
Nasleđivanje

- ❖ Želimo novu apstrakciju, časovnika sa datumom
- ❖ Časovnik sa datumom je *jedna vrsta* časovnika, *specijalizacija* ili *proširenje* apstrakcije časovnika
- ❖ Klasa *ClockWithDate* nasleđuje (*inherits*) klasu *Clock*:

```
class ClockWithDate : public Clock {  
public:  
    ClockWithDate (int y, int m, int d, int h, int min, int s);  
    string getDate (); // Returns current date  
    void setDate (int yy, int mm, int dd); // Year, month, day  
  
private:  
    int y, m, d; // Year, month, day  
};
```

Nasleđivanje

- ❖ Ova relacija naziva se *nasleđivanje* (*inheritance*): klasa *ClockWithDate* nasleđuje sve osobine (atribute, operacije) osnovne klase *Clock*, ali je i *specijalizuje* (*specializes*) ili *proširuje* (*extends*), dodavanjem osobina (atributa, operacija), koje osnovna klasa nema
- ❖ Terminologija:
 - *Osnovna klasa* (*base class*), *generalizacija* (*generalization*), *roditelj* (*parent*)
 - *Izvedena klasa* (*derived class*), *specijalizacija* (*specialization*), *dete* (*child*)
- ❖ Sada se sa objektima izvedene klase *ClockWithDate* može raditi sve što i sa objektima osnovne klase *Clock*, jer su oni *jedna vrsta* (*kind of*) objekata osnovne klase, pošto poseduju (nasleđuju) sve osobine osnovne klase
- ❖ Sa objektima izvedene klase može se raditi i ono što je specifično za tu izvedenu klasu, a nije svojstveno osnovnoj:

```
Clock* simpleClock = new Clock(13,17,0);
ClockWithDate* smartClock = new ClockWithDate(2018,9,13,13,17,0);

simpleClock->setTime(13,20,0);
smartClock->setTime(13,20,0);

simpleClock->setDate(2018,9,14);
smartClock->setDate(2018,9,14);
```