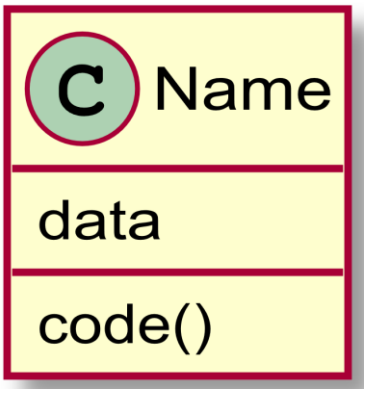
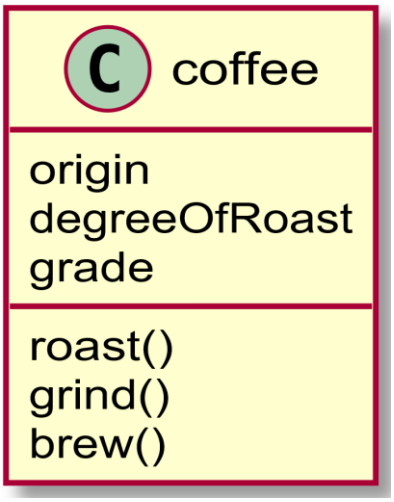
부록 5. PlantUML 코드와 그림

3 장

```
@startuml
' 주석(comment)
class Name {
    data
    code()
}
@enduml
```

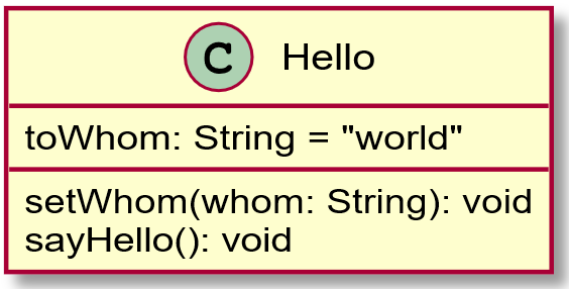



```
@startuml
class Coffee {
    origin
    degreeOfRoast
    grade
    roast()
    grind()
    brew()
}
@enduml
```

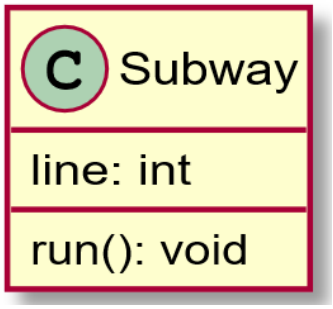


```
@startuml
class Hello {
  void sayHello();
}
@enduml
```


```
@startuml
class Hello {
  toWhom : String = "world"
  sayHello() : void
}
@enduml
```

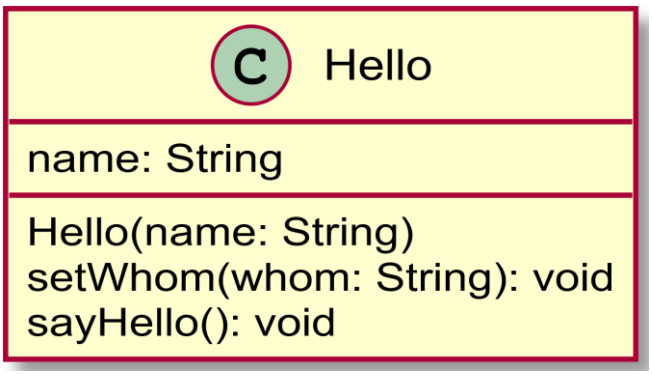



```
@startuml
class Subway {
  line: int
  run(): void
}
@enduml
```

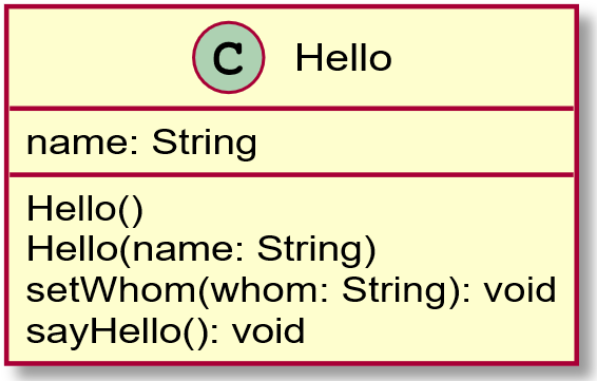



```
@startuml
class Hello {
    name: String
    Hello(name: String)
    setWhom(whom: String): void
    sayHello(): void
```

```
}
@enduml
```

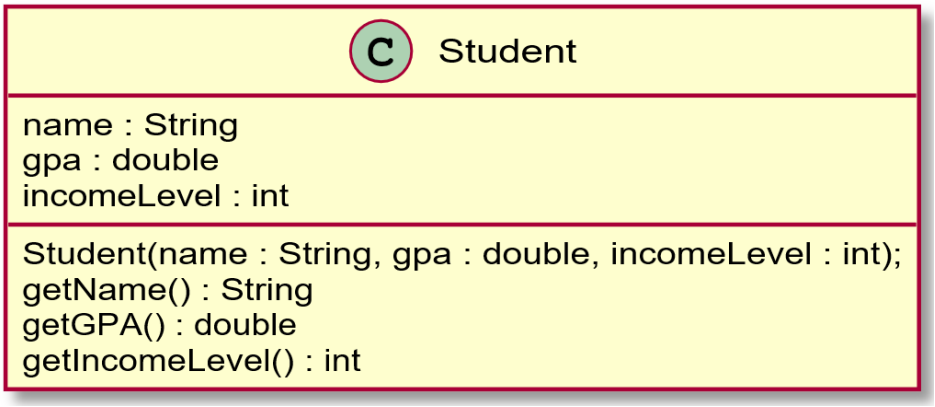



```
@startuml
class Hello {
    name: String
    Hello()
    Hello(name: String)
    setWhom(whom: String): void
    sayHello(): void
}
@enduml
```

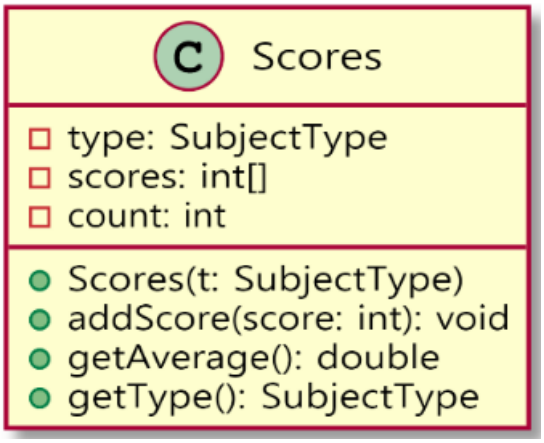


4 장

```
@startuml
class Student {
    name : String
    gpa : double
    incomeLevel : int
    Student(name : String, gpa : double, incomeLevel : int);
    getName() : String
    getGPA() : double
    getIncomeLevel() : int
}
@enduml
```

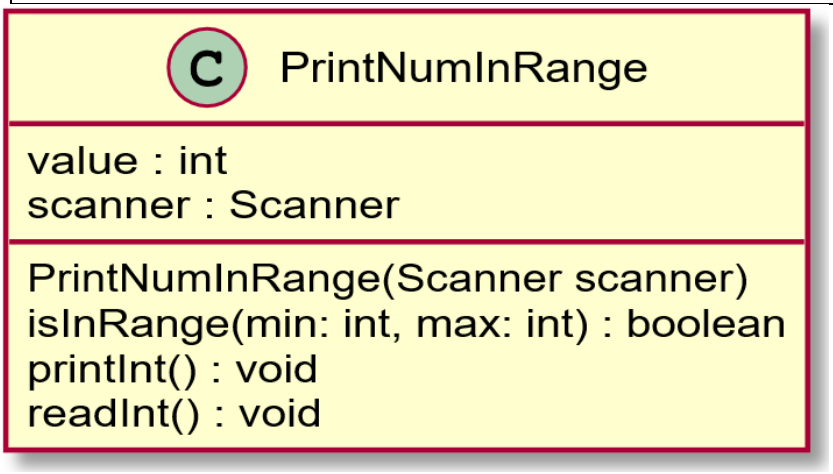



```
@startuml
class Scores {
    -type: SubjectType
    -scores: int[]
    -count: int
    +Scores(t: SubjectType)
    +addScore(score: int): void
    +getAverage(): double
    +getType(): SubjectType
}
@enduml
```

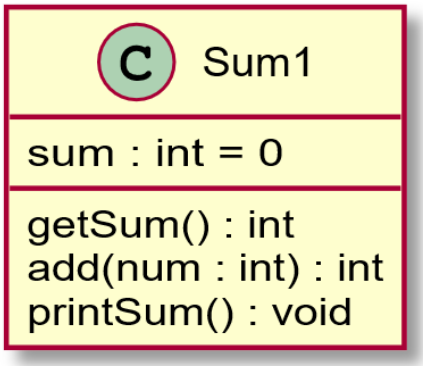


5 장

```
@startuml
class PrintNumInRange {
    value : int
    scanner : Scanner
    PrintNumInRange(Scanner scanner)
    isInRange(min: int, max: int) : boolean
    printInt() : void
    readInt() : void
}
@enduml
```

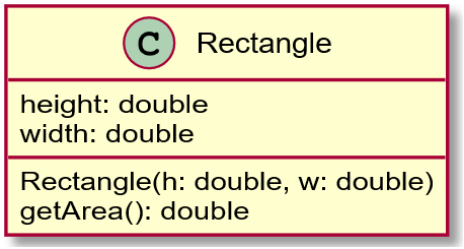



```
@startuml
class Sum1 {
    sum : int = 0
    getSum() : int
    add(num : int) : int
    printSum() : void
}
@enduml
```

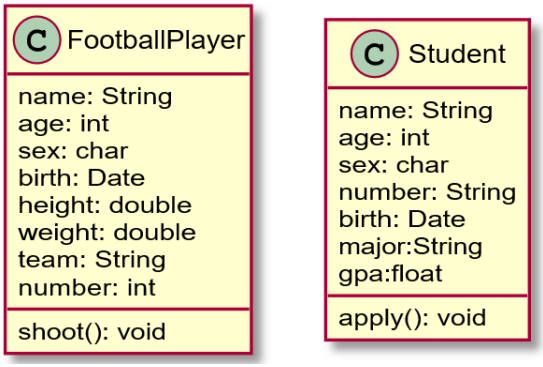


6 장

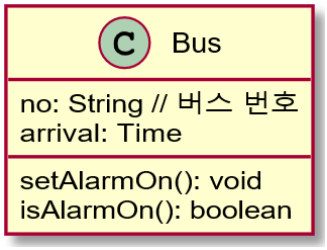
```
@startuml
class Rectangle {
    height: double
    width: double
    Rectangle(h: double, w: double)
    getArea(): double
}
@enduml
```



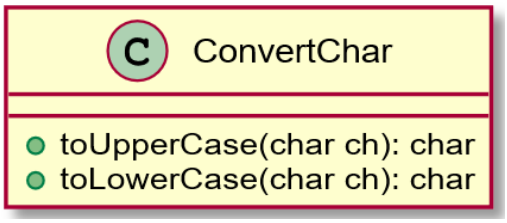

```
@startuml
class FootballPlayer {
    name: String
    age: int
    sex: char
    birth: Date
    height: double
    weight: double
    team: String
    number: int
    shoot(): void
}

class Student {
    name: String
    age: int
    sex: char
    number: String
    birth: Date
    major:String
    gpa:float
    apply(): void
}
@enduml
```



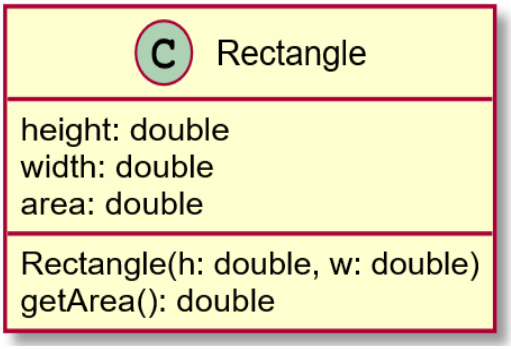
```
@startuml
class Bus {
    no: String  // 버스 번호
    arrival: Time
    setAlarmOn(): void
    isAlarmOn(): boolean
}
@enduml
```



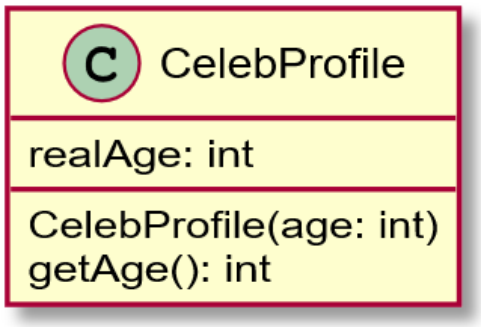

```
@startuml
class ConvertChar {
    +toUpperCase(char ch): char
    +toLowerCase(char ch): char
}
@enduml
```



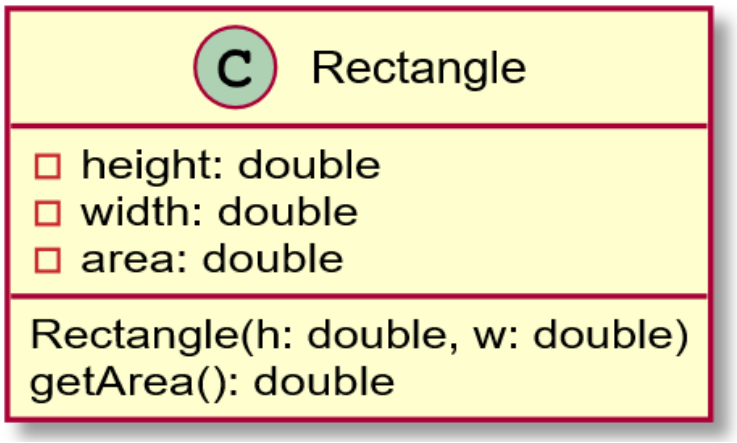

```
@startuml
class Rectangle {
    height: double
    width: double
    area: double
    Rectangle(h: double, w: double)
    getArea(): double
}
@enduml
```



```
@startuml
class CelebProfile {
    realAge: int
    CelebProfile(age: int)
    getAge(): int
}
@enduml
```

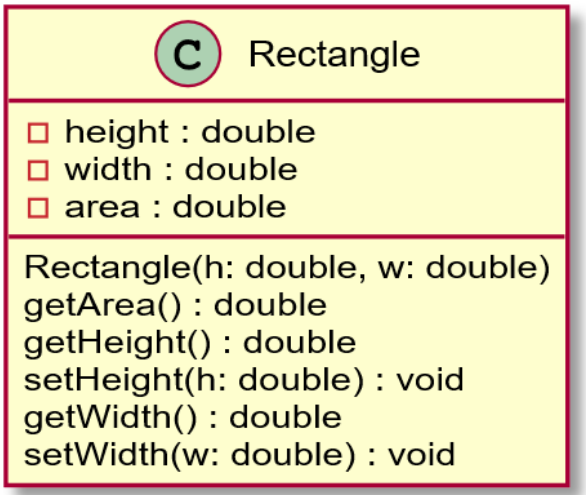



```
@startuml
class Rectangle {
    -height: double
    -width: double
    -area: double
    Rectangle(h: double, w: double)
    getArea(): double
}
@enduml
```

