

**Sveučilište u Zagrebu**  
**PMF – Matematički odjel**



# **Objektno programiranje (C++)**

Vježbe 05 – Nasljeđivanje

**Matej Mihelčić**

# Nasljeđivanje

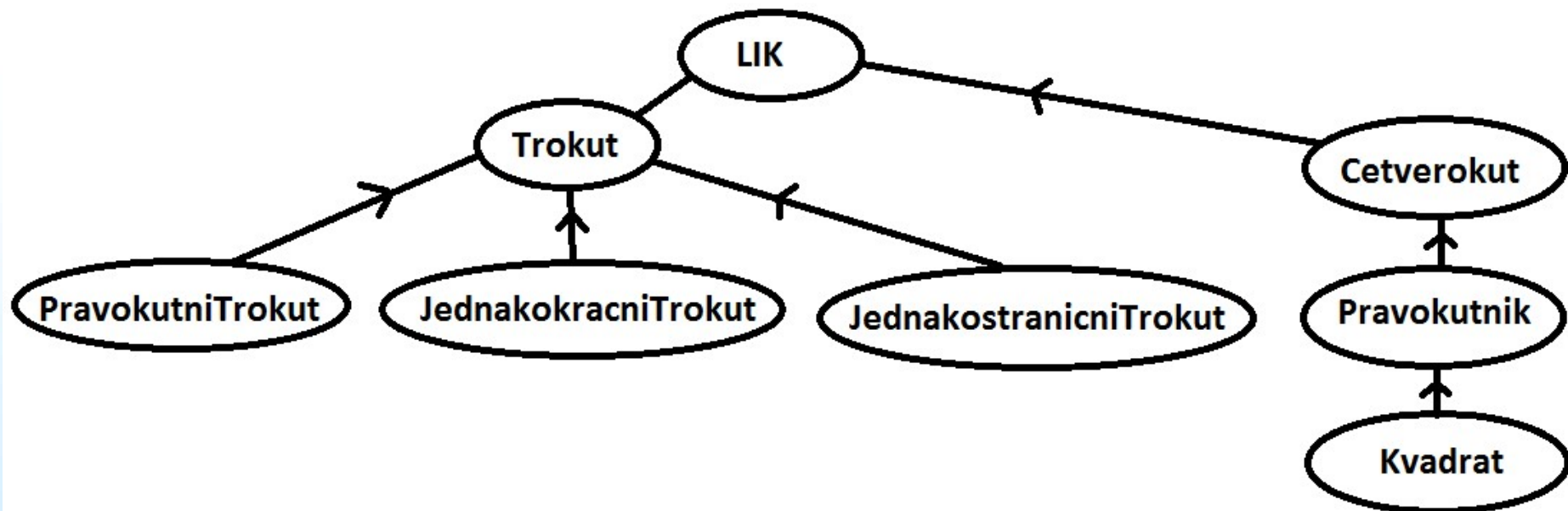
- Dosta često objekti dijele neka svojstva. Npr. živa bića su složeni organizmi, sposobni reagirati na podražaje, sposobni za razmnožavanje, prilagodbu, rast, razvoj itd. Međutim, postoji velika raznolikost u živom svijetu.
- Da bi izbjegli ponavljanje dijeljenih karakteristika unutar klasa koje reprezentiraju raznovrsne ali povezane objekte, koristimo nasljeđivanje.
- Nasljeđivanjem u jeziku C++, klasa dobiva sve **javne** i **protected** članove klase koju nasljeđuje. To uključuje i metode, čije ponašanje specifičnija klasa može i promijeniti. Npr. metodu **rast** bi trebali izmijeniti ovisno o tome o kakvom živom biću je riječ pošto postoje razlike u načinu i u trajanju rasta organizma.
- Osim što sami elementi članovi klase imaju identifikator dopuštenja pristupa (public – dostupno svuda, protected – dostupno unutar klase ili iz klasa koja nasljeđuju klasu, private – dostupno samo unutar klase), tako i sam **proces nasljeđivanja** može biti ostvaren **koristeći te iste identifikatore**.
- **Koristeći public nasljeđivanje**, svi identifikatori pristupa ostaju isti (naslijeđeni public elementi ostaju public u klasi koja nasljeđuje, naslijeđeni protected elementi ostaju protected, private elementi su nedostupni klasi koja nasljeđuje).
- **Koristeći protected nasljeđivanje**, naslijeđeni public elementi **postaju protected** u klasi koja nasljeđuje.
- **Koristeći private nasljeđivanje**, naslijeđeni public i protected elementi **postaju private** u klasi koja nasljeđuje.
- Razlog dodavanja takvih ograničenja pri nasljeđivanju je što neka **klasa koja se nasljeđuje** možda treba samo kao

**pomoć pri internoj implementaciji**, stoga nije potrebno ili poželjno da takvi elementi budu dostupni izvan klase.

# Nasljeđivanje

## Zadatak:

- Implementirajte hijerarhiju klasa kao na slici. Klasa lik je apstraktna i sadrži apstraktne funkcije za računanje opsega i površine lika, kao i funkciju za ispis lika. Klase trokut i četverokut sadrže duljine stranice i kuteve između tih stranica. Možemo pretpostaviti da su stranice uvijek sortirane u uzlaznom poretku, a kutevi identificirani parom indeksa stranica. Napišite konstruktore koji stvaraju lik ukoliko su zadani svi kutevi i sve stranice. Kod trokuta, dvije stranice i kut među njima, tri stranice, dva kuta i zajednička stranica. Kuteve ćemo reprezentirati kao realne brojeve.



