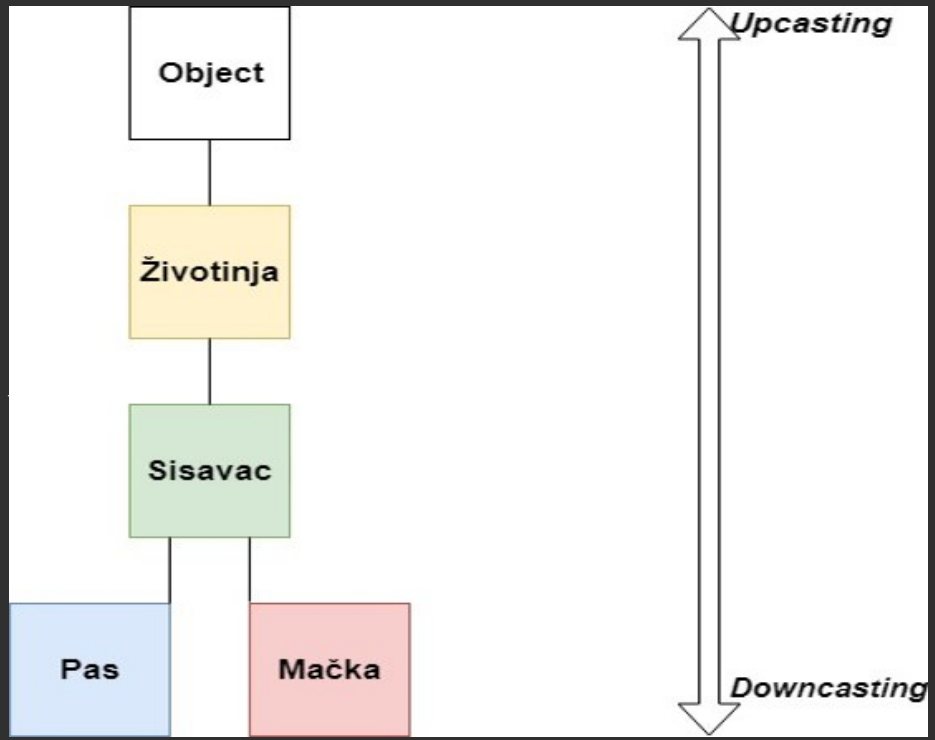
# Uvod u objektno orijentirano programiranje

## Nasljeđivanje i polimorfizam

# Nasljeđivanje

```
class Zivotinja {
    int zdravlje = 100;
}
class Sisavac extends Zivotinja { }
class Macka extends Sisavac { }
class Pas extends Sisavac { }
public class Test {
        public static void main(String[] args)
         Macka c = new Macka();
         System.out.println(c.zdravlje);
         Pas d = new Pas();
         System.out.println(d.zdravlje);
         }
}
```



# Upcasting

```
Macka c = new Macka();
System.out.println(c);
Sisavac m = c; // upcasting
System.out.println(m);
    /*
    Izlaz:
        Macka@a90653
        Macka@a90653
    */
```

- Ne mijenjamo sam objekt već ga samo „označujemo” drukčije
- Dozvoljeno jer je svaka Macka Sisavac
- Ručno podizanje(Upcasting) je dozvoljeno ali nepotrebno
- Sisavac m = (Sisavac) new Macka(); jednako je Sisavac m = new Macka();

# Downcasting

Za razliku od upcasting-a, downcasting se uvijek radi ručno (Svaka Macka je Zivotinja, ali nije svatko tko je naslijedio klasu Zivotinja Macka

```
Macka c1 = new Macka();
Zivotinja a = c1;
Macka c2 = (Macka) a;
```

Primjer krivog korištenja

```
Sisavac m = new Sisavac();
Macka c = (Macka)m;
```

Prilikom izvršenja javlja se greška "java.lang.ClassCastException: Sisavac cannot be cast to Macka"

# Runtime polimorfizam

```
class Sisavac extends Zivotinja {
    public String speak(){
        return '';}
}
class Macka extends Sisavac {
    public String speak(){
        return 'Meow!';}
}
class Pas extends Sisavac {
    public String speak(){
        return 'Woof!';}
 }

 Macka c1 = new Macka();
 Sisavac a = c1;
 a.speak();
```


Što bi bio compile time polimorfizam (static polimorfizam)?

Zadatak 1.
Definirajte klasu Trkac koja će poslužiti za unos natjecatelja na maratonu "Homo si teć". Klasa Trkac će ovisno o daljnjoj implementaciji sadržavati barem:

- varijable za ime, prolazno vrijeme, prijeđeni put i osvojenu nagradu
- konstruktor kojim se zadaju ime i prolazno vrijeme na utrci
- metodu brzina koja ima prazan argument te vraća brzinu trkača
- metodu signature public void dodjela() s praznom implementacijom koja će biti prerađena u podklasama

Varijablu za unos imena i prolaznog vremena je potrebno zaštiti od naknadne izmjene

Zadatak 2:
Napravite klasu Maratonac koja:

- nasljeđuje klasu Trkac
- u konstruktoru definira varijablu prijeđeni put na 42 km
- prerađuje metodu dodjela() tako da u odgovarajuću varijablu upisuje zlato instanci najbržeg među Maratoncima

Zadatak 3.
Napravite klasu Revijalac koja:

- nasljeđuje klasu Trkac
- u konstruktoru definira varijablu prijeđeni put na 5 km
- prerađuje metodu dodjela() tako da u odgovarajuću varijablu upisuje nagrada instanci onih Revijalaca čija brzina premašuje prosječnu brzinu svih trkača zajedno (Maratonaca i Revijalaca)

Napisati Testnu klasu koja će napraviti listu Trkaca, unijeti u tu listu nekoliko Maratonaca i Revijalaca te demonstrirati upotrebu traženih metoda.