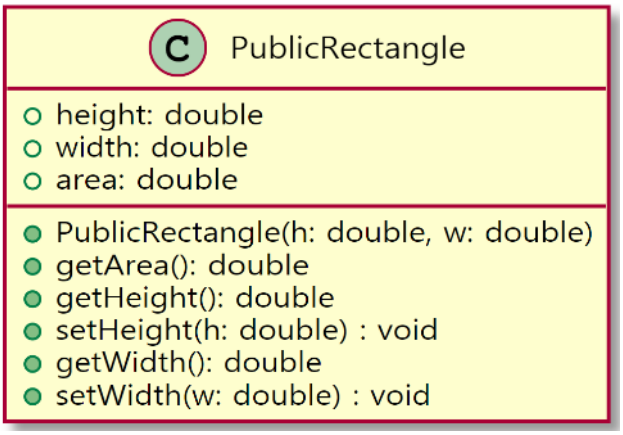



```
@startuml
class Rectangle {
    -height: double
    -width: double
    -area: double
    Rectangle(h: double, w: double)
    getArea(): double
    getHeight(): double
    setHeight(h: double) : void
    getWidth(): double
    setWidth(w: double) : void
}
```

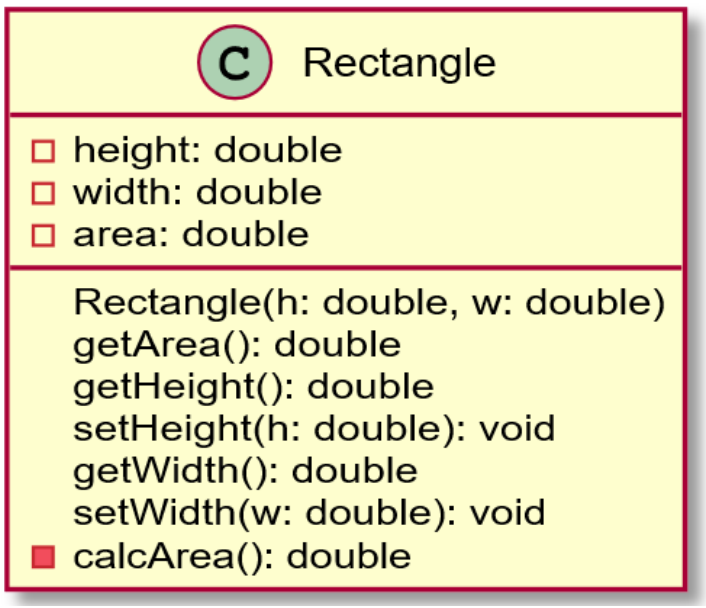
```
@enduml
```



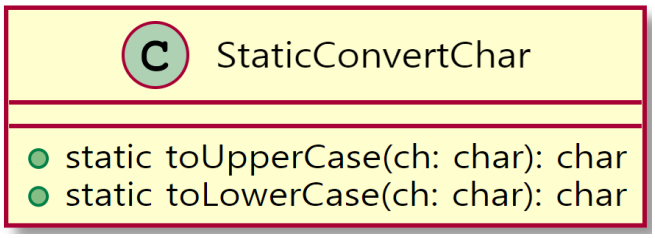

```
@startuml
class PublicRectangle {
    +height: double
    +width: double
    +area: double
    +PublicRectangle(h: double, w: double)
    +getArea(): double
    +getHeight(): double
    +setHeight(h: double) : void
    +getWidth(): double
    +setWidth(w: double) : void
}
@enduml
```



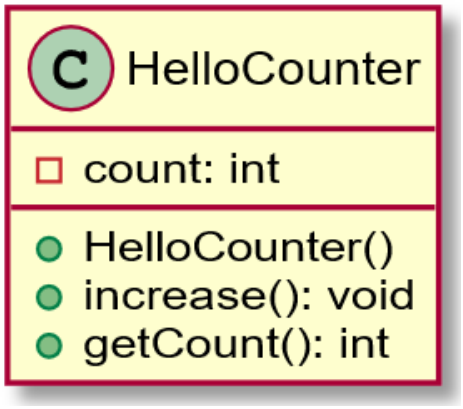

```
@startuml
class Rectangle {
    -height: double
    -width: double
    -area: double
    Rectangle(h: double, w: double)
    getArea(): double
    getHeight(): double
    setHeight(h: double): void
    getWidth(): double
    setWidth(w: double): void
    -calcArea(): double
}
@enduml
```



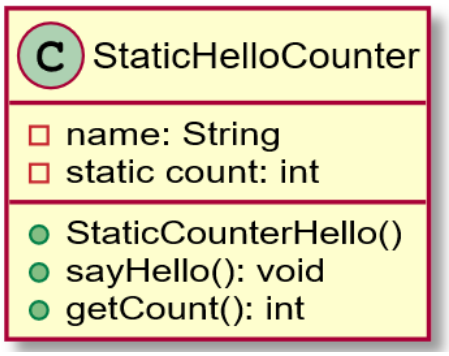

```
@startuml
class StaticConvertChar {
    +static toUpperCase(ch: char): char
    +static toLowerCase(ch: char): char
}
@enduml
```



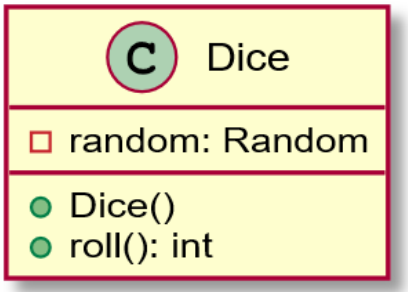

```
@startuml
class HelloCounter {
    -count: int
    +HelloCounter()
    +increase(): void
    +getCount(): int
}
@enduml
```




```
@startuml
class StaticHelloCounter {
    -name: String
    -static count: int
    +StaticCounterHello()
    +sayHello(): void
    +getCount(): int
}
@enduml
```

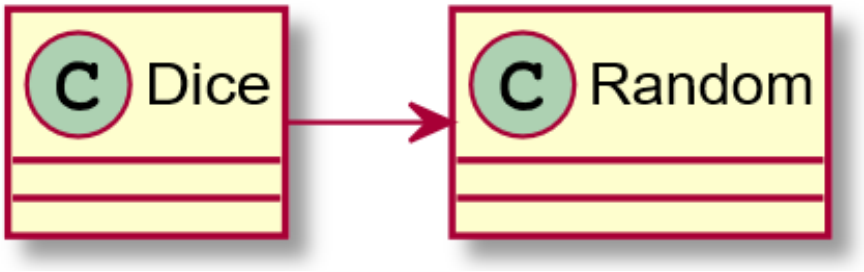



```
@startuml
class Dice {
    -random: Random
    +Dice(random: Random)
    +roll(): int
}
@enduml
```

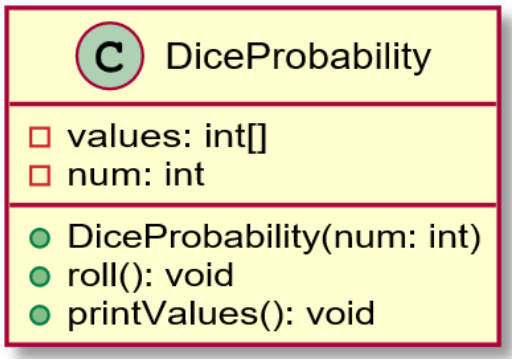



```
@startuml
class Dice
```

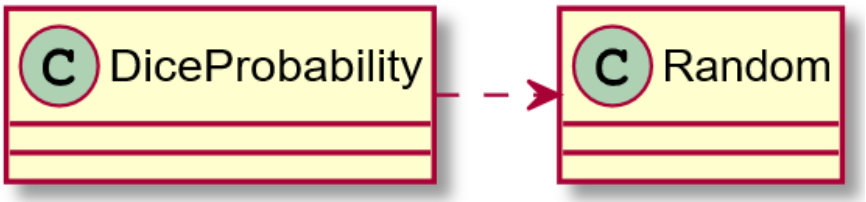
```
class Random
Dice -right-> Random
@enduml
```



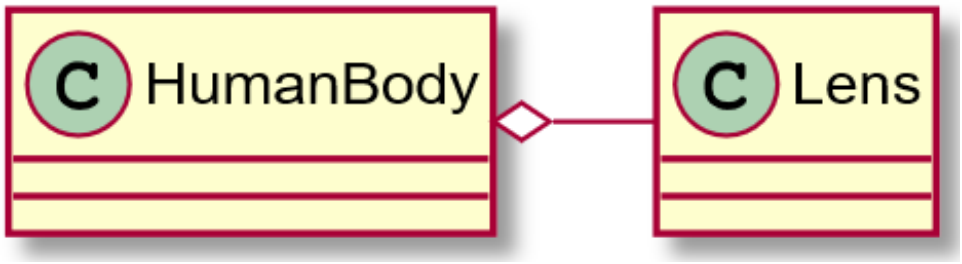

```
@startuml
class DiceProbability {
    -values: int[]
    -num: int
    +DiceProbability(num: int)
    +roll(): void
    +printValues(): void
}
@enduml
```



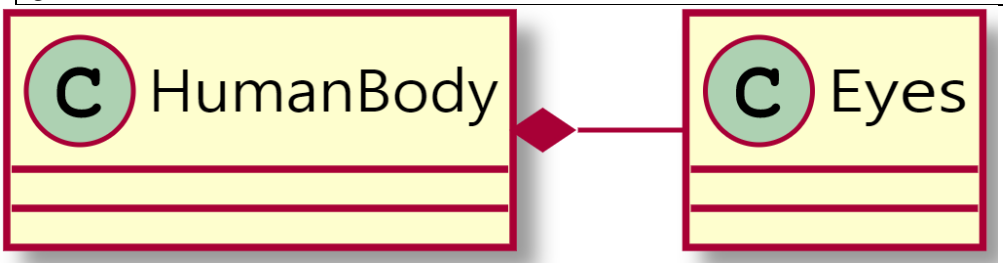

```
@startuml
class DiceProbability
class Random
DiceProbability .right.> Random
@enduml
```



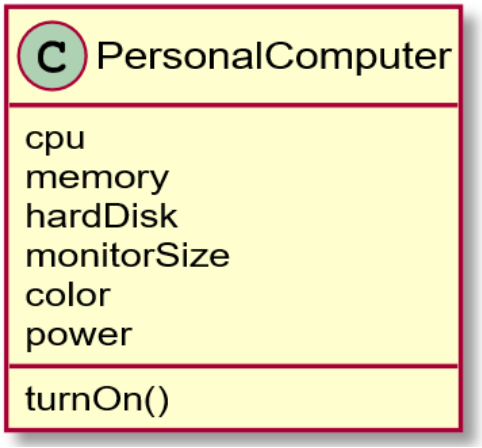

```
@startuml
class HumanBody
class Lens
HumanBody o-- Lens
@enduml
```




```
@startuml
class HumanBody
class Eyes
HumanBody *-- Eyes
@enduml
```

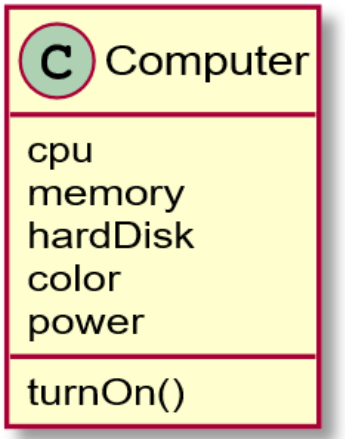


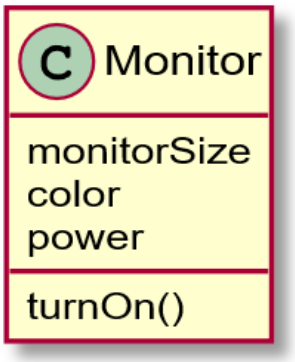

```
@startuml
class PersonalComputer {
    cpu
    memory
    hardDisk
    monitorSize
    computerColor
    computerPower
    monitorColor
    monitorPower
    turnOnComputer()
    turnOnMonitor()
}
@enduml
```




```
@startuml
class Computer {
    cpu
    memory
```

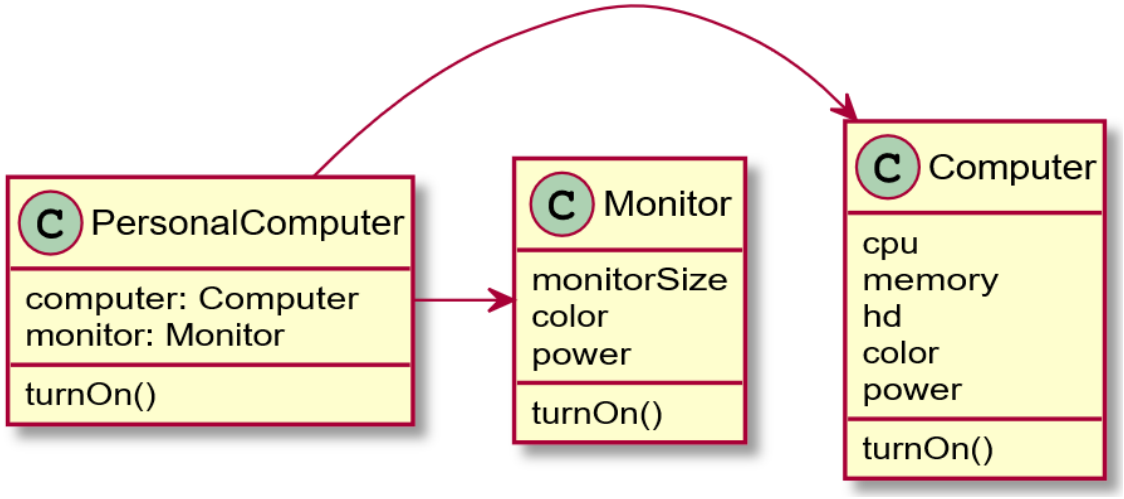
```
    hardDisk
    color
    power
    turnOn()
}

class Monitor {
    monitorSize
    color
    power
    turnOn()
}
@enduml
```



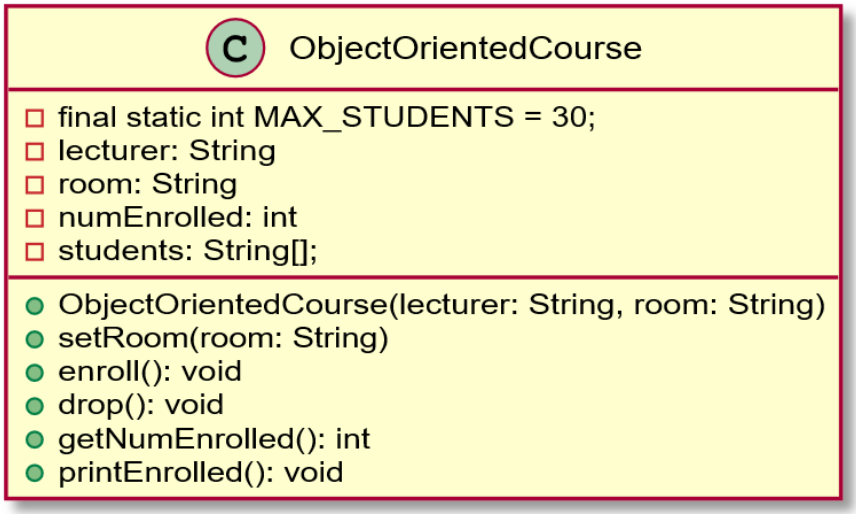



```
@startuml
class Computer {
    cpu
    memory
    hd
    color
    power
    turnOn()
}

class Monitor {
    monitorSize
    color
    power
    turnOn()
}

class PersonalComputer {
    computer: Computer
    monitor: Monitor
    turnOn()
}

PersonalComputer -right-> Computer
PersonalComputer -right-> Monitor
```

```
@enduml
```

