

# **Nasljeđivanje i hijerarhija klasa**

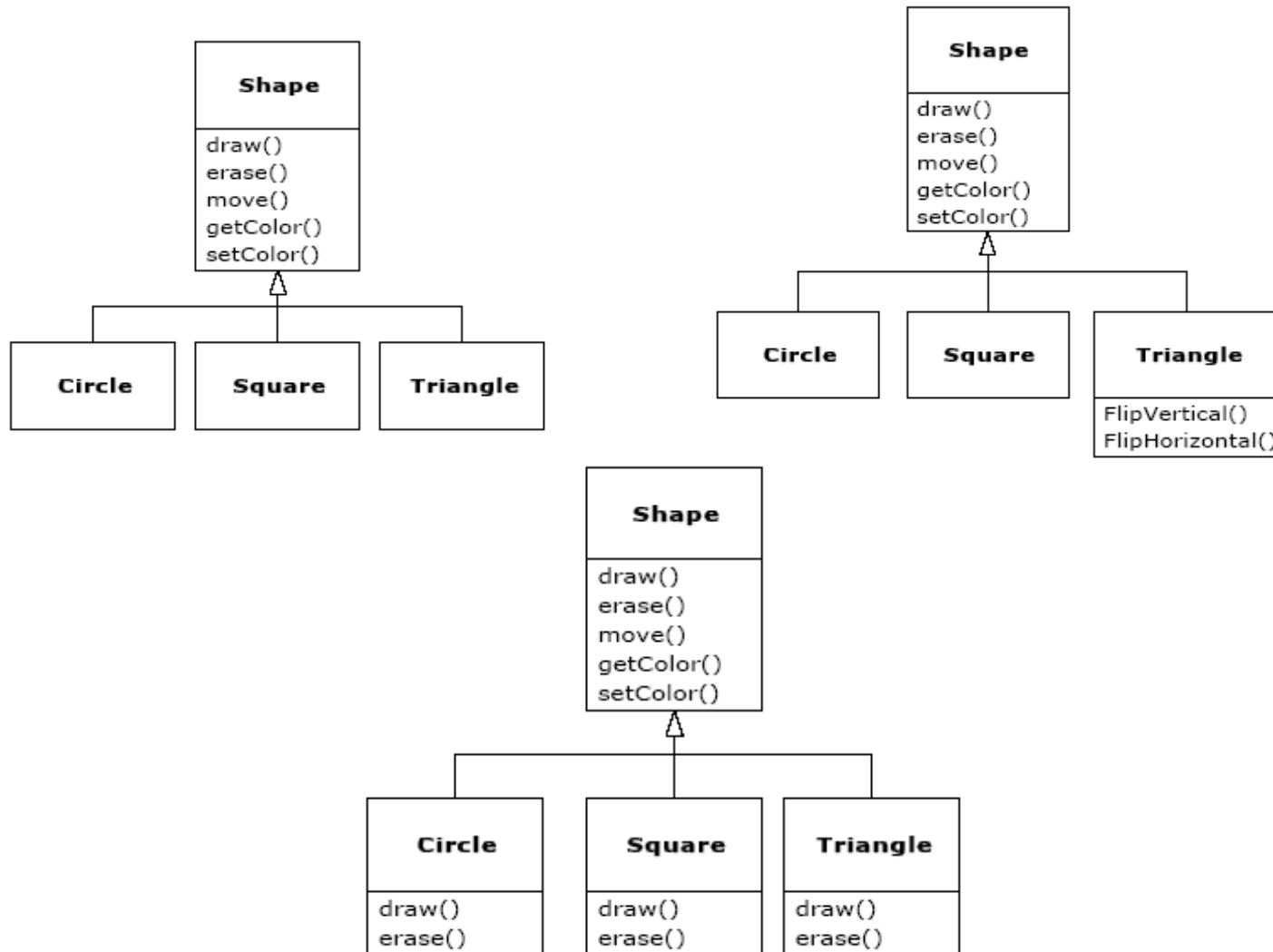
# Apstraktna (bazna) klasa

- Osnovna klasa koja **definira zajednička svojstva** svih objekata bez navođenja kako se pojedina operacija mora obaviti.
- Postoji samo zato da bi ju se moglo naslijediti
- Ostale klase će naslijediti tu klasu te će uvesti i **nova svojstva** potrebna za opisivanje objekata i/ili **redefinirati postojeća** u skladu sa samim objektom.

