

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);  
}
```

Poziv konstruktora osnovne klase kome se prosleđuju neki dobijeni argumenti

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

Poziv konstruktora izvedene klase (koji onda poziva konstruktor osnovne klase)

Polimorfizam

- ❖ Objekti klase *ClockWithDate* moraju drugačije da reaguju na
 - operaciju *tick*, jer moraju da paze i na ažuriranje datuma
 - operaciju *getTime*, jer želimo da vrate i datum i vreme
- ❖ Funkcija članica koja će u izvedenim klasama imati *drugačiju implementaciju* deklariše se u osnovnoj klasi kao *virtuelna* funkcija (*virtual*). Izvedena klasa može da dâ svoju definiciju virtuelne funkcije, ali i ne mora:

```
class Clock {  
public:  
    Clock (int hh, int mm, int ss);  
  
    virtual void tick (); // Tick a second  
    virtual string getTime (); // Returns current time  
    void setTime (int hh, int mm, int ss);  
  
    int getHour() { return h; }  
    int getMinute() { return m; }  
    int getSecond() { return s; }  
    bool isMidnight () { return (h==0 && m==0 && s==0;) }  
  
private:  
    int h, m, s; // Hour, minute, second  
  
};
```