

Nasleđivanje

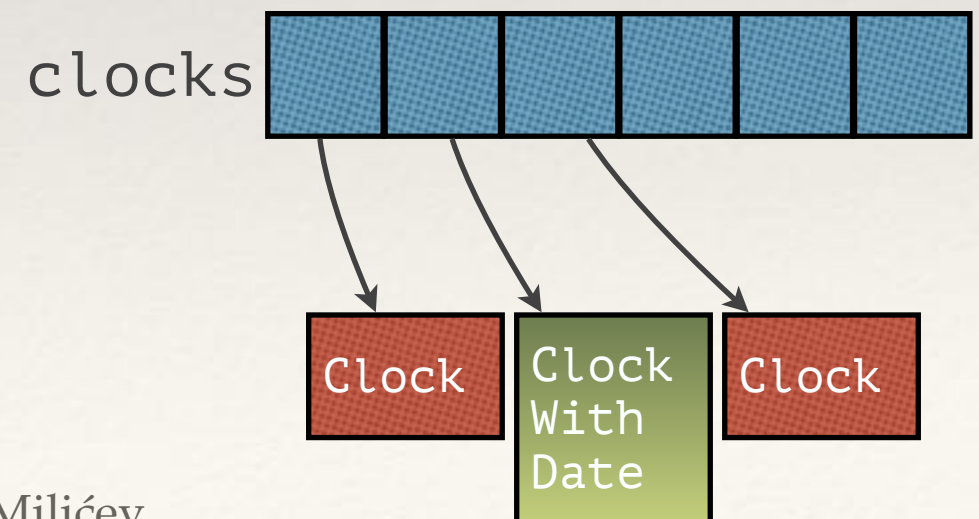
- ❖ Objekti izvedene klase su (indirektne) instance i osnovne klase!
- ❖ *Pravilo supstitucije* (*Substitution rule*, B. Liskov): gde god i kad god se očekuje objekat osnovne klase, može se pojaviti i objekat izvedene klase, jer
 - je objekat izvedene klase takođe i primerak (instanca) osnovne klase, jer
 - se sa njim može raditi sve što i sa objektima osnovne klase, jer
 - objekat izvedene klase ima (nasleđuje) sve osobine osnovne klase.
- ❖ Konverzija koju dozvoljava C++ i to implicitno:

DerivedClass* → BaseClass*

ClockWithDate* → Clock*

Pravilo supstitucije i implicitna konverzija
*ClockWithDate** u *Clock**

```
Lobby::Lobby (unsigned n, string c[], int lg[], int dateYN[]) {  
    ...  
    for (int i=0; i<num; i++) {  
        ...  
        if (dateYN[i])  
            clocks[i] = new ClockWithDate(1970,1,1,h,0,0);  
        else  
            clocks[i] = new Clock(h,0,0);  
    }  
}
```



Nasleđivanje

ClockWithDate

- ❖ Objekat izvedene klase ima sva svojstva osnovne klase — u sebi ima ugrađen *podobjekat* osnovne klase
- ❖ Za inicijalizaciju podobjekta osnovne klase, odnosno nasleđenih svojstava, odgovoran je konstruktor te osnovne klase
- ❖ Kada se kreira objekat izvedene klase, poziva se konstruktor te (izvedene) klase. Taj konstruktor, pre izvršavanja svog tela, *uvek* poziva konstruktor osnovne klase:

```
ClockWithDate::ClockWithDate (int y, int m, int d,  
    int h, int min, int s)  
    : Clock (h,min,s) {  
    setDate(y,m,d);  
}
```

```
Clock* clk = new ClockWithDate(...);
```