# Izvedene klase (derivirane)

- Definirane na osnovu postojećih (baznih) klasa
- Objekti izvedenih klasa **sadrže sve članove osnovne (bazne) klase, a može im se dodavati i nova svojstva.**
- U izvedenoj klasi moguće je zamijeniti (zaobići) željene funkcijske članove bazne klase novima - **overriding**

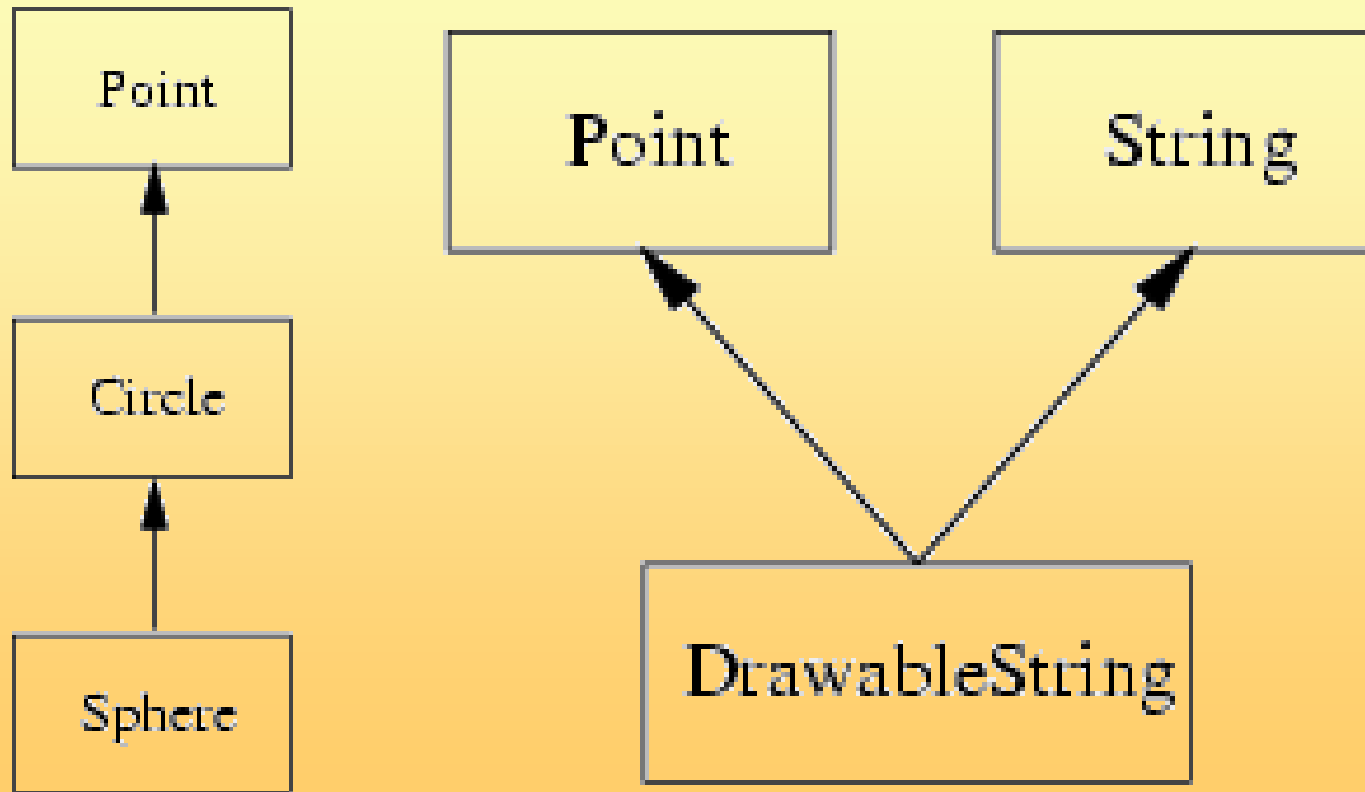
# Primjeri



# Nasljeđivanje

- Nasljeđivanje se određuje tako da se **u deklaraciji klase (iza naziva klase) navede lista baznih klasa** (nazivi) odvojeni zarezom
- Ispred svakog od naziva moguće je staviti neku od ključnih riječi: **public, private ili protected** čime se navodi **tip nasljeđivanja**
- Klasa **može naslijediti i nekoliko drugih klasa**. U tom su slučaju svi podatkovni i funkcijski članovi svih baznih klasa uključeni u novu klasu.
- Ovakav tip nasljeđivanja kada jedna izvedena klasa ima više osnovnih zove se **višestruko nasljeđivanje (multiple inheritance)**

# Primjer:



# Nasljeđivanje i prava pristupa

- Prava pristupa određuju kako objekti komuniciraju s okolinom (npr. programerima koji će izvoditi nove klase iz postojećih treba sakriti osjetljive informacije)
- Tri prava pristupa: **javni (public)**, **privatni (private)** i **zaštićeni (protected)**
- Članovima s javnim pristupom može se pristupiti i izvan klase, dok se članovima s privatnim i zaštićenim pristupom može pristupiti isključivo unutar klase
- Pravo pristupa članu u izvedenoj klasi određeno je pravom pristupa u osnovnoj klasi i vrstom nasljeđivanja pa tri prava pristupa i tri moguća tipa nasljeđivanja daju **devet** mogućih kombinacija.

