

## Увод у објектно оријентисано програмирање

У **процедуралном програмирању**, програми се састоје од функција, а подаци се представљају засебно. Сложене интеракције програмског кода и података из различитих делова програма такав програм чине сложеним и тешким за одржавање, па је за реализацију сложених софтверских система потребан другачији приступ.

**Објектно оријентисано програмирање** је окренуто ка подацима, а не ка функцијама (процедирама). Објектно оријентисани приступ сложене софтверске системе представља као скуп објеката, који се састоје од података и метода којима се врше операције над објектима. Овај приступ организује програме на начин како је организован стварни свет, где су предмети у међусобној вези, како по атрибутима, тако и по активностима.

Објектно оријентисано програмирање (ООП) је има значајне **предности** у односу на класично (процедурално) програмирање.

- 1 . Код писан у ООП је лакши за одржавање и унапређивање, јер није потребно мењати цео код, већ само класе и методе које се мењају.
2. Могу да се моделују сложеније структуре и решавају сложенији проблеми него код процедуралног програмирања.

### Својства објектно оријентисаног програмирања и програмских језика

- **Апстракција** (abstraction) - могућност дефинисања нових типова података (класе објеката)
- **Енкапсулација** (encapsulation) - скривање детаља реализације неког типа (класе)
- **Наслеђивање** (inheritance) - креирање нових типова (класа) помоћу постојећих, који ће наследити све особине старих типова и додати своје специфичности
- **Полиморфизам** (polymorphism) - појављивање типа (класе) у више облика, јер се могу не само додавати своји елементи, већ и мењати наслеђени