JAVA 1

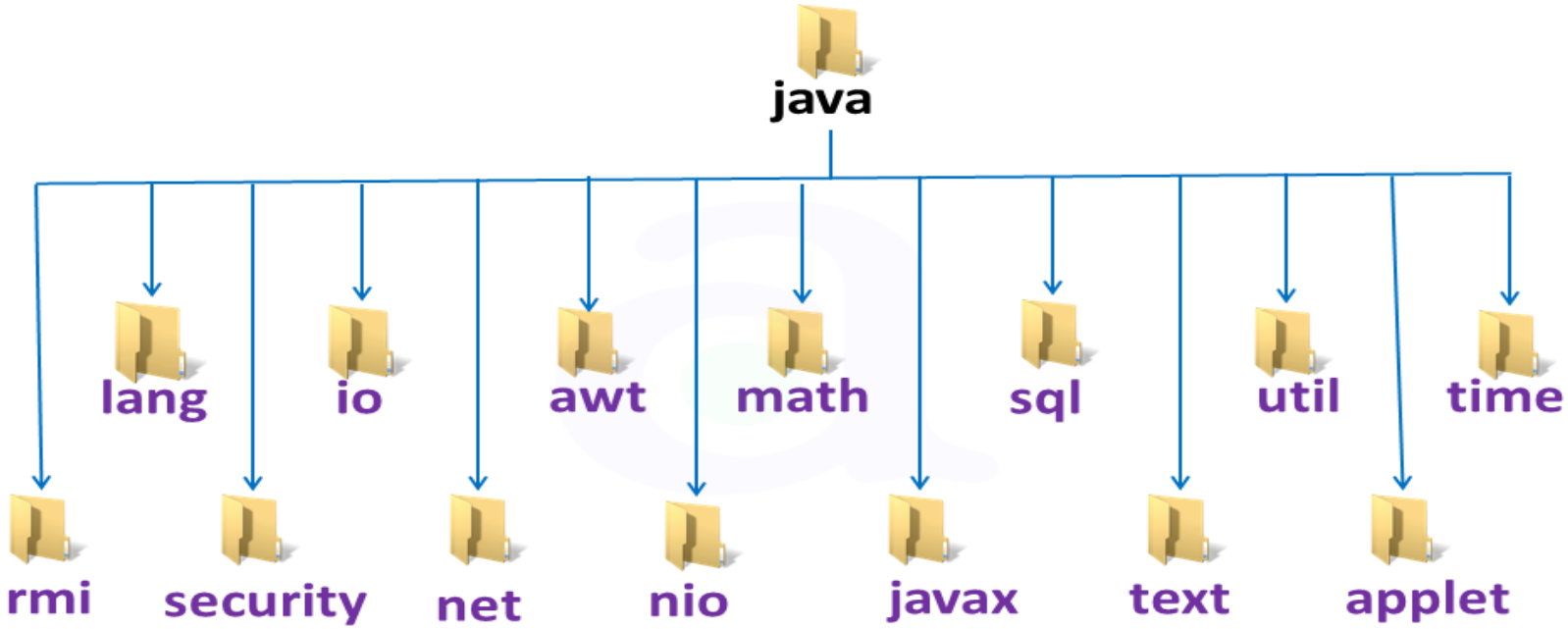
OOP

Teme:

- Packages
- Access modifiers
- Classes
- Wrapper classes - Autoboxing and Unboxing
- Abstract classes, Interfaces
- Encapsulation
- Implicit & explicit inheritance
- Polymorphism
- Static

Packages

- Svi entiteti moraju biti definirani u paketima, koji predstavljaju direktorije podatkovnog sustava



Izvor: https://www.atnyla.com/tutorial/package-introduction/0/79

Access modifiers

Access Levels

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Izvor: https://techdemic.com/java-modifiers/

