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…
```

Pokazivači na objekte

- *Viseći* (ili "landaravi") pokazivač (*dangling pointer, dangling reference*):
  - Uzrok: pokazivač koji je bio vezan za objekat, ali je objekat prestao da postoji; takav pokazivač ima invalidnu vrednost, iako je ona bila validna; kada objekat prestane da postoji, to nema nikakve efekte na pokazivače koji na njega ukazuju:

```
Clock* pc = new Clock(…);
delete pc;
pc->tick();
```

  - Efekti:
    - nedefinisano ponašanje ("tiha" greška): program nastavlja dalje sa potpuno nedefinisanim ponašanjem i na kraju možda negde kasnije izaziva izuzetak
    - korupcija drugih podataka koji su zauzeli mesto tog objekta
  - Tipične situacije u kojima nastaje:
    - nekorektno postupanje sa pokazivačima:

```
Clock* p = new Clock(…);
```

U nekom potpuno drugom kontekstu:

```
Clock* q = p;
```

U nekom potpuno trećem kontekstu:

```
delete p;
```

I potom, opet u nekom potpuno različitom kontekstu:..

...`*q`... ili ...`q->`...