

# Pokazivači na objekte

❖ Dereferenciranje *null* vrednosti pokazivača:

- Uzrok: pristup do objekta preko pokazivača koji ima *null* vrednost; vrednost se ne proverava za vreme izvršavanja prilikom dereferenciranja:

```
Clock* pc = nullptr;  
pc->tick();
```

Greška za vreme izvršavanja

- Efekti:
  - po pravilu greška za vreme izvršavanja, npr. izuzetak koji podiže hardver i prosleđuje operativni sistem (zbog nelegalnog pristupa delu memorije, tipično završava gašenjem programa)
- Tipične situacije u kojima nastaje:
  - neka funkcija koja treba da vrati pokazivač na objekat koji se zahteva, može i da ne vrati to, jer ne može da uradi to što se od nje traži, što je regularna situacija, ili je neregularna, ali funkcija ne podiže izuzetak (a trebalo bi):

```
Clock* pc = ClockFactory::getClock();  
...  
pc->tick();
```

Ova funkcija možda vraća *null*

# Pokazivači na objekte

- Zaštita:
  - dobro razlikovati situacije u kojima funkcije mogu da vrate *null* vrednost pokazivača; ako to nije regularna situacija, funkcija tada treba da baci izuzetak
  - ako to jeste regularna situacija, po povratku iz funkcije obavezno proveravati vrednost na *null*:

```
Clock* pc = ClockFactory::getClock();
```

```
...
```

```
if (pc) pc->tick();
```

- krajnje konzervativan stil programiranja (u žargonu se naziva i “paranoičan”) pretpostavlja da se nikada ne veruje pokazivačima i da se oni uvek proveravaju pre dereferenciranja na bilo koji način; ovo nije potrebno ako se dobro tretiraju izuzeci:

```
if (p) ...p->...
```

```
if (p) ...*p...
```