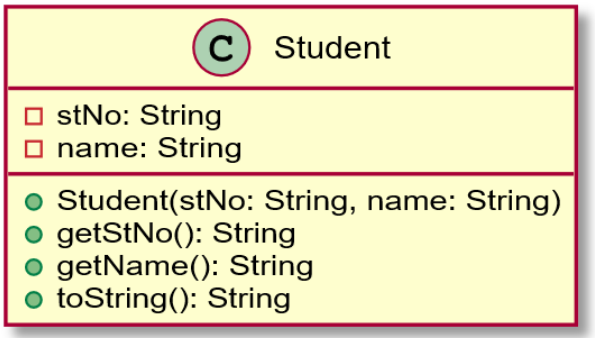



```
@startuml
class ObjectOrientedCourse {
    -final static int MAX_STUDENTS = 30;
    -lecturer: String
    -room: String
    -numEnrolled: int
    -students: String[];
    +ObjectOrientedCourse(lecturer: String, room: String)
    +setRoom(room: String)
    +enroll(): void
    +drop(): void
    +getNumEnrolled(): int
    +printEnrolled(): void
}
@enduml
```

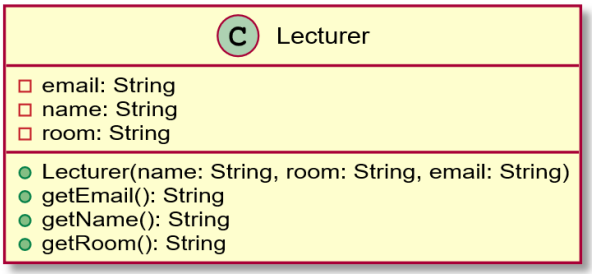



```
@startuml
class Student {
    -stNo: String
    -name: String
    +Student(stNo: String, name: String)
    +getStNo(): String
```

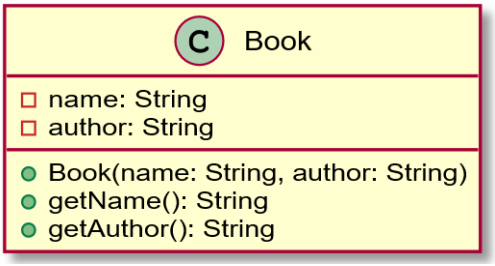
```
    +getName(): String
    +toString(): String
}
@enduml
```



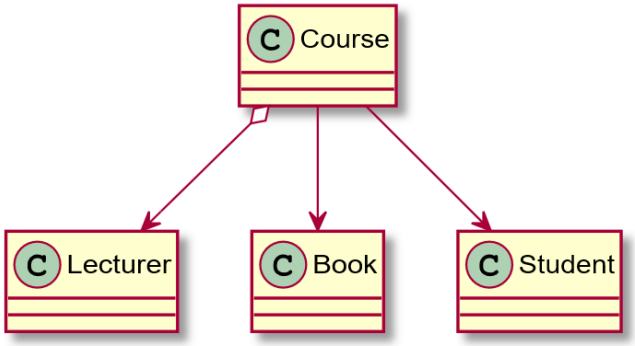

```
@startuml
class Lecturer {
    -email: String
    -name: String
    -room: String

    +Lecturer(name: String, room: String, email: String)
    +getEmail(): String
    +getName(): String
    +getRoom(): String
}
@enduml
```



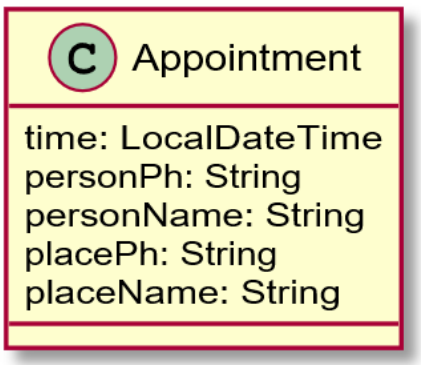

```
@startuml
class Book {
    -name: String
    -author: String
    +Book(name: String, author: String)
    +getName(): String
    +getAuthor(): String
}
@enduml
```




```
@startuml
class Course
class Lecturer
class Book
class Student
Course o--> Lecturer
Course --> Book
Course --> Student
@enduml
```

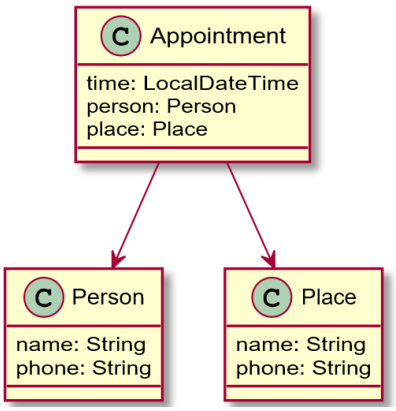



```
@startuml
class Appointment {
    time: LocalDateTime
    personPh: String
    personName: String
    placePh: String
    placeName: String
}
@enduml
```

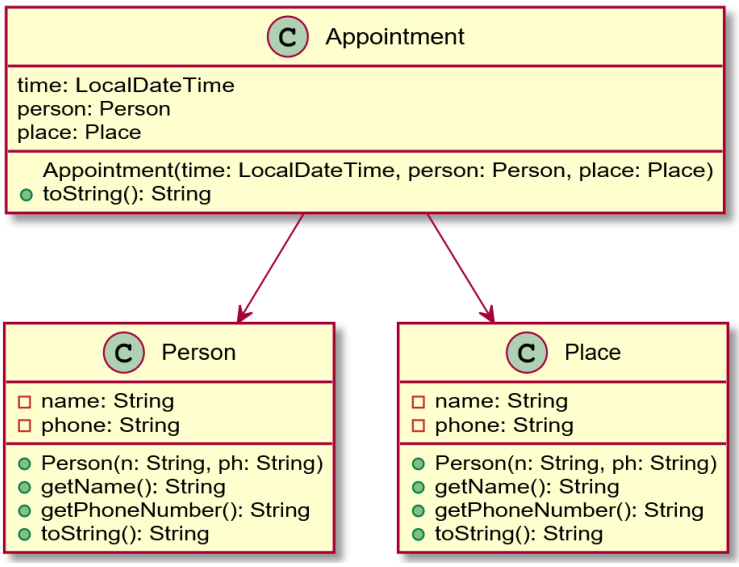



```
@startuml
class Appointment {
    time: LocalDateTime
    person: Person
    place: Place
```

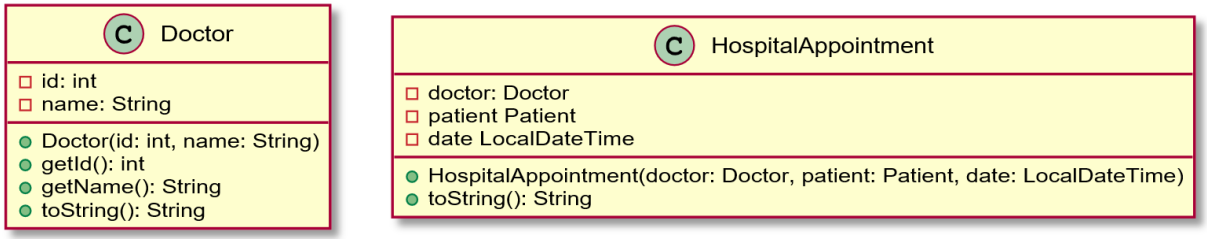
```
}
class Person {
    name: String
    phone: String
}
class Place {
    name: String
    phone: String
}
Appointment --> Person
Appointment --> Place
@enduml
```



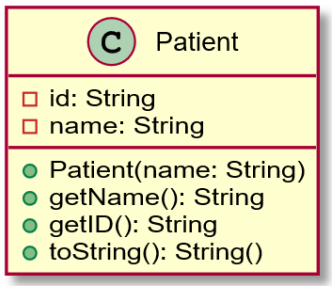

```
@startuml
class Appointment {
    time: LocalDateTime
    person: Person
    place: Place
    Appointment(time: LocalDateTime, person: Person, place: Place)
    +toString(): String
}
class Person {
    -name: String
    -phone: String
    +Person(n: String, ph: String)
    +getName(): String
    +getPhoneNumber(): String
    +toString(): String
}
class Place {
    -name: String
    -phone: String
    +Person(n: String, ph: String)
    +getName(): String
    +getPhoneNumber(): String
    +toString(): String
}
Appointment --> Person
Appointment --> Place
@enduml
```



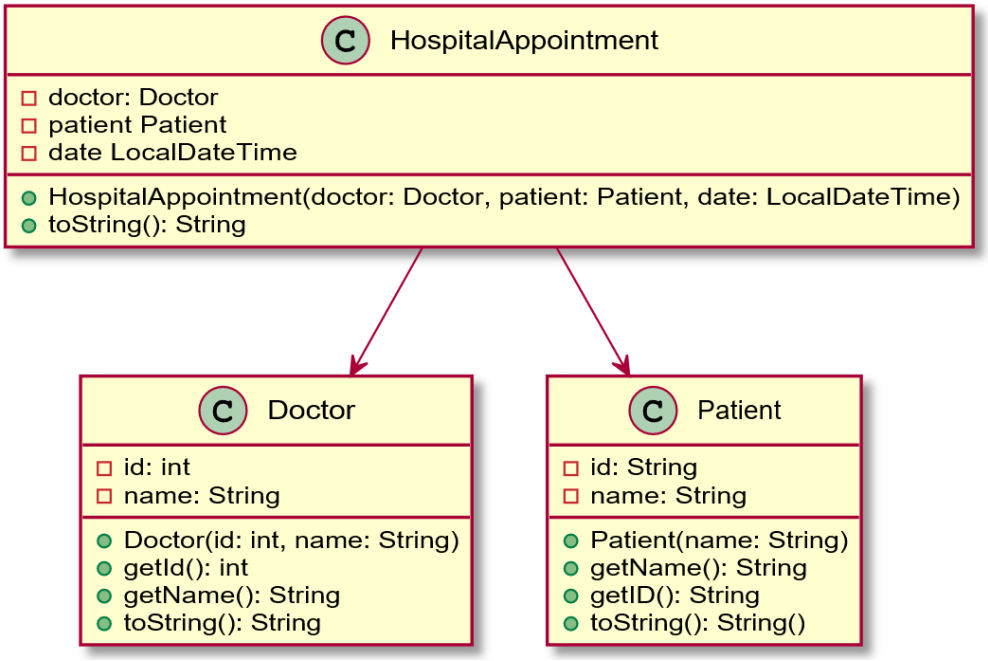

```
@startuml
class Doctor {
    -id: int
    -name: String
    +Doctor(id: int, name: String)
    +getId(): int
    +getName(): String
    +toString(): String
}

class HospitalAppointment {
    -doctor: Doctor
    -patient Patient
    -date LocalDateTime
    +HospitalAppointment(doctor: Doctor, patient: Patient, date:
LocalDateTime)
    +toString(): String
}
@enduml
```



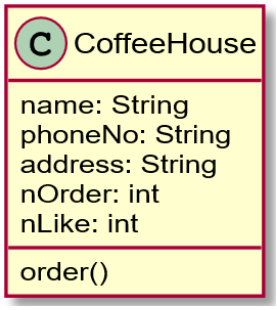

```
@startuml
class Patient {
    -id: String
    -name: String
    +Patient(name: String)
    +getName(): String
    +getID(): String
    +toString(): String()
}
@enduml
```



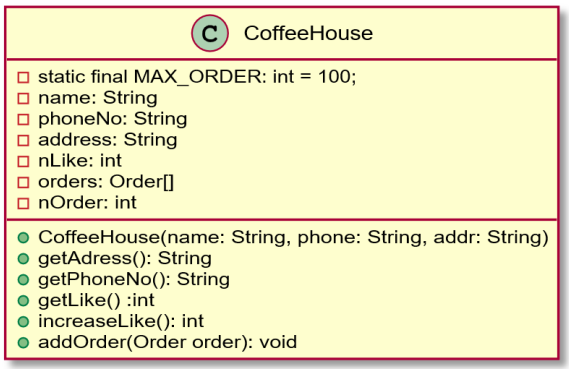

```
@startuml
'Doctor, Patient, HospitalAppointment 는 앞에서 작성한 내용과 같음
class Doctor
class Patient
class HospitalAppointment

HospitalAppointment --> Doctor
HospitalAppointment --> Patient
@enduml
```

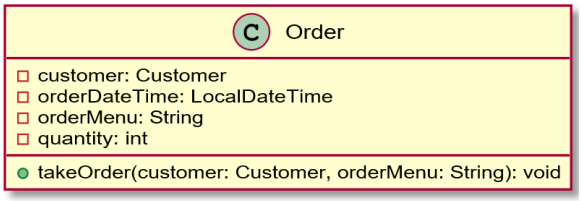



```
@startuml
class CoffeeHouse {
    name: String
    phoneNo: String
    address: String
    orders: String[]
    nOrder: int
    nLike: int
    order()
}
@enduml
```