Classes

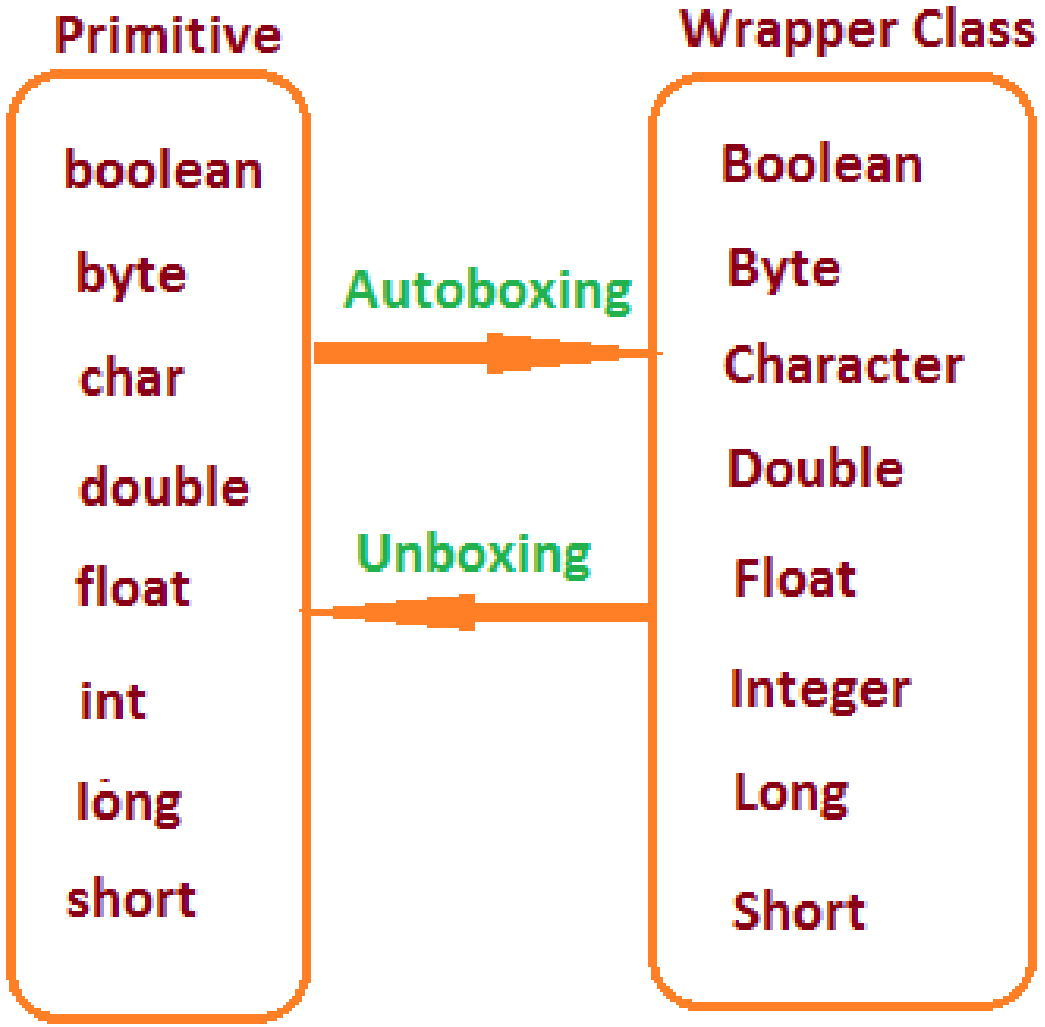
- *template* za kreiranje objekata
- ime *javne* klase mora biti jednako imenu datoteke
- može se *otvoriti* konstruktorom, ali i getterima i setterima
- referentni tip vs. podatkovni tip
- *stack vs. heap*
- final
- *Garbage collector*

Wrapper classes

- *Obuhvaća* primitivni tip
- Autoboxing

  Integer x = 5;
- Unboxing

  int sum = x + 1;



