imati efekta samo na prevođenje, bez ikakvog efekta (same konverzije) za vreme izvršavanja (prevodilac ne generiše nikakav poseban kod koji vrši tu konverziju); prevodilac u vreme prevođenja može prosto smatrati da nakon konverzije ista vrednost ima drugi tip, i tu vrednost nadalje drugačije tumačiti;
na primer: konverzija objekta nekonstantnog tipa u isti takav konstantni tip, konverzija pokazivača na izvedenu klasu u pokazivač na jedinu (i dostupnu) osnovnu klasu (i dalje ukazuje na isti objekat), konverzija iz *false* u 0, konverzija iz *char* u *int* i slično
- ❖ Konverzija može imati efekat u vreme izvršavanja, pa prevodilac generiše odgovarajući kod koji za vreme izvršavanja vrši tu konverziju, na primer tako što:
 - vrednost jednog tipa mora da dobije drugačiju binarnu predstavu za vreme izvršavanja, a možda i konceptualno drugačiju vrednost: na primer, konverzija racionalnog broja tipa *double* u ceo broj *int* promeniće svakako svoju binarnu predstavu (iz pokretnog zareza u drugi komplement), ali možda i vrednost (bez decimala)
 - konverzijom može nastati potpuno novi objekat, pa i objekat klase

Konverzije

- ❖ Konverzija može biti definisana (u smislu da je propisan način kako se vrši, za koje tipove i kada se može vršiti i šta joj je semantika):
 - samim pravilima jezika: tzv. *ugrađena (built-in)*; na primer, konverzija *char* u *int*, ili konverzija pokazivača / reference na izvedenu klasu u pokazivač / referencu na dostupnu osnovnu klasu
 - u programu: tzv. *korisnički definisane konverzije (user-defined conversion)*, za korisničke tipove, odnosno klase
- ❖ Konverzija se može raditi:
 - *implicitno*, što znači da je radi prevodilac bez ikakvog posebnog eksplicitnog zahteva programera; ako je na datom mestu potrebna konverzija, i ako je ona definisana tako da se može raditi implicitno, prevodilac će je izvršiti; ovakve konverzije su samo one koje se smatraju “bezbednim”, u smislu da na neki način čuvaju korektnost semantike programa
 - eksplicitno zahtevane u programu jednim od sledećih operatora:
 - *const_cast*
 - *static_cast*
 - *dynamic_cast*
 - *reinterpret_cast*
 - operator *cast* nasleđen iz jezika C: *(type)expression*