
Pokazivači na objekte

- ❖ Na jeziku C++, kao i na jeziku C, vrednosti pokazivača predstavljaju adrese u memoriji. Međutim, ovo je stvar implementacije i konkretna vrednost pokazivača ne treba da bude od interesa, osim u prilikama kada se program debuguje i prati ta vrednost (u smislu jednakosti ili promena)
- ❖ Svaka vrednost tipa pokazivača na objekat može biti jedna od sledećih:
 - takva da ukazuje na neki objekat, ili
 - takva da ukazuje iza kraja nekog objekta, ili
 - *null* vrednost tog tipa, ili
 - invalidna (nekorektna) vrednost pokazivača
- ❖ Vrednost pokazivača koji ukazuje na neki objekat predstavlja adresu prvog bajta u memoriji koji zauzima taj objekat. Vrednost pokazivača koji ukazuje iza nekog objekta predstavlja adresu prvog bajta iza dela memorije koju zauzima taj objekat
- ❖ Dva pokazivača koji predstavljaju istu adresu mogu imati različite binarne vrednosti kod nekih arhitektura, ali njihovo poređenje na jednakost daje *true*
- ❖ Prva tri slučaja su korektne situacije, odnosno vrednost pokazivača je validna, iako se samo u prvom slučaju ta vrednost može dereferencirati (pristupiti objektu na koji ta vrednost ukazuje) sa definisanim značenjem; u svim drugim situacijama pristup do objekta ima nepredvidivo ponašanje: može izazvati hardverski izuzetak (npr. zbog pristupa delu memorije koji nije dozvoljen) ili “tiho” nepredviđeno ponašanje (bag)
- ❖ Pokazivač ne sadrži nikakvu dodatnu informaciju osim adrese. Za vreme izvršavanja ne vrše se nikakve dodatne provere validnosti vrednosti pokazivača. Zbog toga su pokazivači veoma osetljivi na ozbiljne probleme u programu i programer je odgovoran da obezbedi ispravnost pokazivača i njihove upotrebe. Ovo je jedan od najosetljivijih delova jezika C i C++

Pokazivači na objekte

Prema tome, pokazivači na jezicima C i C++ su veoma nebezbedni i mogu da uzrokuju veoma neprijatne greške. Evo nekih tipičnih problema sa pokazivačima:

- ❖ Neispravan (invalidan) pokazivač:

- Uzrok: pristup do objekta preko pokazivača koji ima invalidnu vrednost (predstavlja adresu na kojoj se ne nalazi objekat datog tipa); ni pokazivač ni objekat na koga on ukazuje se ne proveravaju za vreme izvršavanja:

```
Clock* p;  
p->tick();
```

- Efekti:
 - moguća greška za vreme izvršavanja, npr. izuzetak koji podiže hardver i prosleđuje operativni sistem, zbog nelegalnog pristupa delu memorije, tipično završava gašenjem programa
 - nedefinisano ponašanje ("tiha" greška): program nastavlja dalje sa potpuno nedefinisanim ponašanjem i na kraju možda negde kasnije izaziva izuzetak
- Tipične situacije u kojima nastaje:
 - neinicijalizovan pokazivač: gornji primer