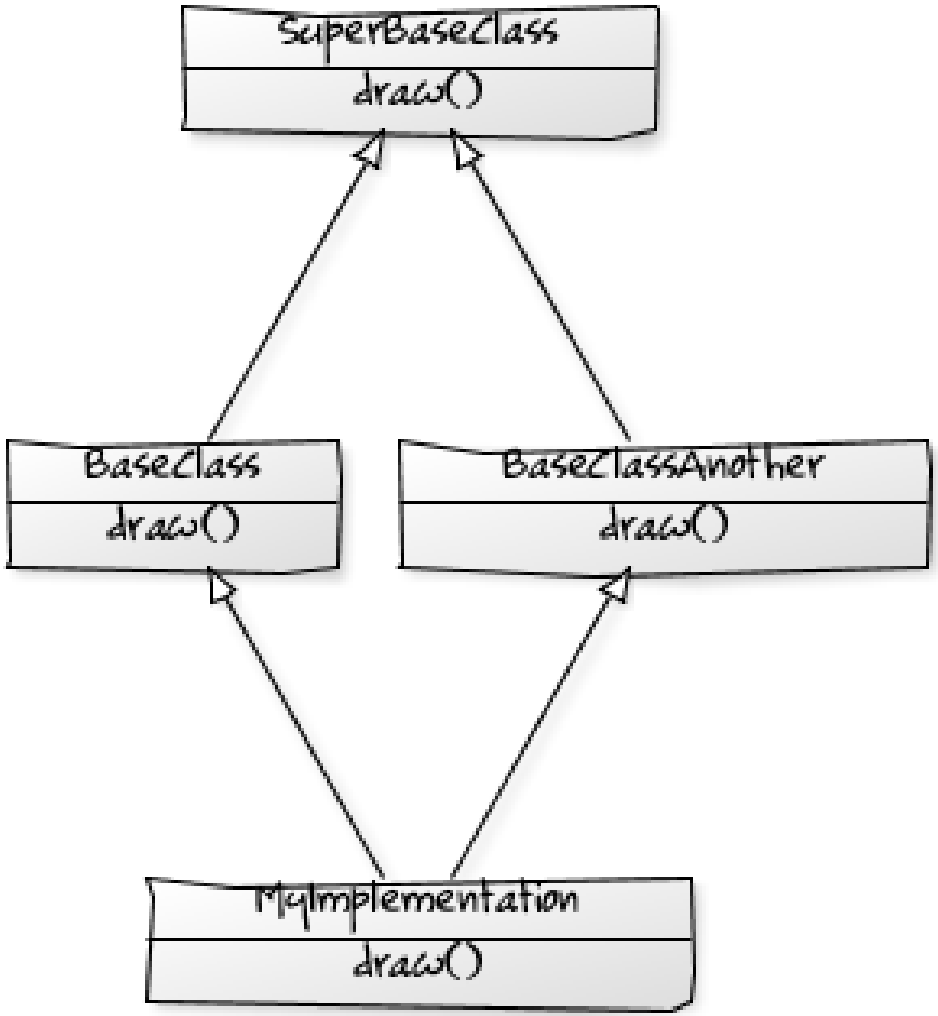
Izvor: http://www.javatechblog.com/java/autoboxing-and-unboxing-in-java-with-example/amp/

Abstract Classes

- Može sadržavati apstraktne, ali i konkretne metode
- Ako postoji jedna apstraktna metoda, klasa je apstraktna
- Ne može se inicijalizirati
- Nasljednici moraju implementirati apstrakcije ili su i sami apstrakcija – *contract*
- Apstraktna klasa ne mora implementirati metode Interface-a

