
Pokazivači na objekte

- ❖ Pokazivač može da ne ukazuje ni na šta; vrednost takvog pokazivača naziva se *null* vrednost
- ❖ Simbolička vrednost *null* označava se kao *nullptr*; pre njegovog uvođenja, kao simbolička vrednost *null* koristio se:
 - literal *0*, sa simboličkom vrednošću *null* za pokazivače
 - makro *NULL* defnisan u standardnom zaglavlju jezika C
- ❖ Ove stare simboličke vrednosti mogu i dalje da se upotrebljavaju, ali one mogu da dovedu do drugačijeg tumačenja tipa u različitim kontekstima (kao celobrojni literal *0*), dok je *nullptr* različitog tipa
- ❖ Pokazivač se može:
 - inicijalizovati vrednošću *null* ili mu se dodeliti ta vrednost:

```
Clock* pc1 = nullptr;
```

- porediti na jednakost/nejednakost sa tom vrednošću

```
if (pc1==nullptr)...
```

```
if (pc1!=nullptr)...
```

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++