# Zaštićeno (protected) pravo pristupa

- Primarno služi stvaranju sučelja za nasljeđivanje, određuje članove koji su namijenjeni korištenju u deriviranim klasama
- Članovima određenim kao *protected* ne može se pristupiti izvan klase (npr. pomoću objekta te klase), dakle nisu javno dostupni, no moguće ih je koristiti u funkcijskim članovima derivirane klase
- Privatni članovi nisu dostupni u izvedenim klasama
- Javni članovi su dostupni i korisnicima objekta



# Javna bazna klasa (*public base class*)

- Određena klasa je javna bazna klasa ako je u listi prilikom nasljeđivanja navedena pomoću ključne riječi *public*
- Svi elementi bazne klase bit će uključeni u deriviranu klasu u kojoj će zadržati svoje originalno pravo pristupa
- Privatni članovi neće biti dostupni iz deriviranih klasa niti iz preostalog programa
- Zaštićenim članovima moći će se pristupiti iz derivirane klase, ali ne i iz glavnog programa. Zaštićeni članovi isključeni su iz javnog sučelja klase, ali su uključeni u sučelje namijenjeno nasljeđivanju.
- Javni članovi ostaju dostupni i iz glavnog programa i iz izvedene klase (dio su i javnog sučelja i sučelja namijenjenog nasljeđivanju)

# Privatna bazna klasa (*private base class*)

- Klasa je **privatna bazna klasa** ako se (u listi nasljeđivanja) navede s ključnom riječi `private` ili se ključna riječ koja određuje tip nasljeđivanja izostavi.
- Prilikom privatnog nasljeđivanja, privatni članovi bazne klase nisu dostupni izvedenoj klasi, dok javni i zaštićeni članovi bazne klase postaju privatni članovi izvedene klase.

# Zaštićena bazna klasa (*protected base class*)

- **Zaštićeno nasljeđivanje** određuje se tako da se prije naziva bazne klase navede ključna riječ **protected**.
- Time se svi javni i zaštićeni članovi prenose kao zaštićeni u izvedenu klasu.
- Sve izvedene klase mogu imati pristup članovima bazne klase, a ostatak programa tim članovima ne može pristupati.

# Pravila nasljeđivanja pri pristupu baznoj klasi

- Svi članovi `private` bazne klase postaju privatni članovi derivirane klase.
- Članovi `public` bazne klase zadržavaju njihove pristupne karakteristike u deriviranoj klasi.
- Privatni članovi `protected` bazne klase postaju privatni članovi derivirane klase, javni i zaštićeni članovi postaju zaštićeni članovi derivirane klase.

| Base Class       | Private Derived | Public Derived   | Protected Derived |
|------------------|-----------------|------------------|-------------------|
| Private Member   | <i>private</i>  | <i>private</i>   | <i>private</i>    |
| Public Member    | <i>private</i>  | <i>public</i>    | <i>protected</i>  |
| Protected Member | <i>private</i>  | <i>protected</i> | <i>protected</i>  |

# Nasljeđivanje

- jedna je klasa bazna, druga derivirana
- baznu klasu može temeljito testirati jedna skupina/osoba, a koristiti derivirana klasa koju je razvila neka druga skupina/osoba
- derivirana klasa može dodati nove mogućnosti baznoj klasi
- prilikom deklariranja objekta iz derivirane klase, automatski se kreira objekt bazne klase i sadržan je u objektu iz derivirane klase, a članovi bazne klase postaju i članovi derivirane klase

# Osnove mehanizma nasljeđivanja

- bazna se klasa može definirati bez da se u obzir uzima proces nasljeđivanja
- jedino što daje naslutiti da će biti bazna je specifikator pristupa `protected`.
- obično se bazne klase dizajniraju s namjerom da budu bazne za druge klase – bazna klasa mora biti neovisna i na siguran i efikasan način raditi i upravljati sa pristupom svojim podacima

# Specifikator pristupa `protected`

- Članovima bazne klase određenim kao `protected` može se pristupiti direktno preko funkcija deriviranih klasa i bazne klase. Zahtjeva pažljivo rukovanje.
- Članovima bazne klase određenim kao `private` može se pristupiti direktno samo preko funkcija bazne klase. Neki programeri preferiraju samo `private` pristup.

# Definiranje derivirane klase

- **Derivirana klasa mora naznačiti klasu koju nasljeđuje.**
- **Naznačuje se i oblik nasljeđivanja:**
  - public
  - private
  - protected
- **Najčešće je public nasljeđivanje**
- **Podatkovni članovi predstavljaju podatke koje derivirana klasa treba.**
- **Funkcijski članovi rade sa članovima i derivirane i bazne klase.**