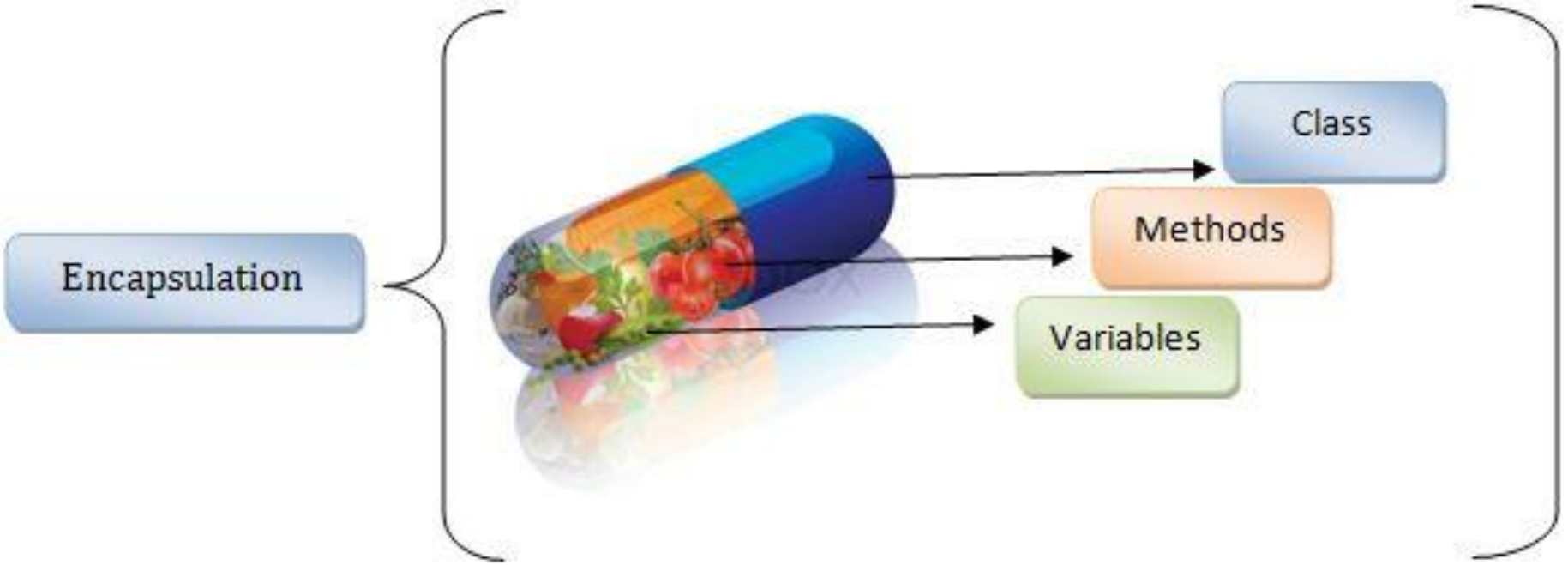
Interfaces

- Diamond problem
- Complete abstraction
- *public by default*



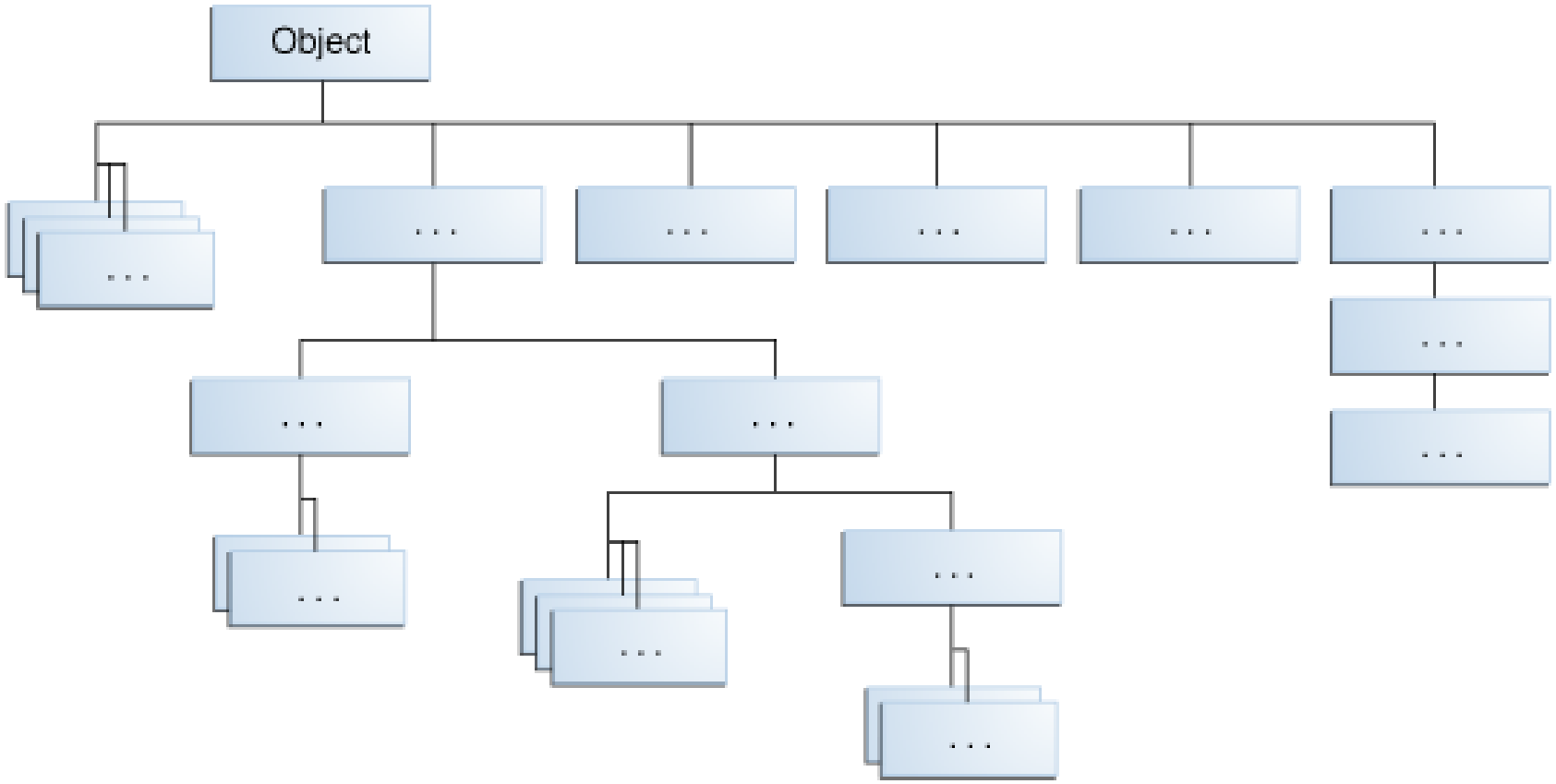
Izvor: https://solution-0.blogspot.com/2018/05/62-info-how-java-solve-diamond-problem.html