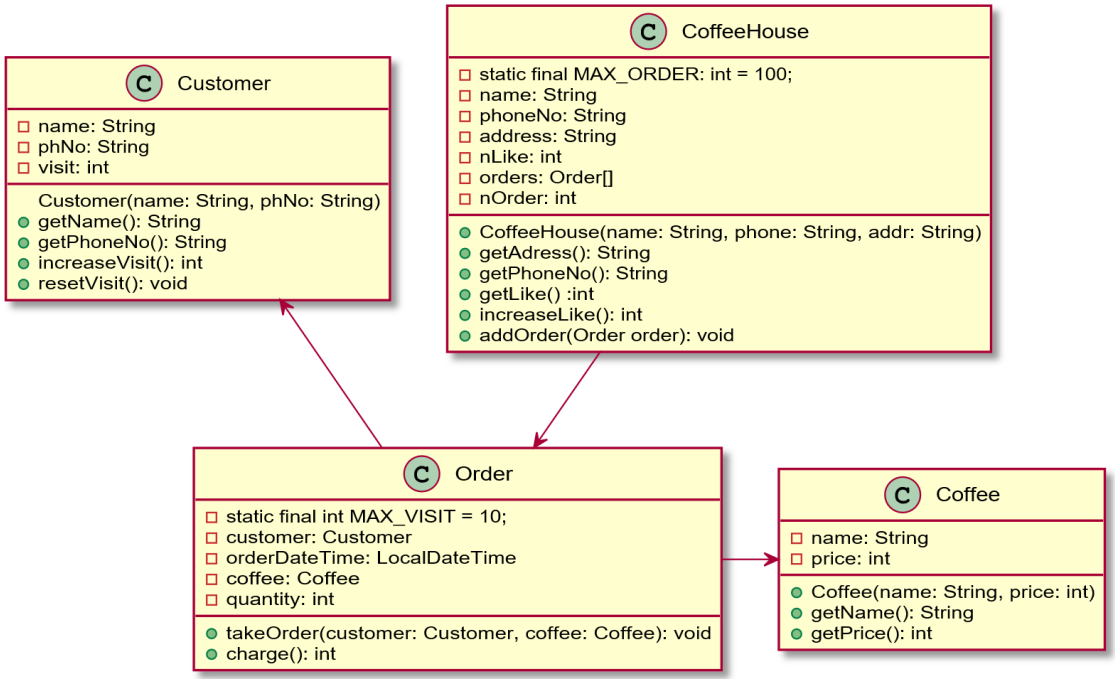

```
@startuml
class CoffeeHouse {
    -static final MAX_ORDER: int = 100;
    -name: String
    -phoneNo: String
    -address: String
    -nLike: int
    -orders: Order[]
    -nOrder: int
    +CoffeeHouse(name: String, phone: String, addr: String)
    +getAdress(): String
    +getPhoneNo(): String
    +getLike() :int
    +increaseLike(): void
    +addOrder(Order order): void
}
class Order {
    -customer: Customer
    -orderDateTime: LocalDateTime
    -orderMenu: String
    -quantity: int
    +takeOrder(customer: Customer, orderMenu: String): void
}
@enduml
```






```
@startuml
class CoffeeHouse {
    -static final MAX_ORDER: int = 100;
    -name: String
    -phoneNo: String
    -address: String
    -nLike: int
    -orders: Order[]
    -nOrder: int
    +CoffeeHouse(name: String, phone: String, addr: String)
```

```
    +getAdress(): String
    +getPhoneNo(): String
    +getLike() :int
    +increaseLike(): int
    +addOrder(Order order): void
}
class Order {
    -static final int MAX_VISIT = 10;
    -customer: Customer
    -orderDateTime: LocalDateTime
    -coffee: Coffee
    -quantity: int
    +takeOrder(customer: Customer, coffee: Coffee): void
    +charge(): int
}
class Customer {
    -name: String
    -phNo: String
    -visit: int
    Customer(name: String, phNo: String)
    +getName(): String
    +getPhoneNo(): String
    +increaseVisit(): int
    +resetVisit(): void
}
class Coffee {
    -name: String
    -price: int
    +Coffee(name: String, price: int)
    +getName(): String
    +getPrice(): int
}
CoffeeHouse --> Order
Order-up->Customer
Order->Coffee
@enduml
```



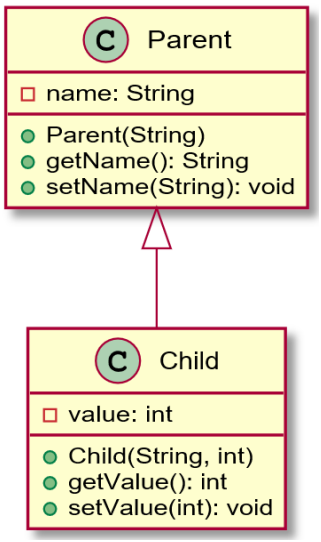
@startuml

```
class Dice {
    -number: int
    +roll(): void
    +getNum(): int
}

class Player {
```

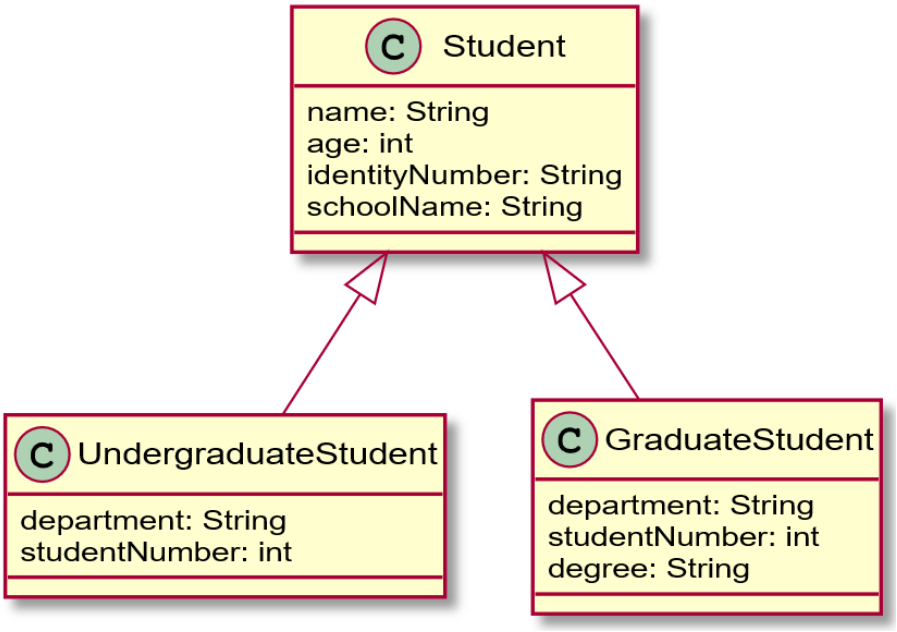
```
    -name: String
    -num: int
    +Player(name: String)
    +play(): void
}

class DiceGame {
    -p1: Player
    -p2: Player
    +startGame(): void
    +static main(args: String[]): void
}
@enduml
```

7 장

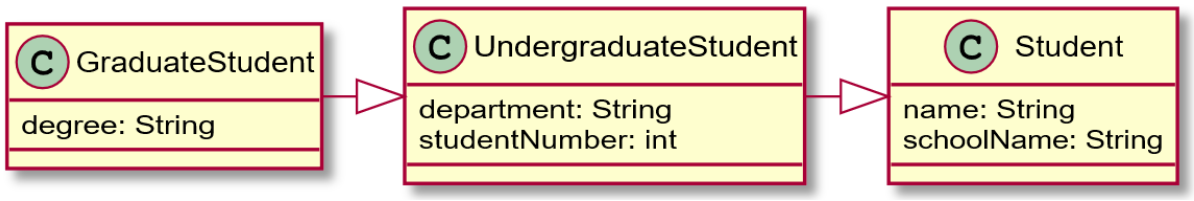
```
@startuml
class Parent {
   -name: String
   +Parent(String)
   +getName(): String
   +setName(String): void
}

class Child extends Parent {
    -value: int
    +Child(String, int)
    +getValue(): int
    +setValue(int): void
}
' Child --|> Parent ` 주석 처리됨. 앞에서 extends 를 사용하므로 불필요함
@enduml
```



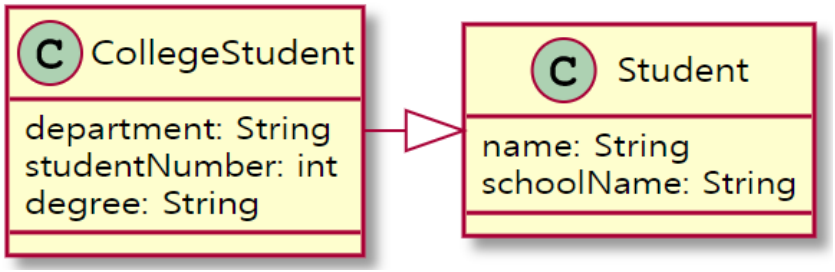

```
@startuml
class Student {
   name: String
   schoolName: String
}

class UndergraduateStudent extends Student {
    department: String
    studentNumber: int
}

class GraduateStudent extends Student {
    department: String
    studentNumber: int
    degree: String
}
@enduml
```



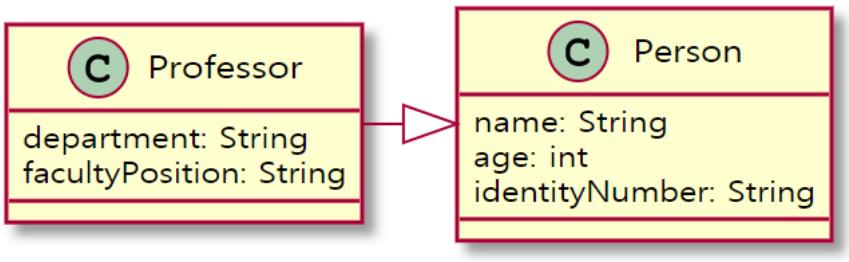

```
@startuml
class Student {
   name: String
   schoolName: String
}

class UndergraduateStudent {
    department: String
    studentNumber: int
}

class GraduateStudent {
    degree: String
}

Student <|-left- UndergraduateStudent
UndergraduateStudent <|-left- GraduateStudent
@enduml
```



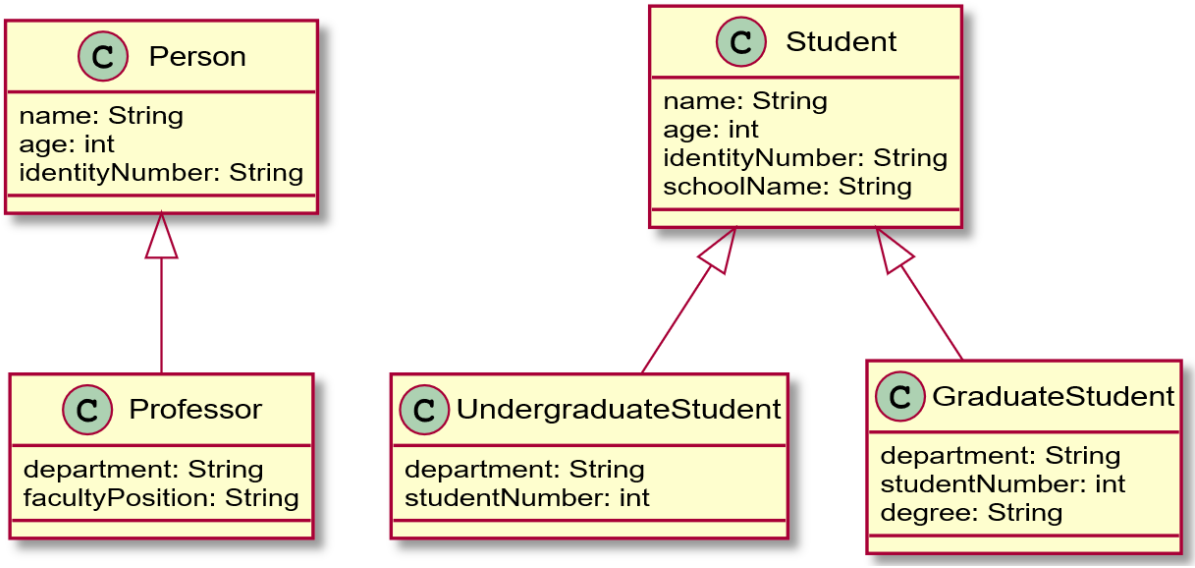

```
@startuml
class Student {
   name: String
   schoolName: String
}

class CollegeStudent {
    department: String
    studentNumber: int
    degree: String
}

Student <|-- CollegeStudent
@enduml
```



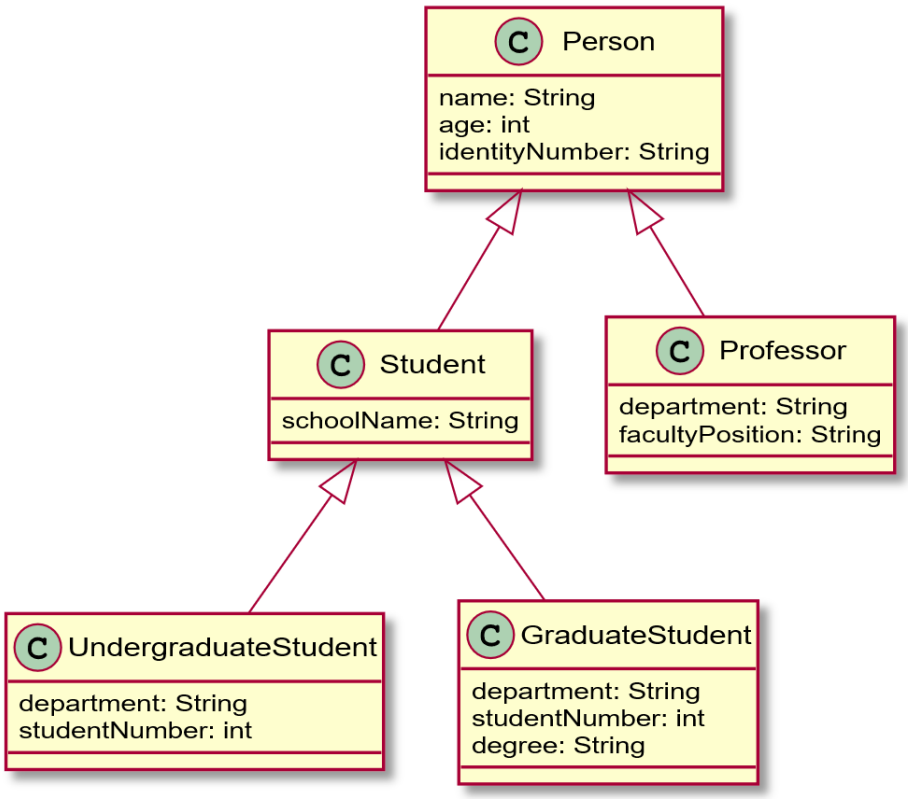

```
@startuml
class Person {
   name: String
   age: int
   identityNumber: String
}

class Professor extends Person {
    department: String
    facultyPosition: String
}
@enduml
```



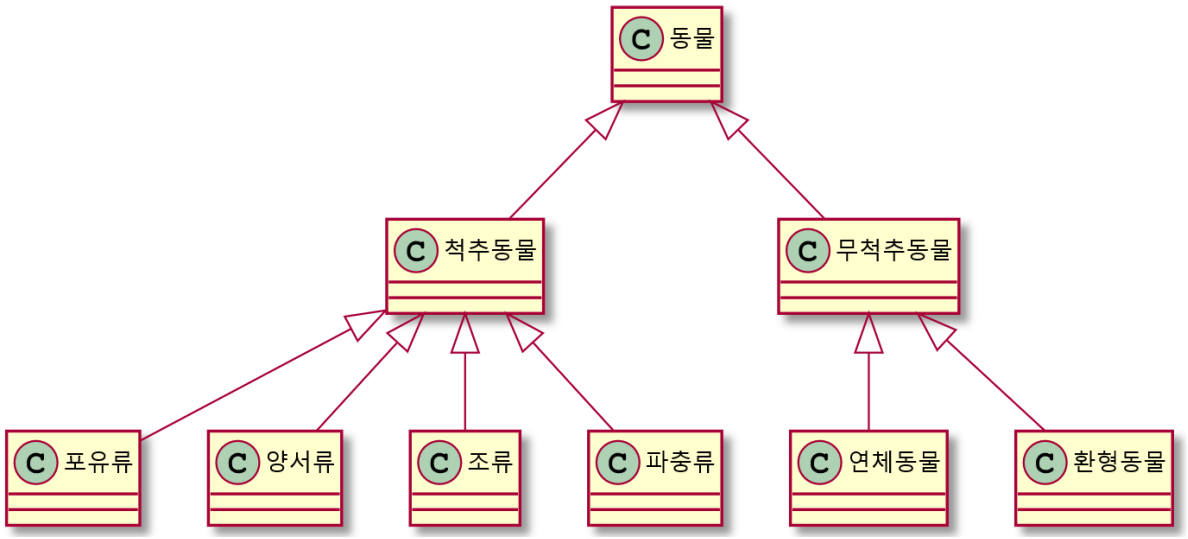

```
@startuml
class Person {
   name: String
   age: int
   identityNumber: String
}

class Student {
    name: String
    age: int
    identityNumber: String
    schoolName: String
}

class UndergraduateStudent extends Student {
    department: String
    studentNumber: int
}

class GraduateStudent extends Student {
    department: String
    studentNumber: int
    degree: String
}

class Professor extends Person {
    department: String
    facultyPosition: String
}
@enduml
```