# Polimorfizam

## Zadatak:

- Što se događa ukoliko napravimo sljedeće?

Lik \*l;

Cetverokut c;

Trokut t;

l = &c; .

l->ispis();

l = &t;

l->ispis();

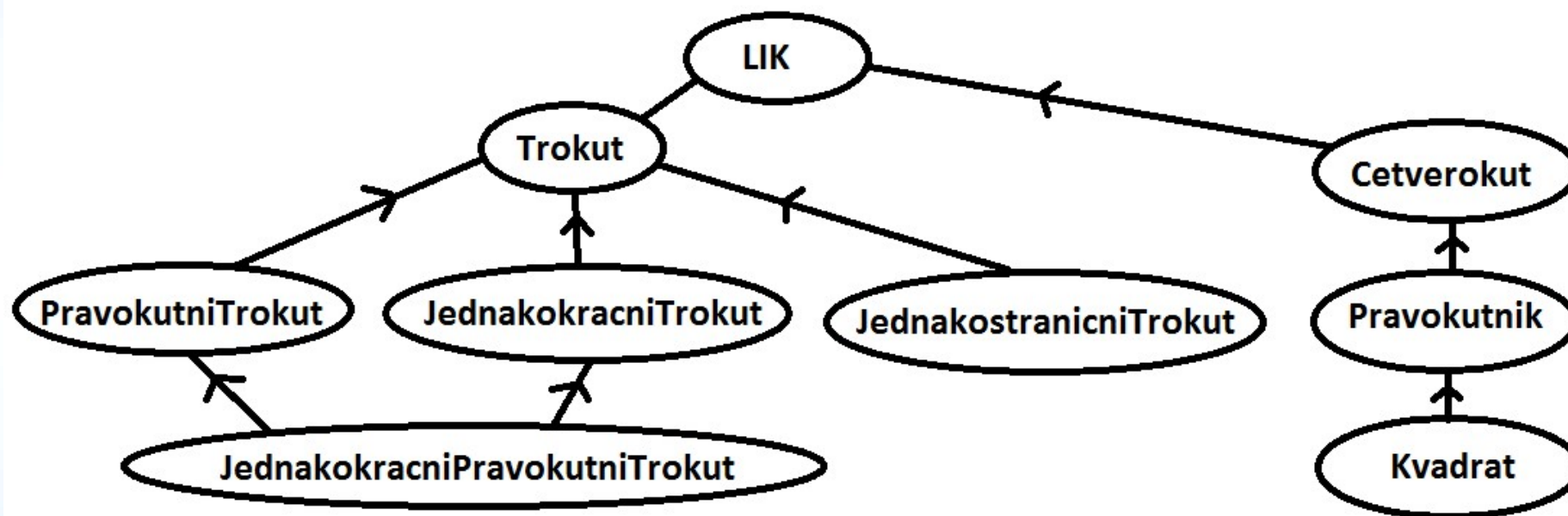
Modificirajte program tako da se pozove funkcija ispis() **odgovarajuće klase**.

- Stavite sve kreirane objekte podklase klase Lik u jedan vektor i ispišite ukupnu površinu svih likova.

# Zadatak

## Zadatak:

- Modificirajte program tako da izmijenjena hijerarhija ispravno radi.



# Zadatak

## Zadatak:

Napravite hijerarhiju generičkih klasa koje reprezentiraju uređeni par (imena Par), te njegovo proširenje BrojeciPar. BrojeciPar ima po jedan brojac za svaku komponentu para. Takva struktura može služiti za brojanje broja pobjeda u međusobnim susretima, brojanje dugovanja, razmijenjenih poruka i slično. Implementirajte operacije dohvaćanja i pristupa određenoj komponenti klase, definirajte operatore zbrajanja, oduzimanja, množenja i dijeljenja uređenog para. Implementirajte operator ispisa Para i BrojecegPara. Definirajte operatore inkrementa brojaca obje komponente BojecegPara. Predefinirajte prefiks i postfix increment tako da prefiks inkrementira prvu komponentu Para a postfix drugu komponentu. Operator prefiks inkrement inkrementira i prvi brojac ukoliko se radi o BrojecemParu a postfix inkrementira drugi brojac. Koristeći strukturu BrojeciPar, za zadane znakove x i y, pretražite datoteku „pjesma.txt” te kod prvog pronalaska znaka x ili y, postavite taj znak kao prvu komponentu para, kod prvog pronalaska preostalog znaka, postavite taj znak kao drugu komponentu para. Kod svakog sljedećeg pronalaska znaka, inkrementirajte brojač odgovarajućeg znaka za jedan. Ispišite konačan BrojeciPar.