Encapsulation



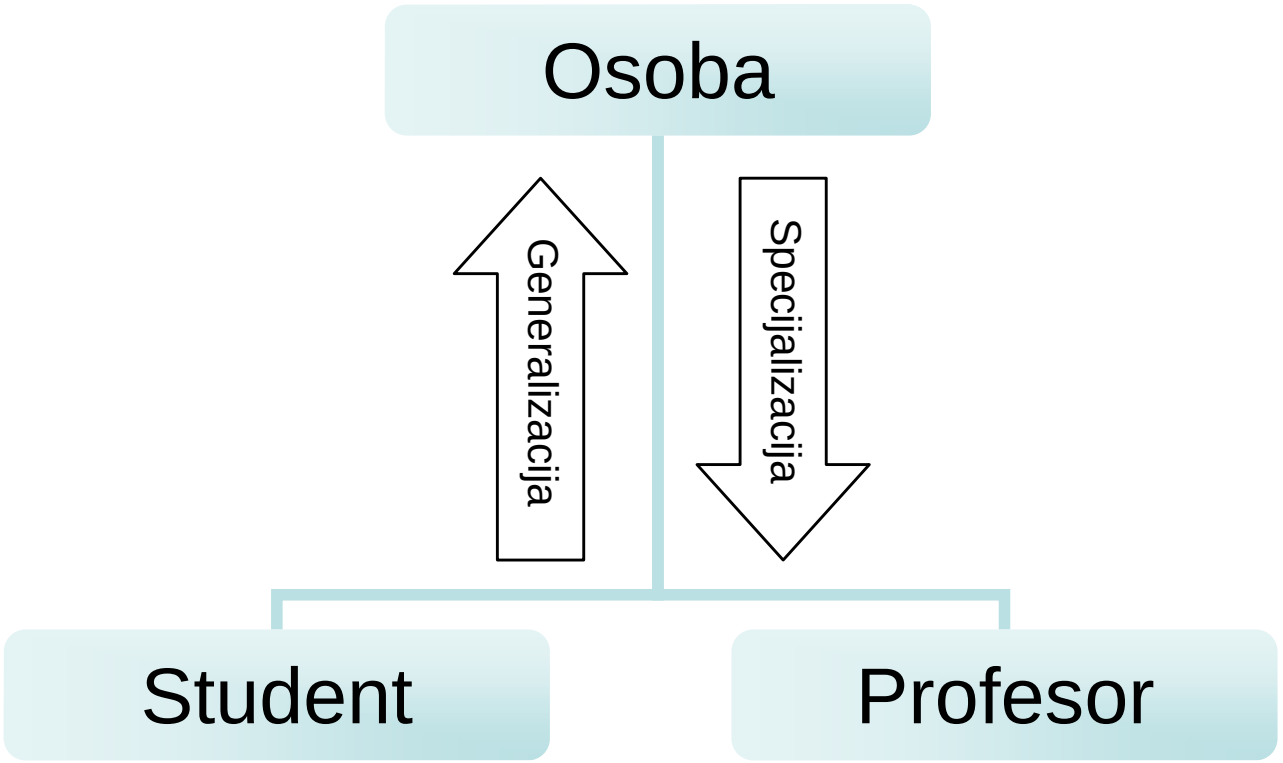
Izvor:http://www.rssofttech.com/encapsulation-in-java-how-to-master-oops-with-encapsulation/

Implicit inheritance



Izvor:https://docs.oracle.com/javase/tutorial/java/IandI/subclasses.html

Explicit Inheritance



Polymorphism

- Mogućnost objekta da poprimi više različitih oblika
- Objekti se mogu *promatrati* na različite načine u različitim kontekstima
- Promovira *code reuse* i *separation of concerns*

Static

- varijable, metode, blokovi, ugnježđene klase
- inicijalizacija prilikom učitavanja klase u virtualnu mašinu
- java.lang.Math

```
public final class Math {
    /**
     * Don't let anyone instantiate this class.
     */
    private Math() {}
    /**
}
```

- static import:

```
static import java.lang.Math.*
```

Demo

- Project

Izvor:http://www.jnhsolutions.com/contact-us/request-a-demo/

Hvala na pažnji!