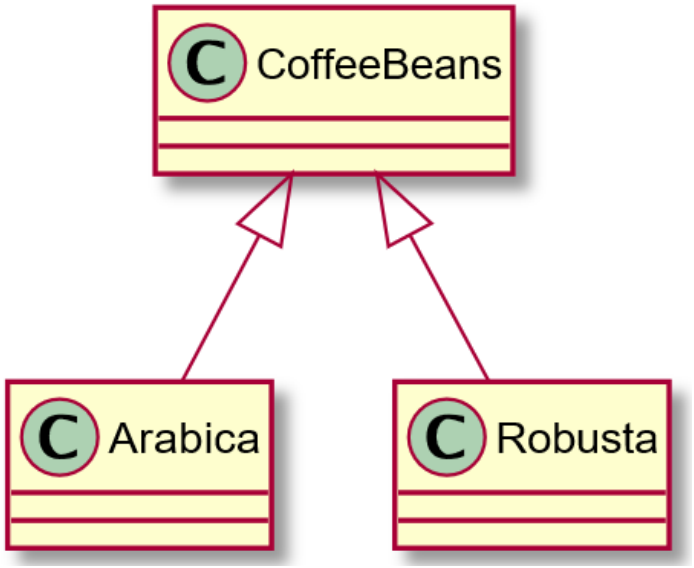

```
@startuml
class Person {
   name: String
   age: int
   identityNumber: String
}

class Student extends Person {
    schoolName: String
}

class UndergraduateStudent extends Student {
    department: String
    studentNumber: int
}

class GraduateStudent extends Student {
    department: String
    studentNumber: int
    degree: String
}

class Professor extends Person {
    department: String
    facultyPosition: String
}
@enduml
```



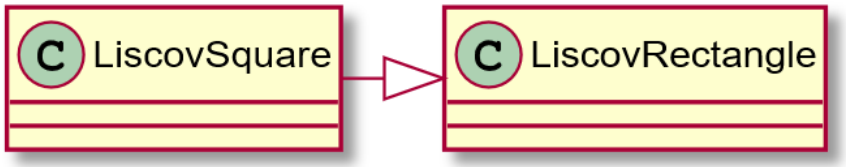

```
@startuml
class 동물
class 척추동물
class 포유류
class 양서류
class 조류
class 무척추동물
class 연체동물
class 환형동물

동물 <|-- 척추동물
동물 <|-- 무척추동물
척추동물 <|-- 포유류
척추동물 <|-- 양서류
척추동물 <|-- 조류
척추동물 <|-- 파충류
무척추동물 <|-- 연체동물
무척추동물 <|-- 환형동물
@enduml
```



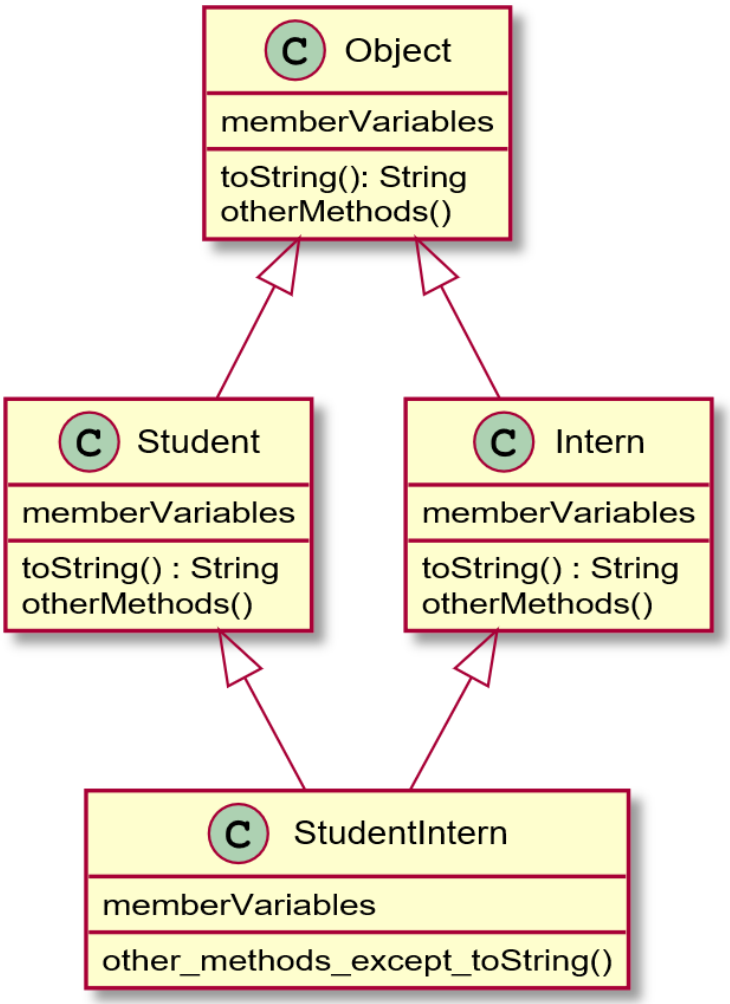

```
@startuml
class CoffeeBean
class Arabica
class Robusta
CoffeeBean <|-- Arabica
CoffeeBean <|-- Robusta
@enduml
```



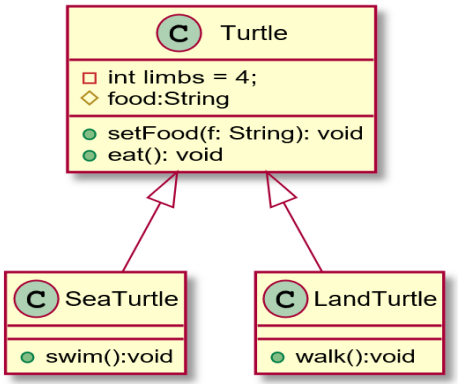

```
@startuml
class LiscovSquare
class LiscovRectangle
LiscovSquare -right-|> LiscovRectangle
@enduml
```



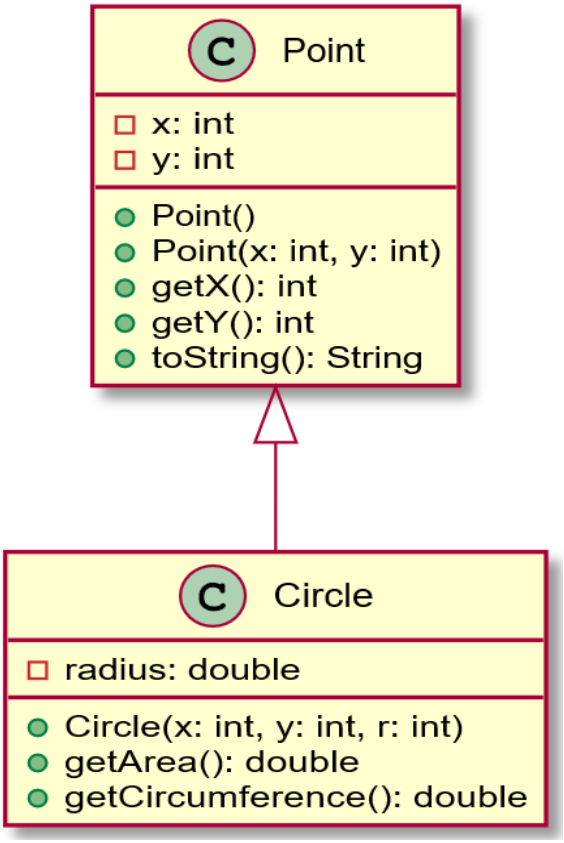
```
@startuml
class Object {
    memberVariables
    toString(): String
    otherMethods()
}
class Student {
    memberVariables
    toString() : String
    otherMethods()
}
class Intern {
    memberVariables
    toString() : String
    otherMethods()
}
class StudentIntern {
    memberVariables
    other_methods_except_toString()
}
Object <|-- Student
Object <|-- Intern
Student <|-- StudentIntern
Intern <|-- StudentIntern
@enduml
```



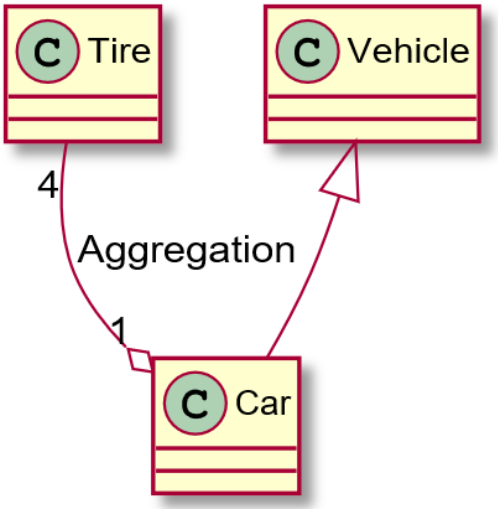
```
@startuml
class Turtle {
    -int limbs = 4;
    #food:String
    +setFood(f: String): void
    +eat(): void
}
class SeaTurtle extends Turtle{
    +swim():void
}
class LandTurtle extends Turtle {
    +walk():void
}
@enduml
```



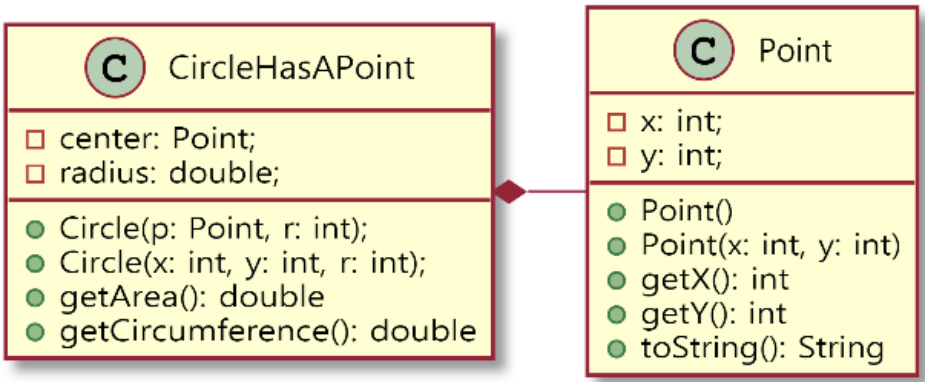

```
@startuml
class Point {
    -x: int
    -y: int
    +Point()
    +Point(x: int, y: int)
    +getX(): int
    +getY(): int
    +toString(): String
}

class Circle extends Point {
    -radius: double;
    +Circle(x: int, y: int, r: int);
    +getArea(): double
    +getCircumference(): double
}
@enduml
```



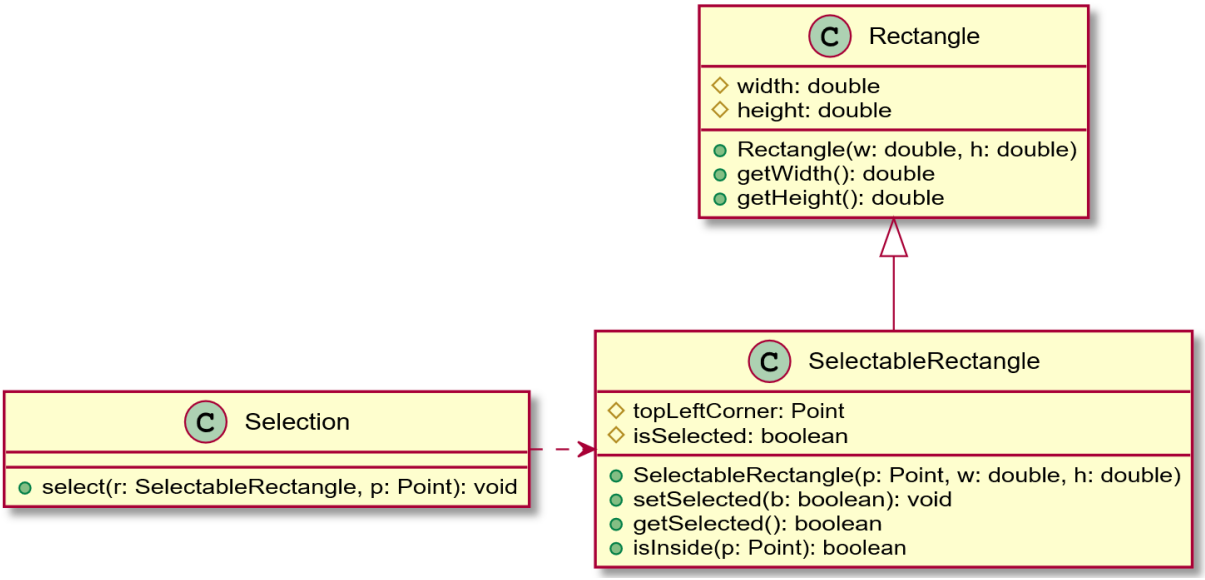

```
@startuml
class Vehicle
class Tire
class Car
Vehicle <|-- Car
Car "1" o-up- "4" Tire: Aggregation
@enduml
```



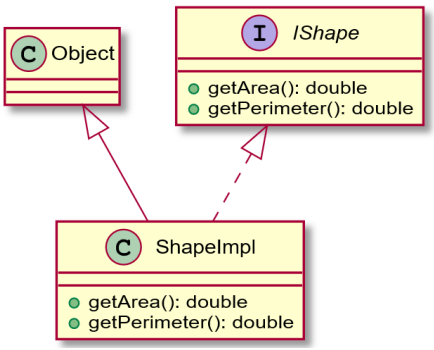
```
@startuml
class Point {
    -x: int;
    -y: int;
    +Point()
    +Point(x: int, y: int)
    +getX(): int
    +getY(): int
    +toString(): String
}

class CircleHasAPoint {
    -center: Point;
    -radius: double;
    +Circle(p: Point, r: int);
    +Circle(x: int, y: int, r: int);
    +getArea(): double
    +getCircumference(): double
}

CircleHasAPoint *-right- Point
@enduml
```



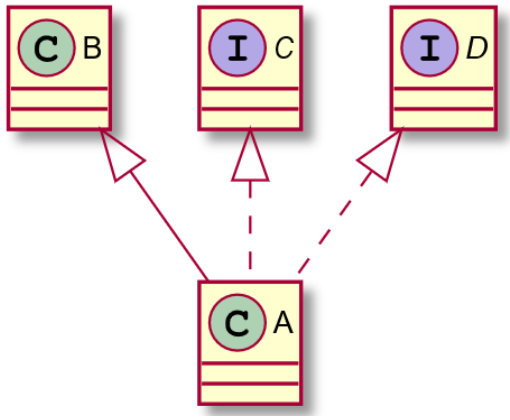

```
@startuml
class Rectangle {
    #width: double
    #height: double
    +Rectangle(w: double, h: double)
    +getWidth(): double
    +getHeight(): double
}

class SelectableRectangle extends Rectangle {
    #topLeftCorner: Point
    #isSelected: boolean
    +SelectableRectangle(p: Point, w: double, h: double)
    +setSelected(b: boolean): void
    +getSelected(): boolean
    +isInside(p: Point): boolean
}

class Selection {
    +select(r: SelectableRectangle, p: Point): void
}

Selection .right.> SelectableRectangle
@enduml
```



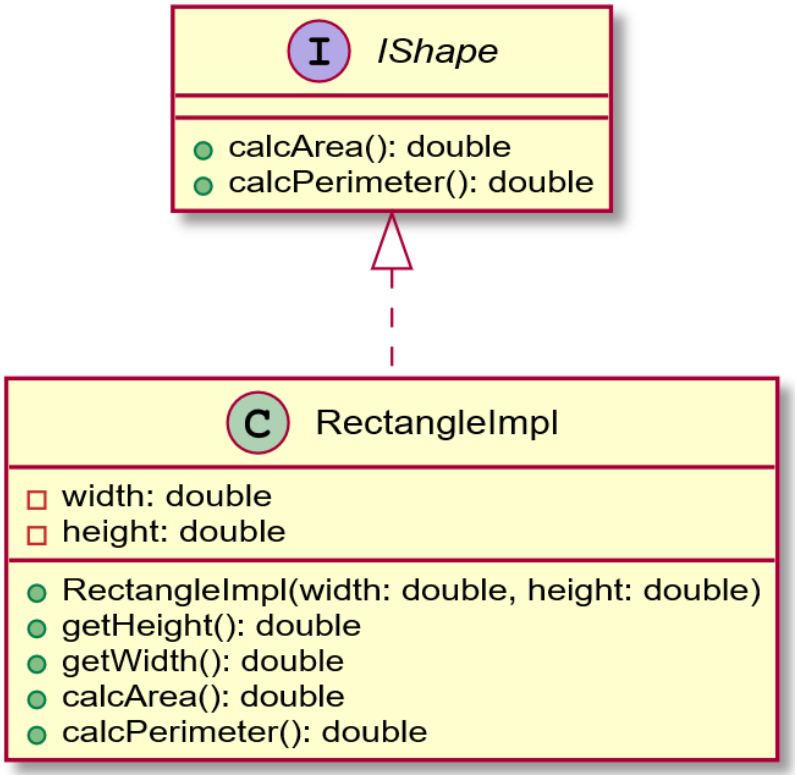

```
@startuml
interface IShape {
    +getArea(): double
    +getPerimeter(): double
}

class Object

class ShapeImpl implements IShape {
    +getArea(): double
    +getPerimeter(): double
}

ShapeImpl -up-|> Object
@enduml
```



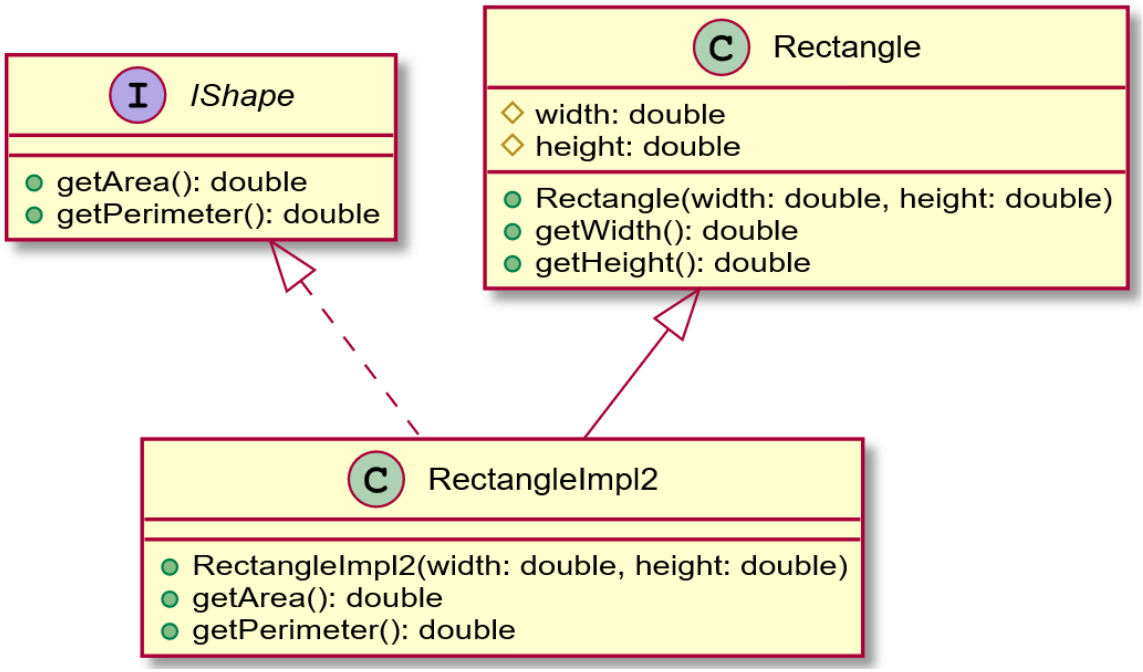

```
@startuml
class A
class B
interface C
interface D
B <|-- A
C <|.. A
D <|.. A
@enduml
```




```
@startuml
interface IShape {
    +getArea(): double
    +getPerimeter(): double
}
class RectangleImpl {
    -width: double
    -height: double
    +RectangleImpl(width: double, height: double)
    +getHeight(): double
    +getWidth(): double
    +getArea(): double
    +getPerimeter(): double
}
RectangleImpl .up.|> IShape
@enduml
```

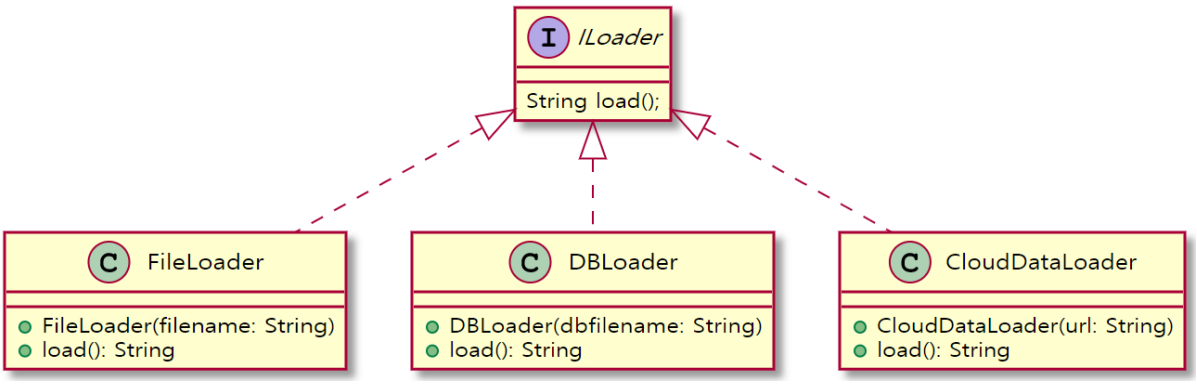


```
@startuml
interface IShape {
    +getArea(): double
    +getPerimeter(): double
}
class Rectangle {
    #width: double
    #height: double
    +Rectangle(width: double, height: double)
    +getWidth(): double
    +getHeight(): double
}
class RectangleImpl2 {
    +RectangleImpl2(width: double, height: double)
    +getArea(): double
    +getPerimeter(): double
}
Rectangle <|-- RectangleImpl2
RectangleImpl2 .up.|> IShape
@enduml
```

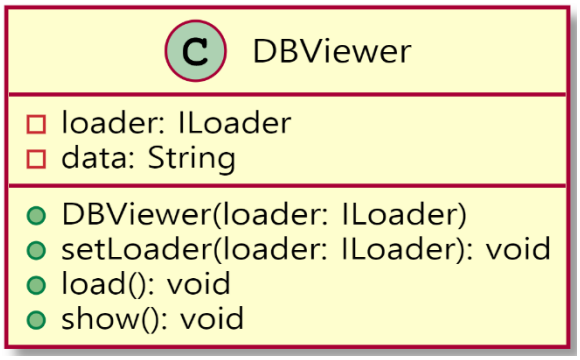



```
@startuml
interface ILoader {
    String load();
}

class FileLoader implements ILoader {
    +FileLoader(filename: String)
    +load(): String
}

class DBLoader implements ILoader {
    +DBLoader(dbfilename: String)
    +load(): String
}

class CloudDataLoader implements ILoader {
    +CloudDataLoader(url: String)
    +load(): String
}
```

```
@enduml
```

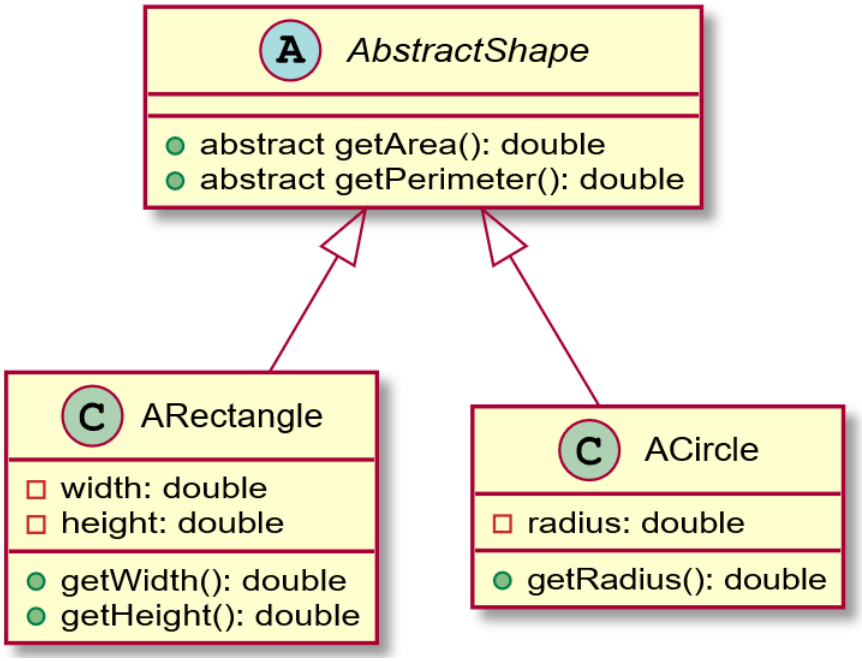



```
@startuml
class DBViewer {
    -loader: ILoader
    -data: String

    +DBViewer(loader: ILoader)
    +setLoader(loader: ILoader): void
    +load(): void
    +show(): void
}
@enduml
```

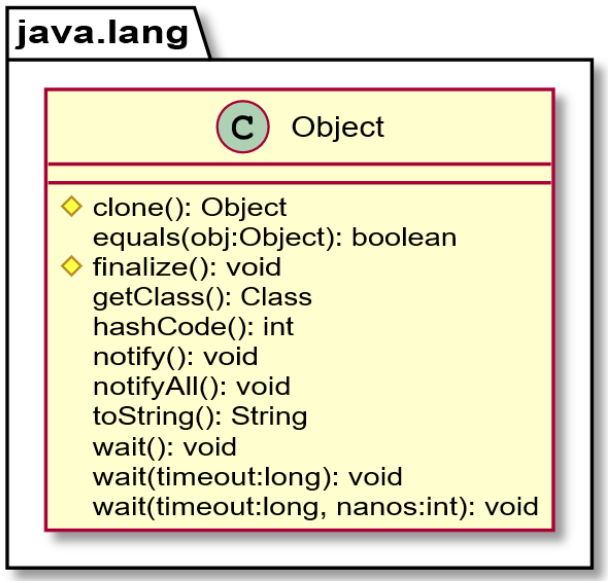



```
@startuml
abstract class AbstractShape {
    +abstract getArea(): double
    +abstract getPerimeter(): double
}

class ARectangle {
    -width: double
    -height: double
    +getWidth(): double
    +getHeight(): double
}

class ACircle {
    -radius: double
    +getRadius(): double
}

ARectangle -up-|> AbstractShape
```

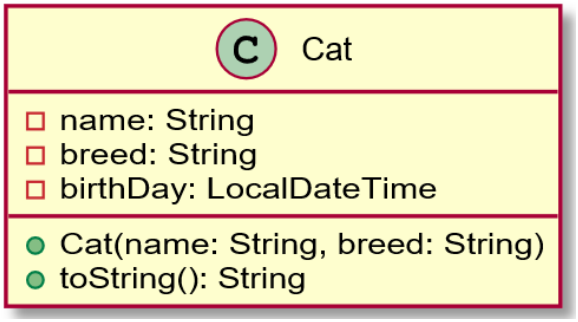
```
ACircle -up-|> AbstractShape
@enduml
```



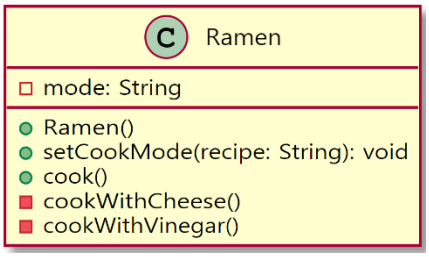

```
@startuml
class java.lang.Object{
    #clone(): Object
    equals(obj:Object): boolean
    #finalize(): void
    getClass(): Class
    hashCode(): int
    notify(): void
    notifyAll(): void
    toString(): String
    wait(): void
    wait(timeout:long): void
    wait(timeout:long, nanos:int): void
}
@enduml
```



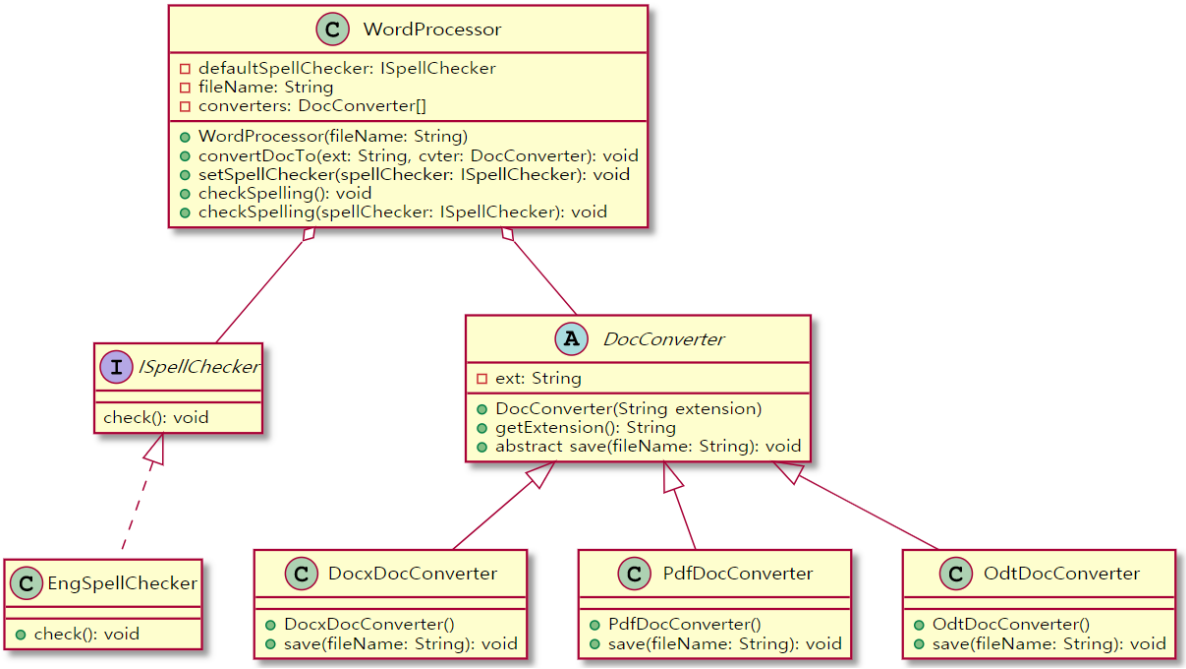
```
@startuml
class Cat {
    -name: String
    -breed: String
    -birthDay: LocalDateTime
    +Cat(name: String, breed: String)
    +toString(): String
}
@enduml
```



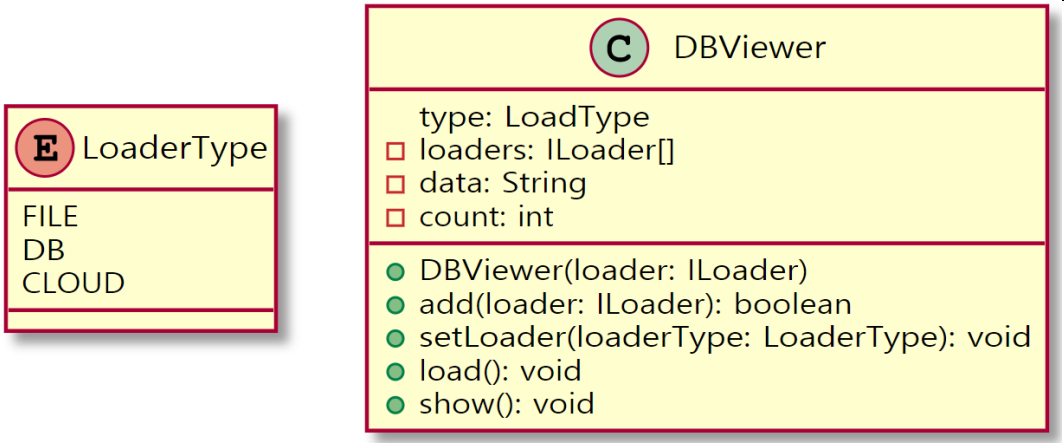

```
@startuml
class Ramen {
    -mode: String
    +Ramen()
    +setCookMode(recipe: String): void
    +cook(): void
    -cookWithCheese(): void
    -cookWithVinegar(): void
}
@enduml
```



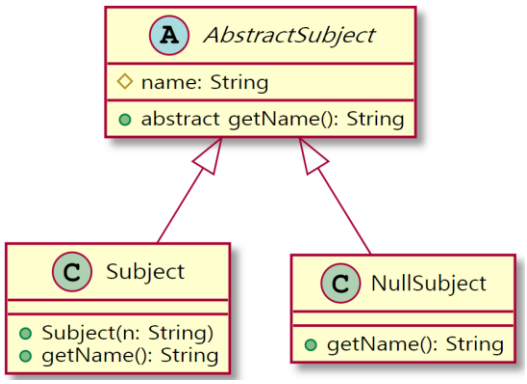
```
@startuml
interface ISpellChecker {
    check(): void
}

abstract class DocConverter {
    -ext: String

    +DocConverter(String extension)
    +getExtension(): String
    +abstract save(fileName: String): void
}

class DocxDocConverter extends DocConverter {
    +DocxDocConverter()
    +save(fileName: String): void
}

class PdfDocConverter extends DocConverter {
    +PdfDocConverter()
    +save(fileName: String): void
}

class OdtDocConverter extends DocConverter {
    +OdtDocConverter()
    +save(fileName: String): void
}

class WordProcessor {
    -defaultSpellChecker: ISpellChecker
    -fileName: String
    -converters: DocConverter[]

    +WordProcessor(fileName: String)
    +convertDocTo(ext: String, cvter: DocConverter): void
    +setSpellChecker(spellChecker: ISpellChecker): void
    +checkSpelling(): void
    +checkSpelling(spellChecker: ISpellChecker): void
}

class EngSpellChecker implements ISpellChecker {
    +check(): void
}

WordProcessor o-- ISpellChecker
WordProcessor o-- DocConverter
@enduml
```




```
@startuml
enum LoaderType {
    FILE
    DB
    CLOUD
}

class DBViewer {
    type: LoadType
    -loaders: ILoader[]
    -data: String
    -count: int

    +DBViewer(loader: ILoader)
    +add(loader: ILoader): boolean
    +setLoader(loaderType: LoaderType): void
    +load(): void
    +show(): void
}
@enduml
```

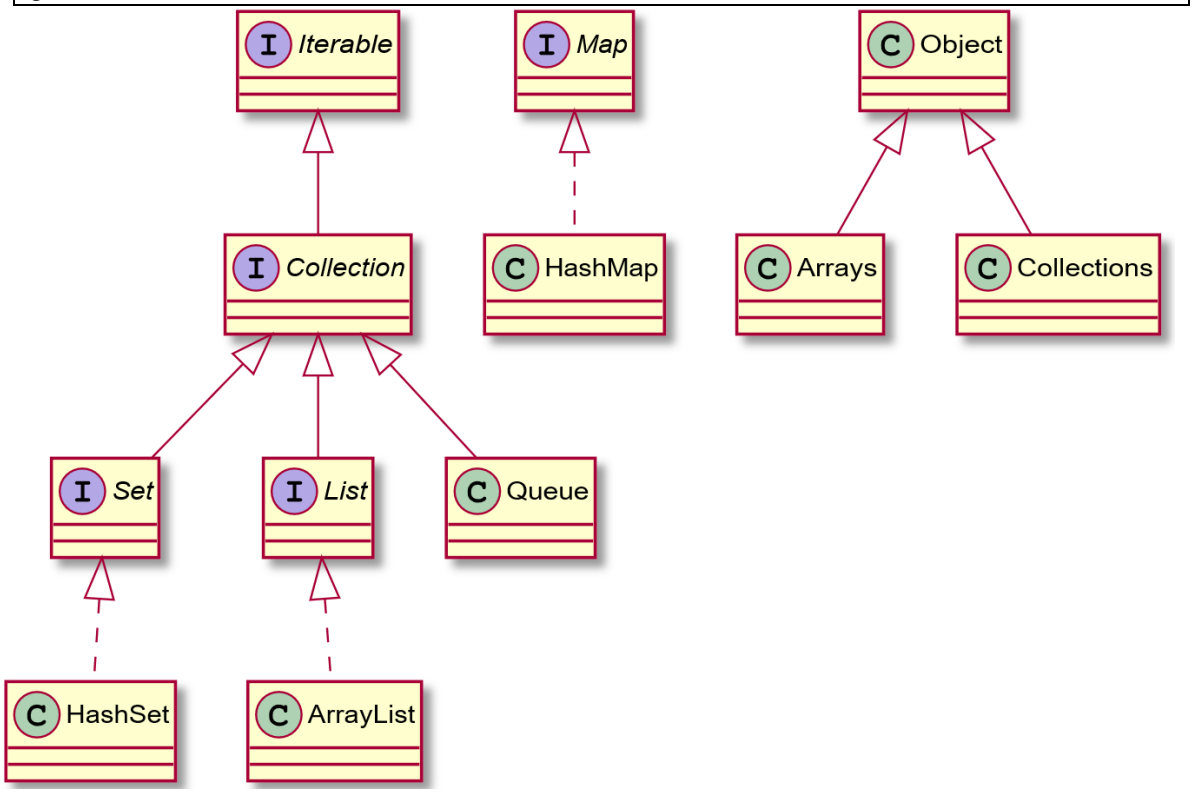


```
@startuml
abstract class AbstractSubject {
    #name: String
    +abstract getName(): String
}
class Subject extends AbstractSubject {
    +Subject(n: String)
    +getName(): String
}

class NullSubject extends AbstractSubject {
    +getName(): String
}
@enduml
```

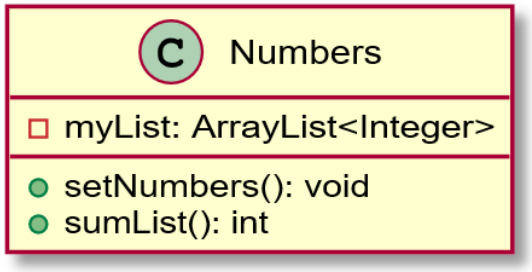


8 장

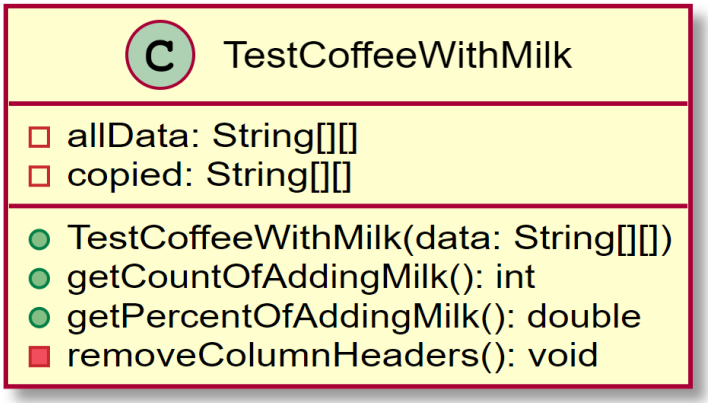
```
@startuml
interface Iterable
interface Collection
interface Set
interface List
interface Map
Iterable <|-- Collection
Collection <|-- Set
Collection <|-- List
Collection <|.. Queue
List <|.. ArrayList
Set <|.. HashSet
Map <|.. HashMap
class Object
class Arrays
class Collections
Object <|-- Arrays
Object <|-- Collections
@enduml
```



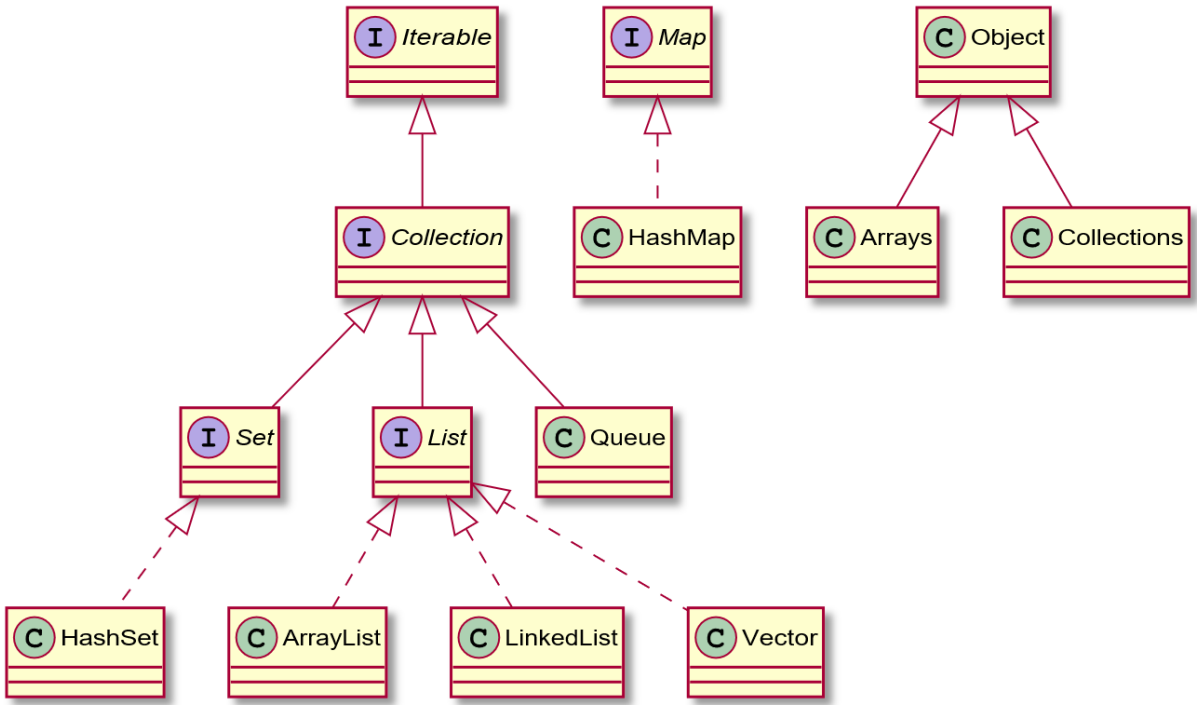

```
@startuml
class Numbers {
    myList: ArrayList<Integer>
    +setNumbers(): void
    +sumList(): int
}
@enduml
```



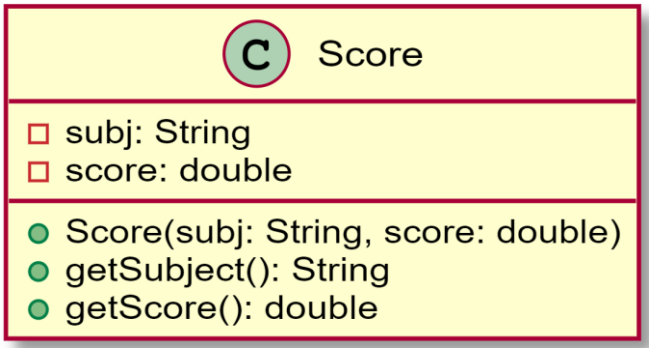

```
@startuml
class TestCoffeeWithMilk {
    -allData: String[][]
    -copied: String[][]

    +TestCoffeeWithMilk(String[][] data)
    +getCountOfAddingMilk(): int
    +getPercentOfAddingMilk(): double
    -removeColumnHeaders(): void
}
@enduml
```




```
@startuml
interface Iterable
interface Collection
interface Set
interface List
interface Map
Iterable <|-- Collection
Collection <|-- Set
Collection <|-- List
Collection <|-- Queue
List <|.. ArrayList
List <|.. LinkedList
List <|.. Vector
Set <|.. HashSet
Map <|.. HashMap
class Object
class Arrays
class Collections
Object <|-- Arrays
Object <|-- Collections
@enduml
```

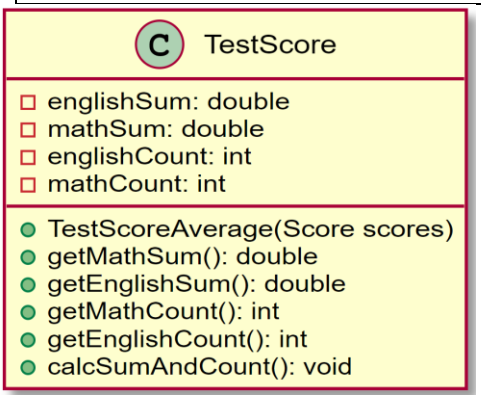



```
@startuml
class Score {
    -subj: String
    -score: double

    +Score(subj: String, score: double)
    +getSubject(): String
    +getScore(): double
}
@enduml
```

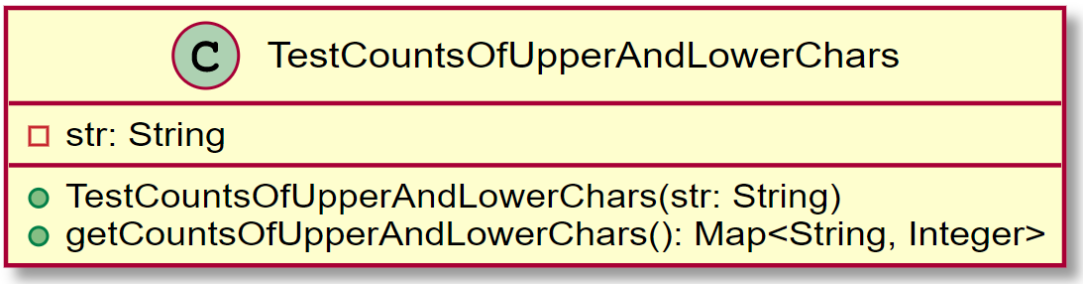



```
@startuml
class TestScore {
    -englishSum: double
    -mathSum: double
    -englishCount: int
    -mathCount: int
    -scores: Score[]

    +TestScore(scores: Score[])
    +getMathSum(): double
    +getEnglishSum(): double
    +getMathCount(): int
    +getEnglishCount(): int
    +calcSumAndCount(): void
}
```

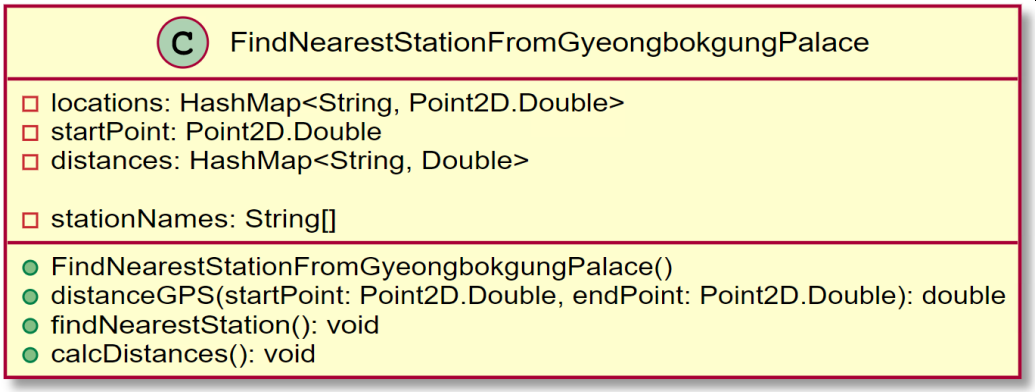
```
@enduml
```



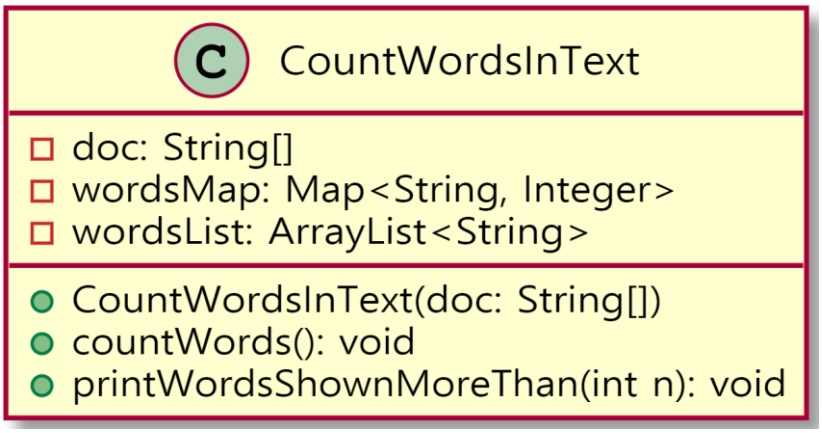

```
@startuml
class TestCountsOfUpperAndLowerChars {
    -str: String
    +TestCountsOfUpperAndLowerChars(str: String)
    +getCountsOfUpperAndLowerChars(): Map<String, Integer>
}
@enduml
```




```
@startuml
class FindNearestStationFromGyeongbokgungPalace {
    -locations: HashMap<String, Point2D.Double>
    -startPoint: Point2D.Double
    -distances: HashMap<String, Double>

    -stationNames: String[]

    +FindNearestStationFromGyeongbokgungPalace()
    +distanceGPS(startPoint: Point2D.Double, endPoint: Point2D.Double):
double
    +findNearestStation(): void
    +calcDistances(): void
}
@enduml
```

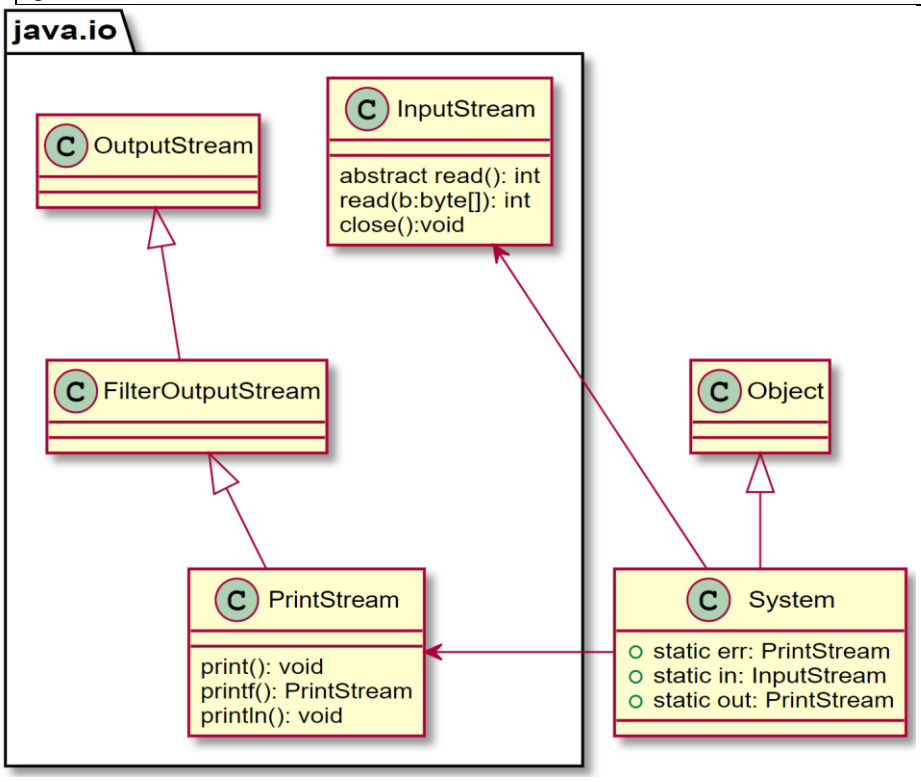


```
@startuml
class CountWordsInText {
    -doc: String[]
    -wordsMap: Map<String, Integer>
    -wordsList: ArrayList<String>
    +CountWordsInText(doc: String[])
    +countWords(): void
    +printWordsShownMoreThan(int n): void
}
@enduml
```

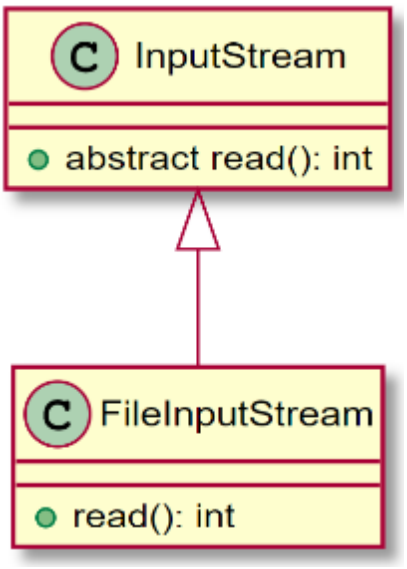


9 장

```
@startuml
class Object
class System {
    +static err: PrintStream
    +static in: InputStream
    +static out: PrintStream
}
class java.io.InputStream {
    abstract read(): int
    read(b:byte[]): int
    close():void
}
class java.io.PrintStream {
    print(): void
    printf(): PrintStream
    println(): void
}
java.io.PrintStream -up-|> java.io.FilterOutputStream
java.io.FilterOutputStream -up-|> java.io.OutputStream
Object <|-- System
System -left-> java.io.InputStream
System -right-> java.io.PrintStream
@enduml
```

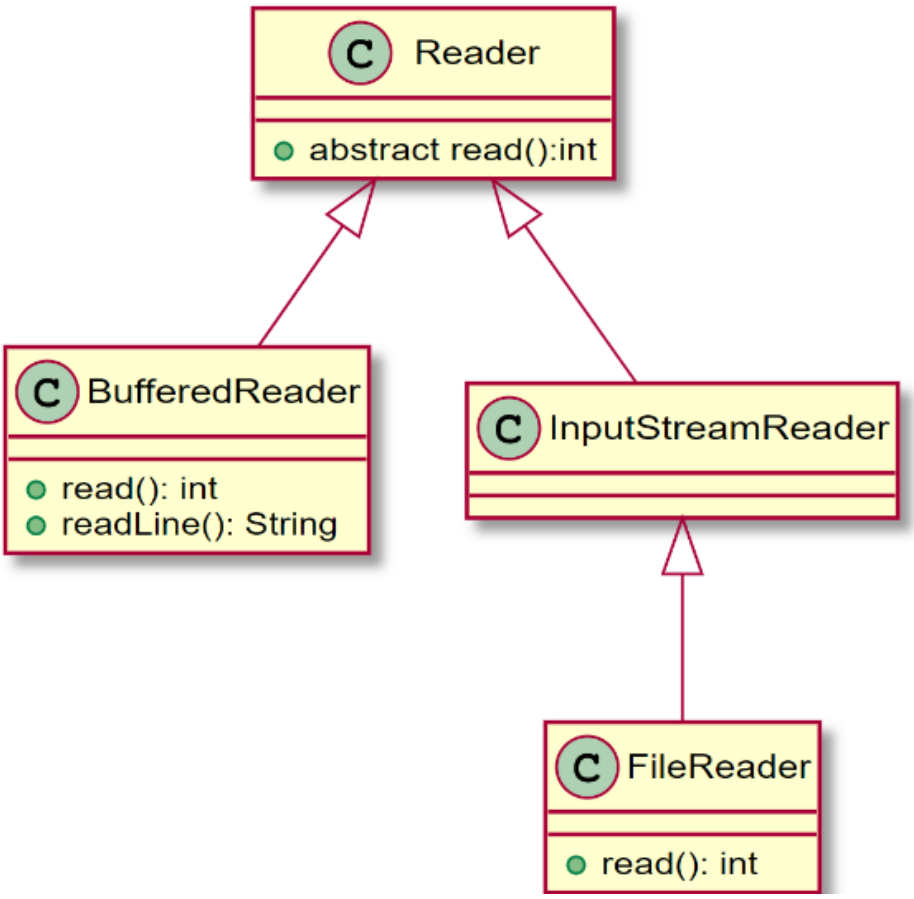



```
@startuml
class InputStream {
  +abstract read(): int
}
class FileInputStream extends InputStream {
  +read(): int
}
@enduml
```



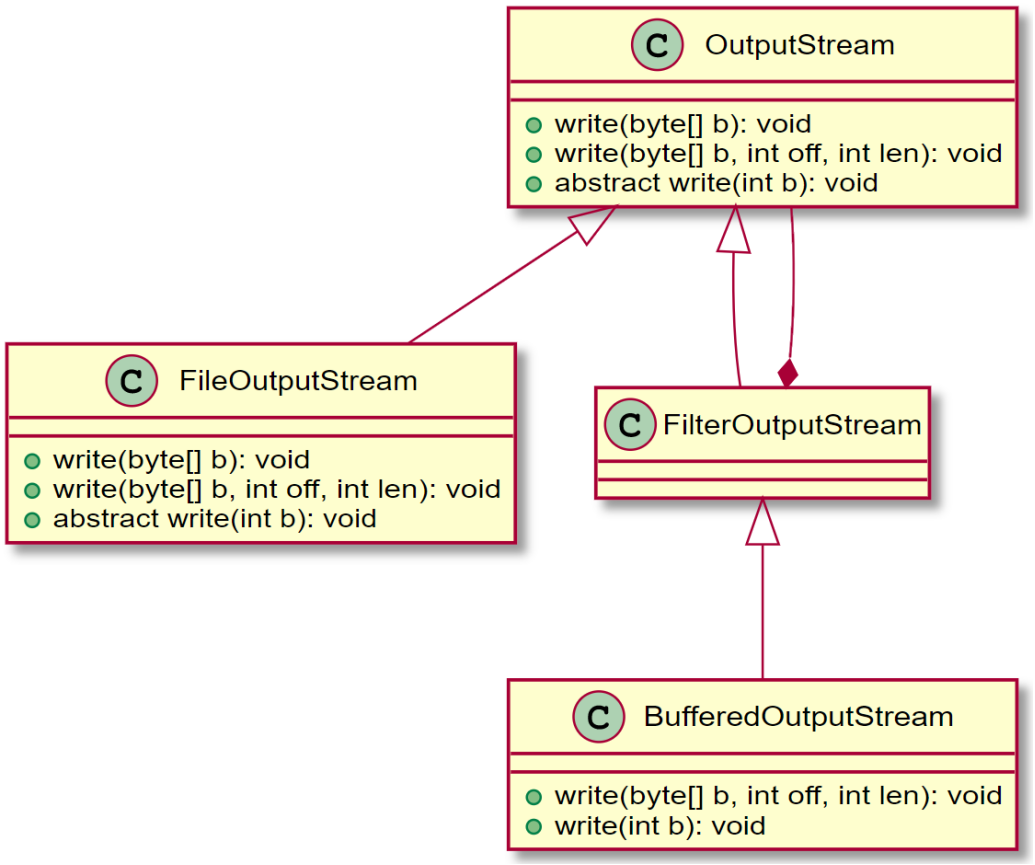

```
@startuml
class Reader {
    +abstract read(): int
}
class FileReader {
    +read(): int
}
class BufferedReader {
 +read(): int
 +readLine(): String
}
Reader <|-- BufferedReader
Reader <|-- InputStreamReader
InputStreamReader <|-- FileReader
```

```
@enduml
```



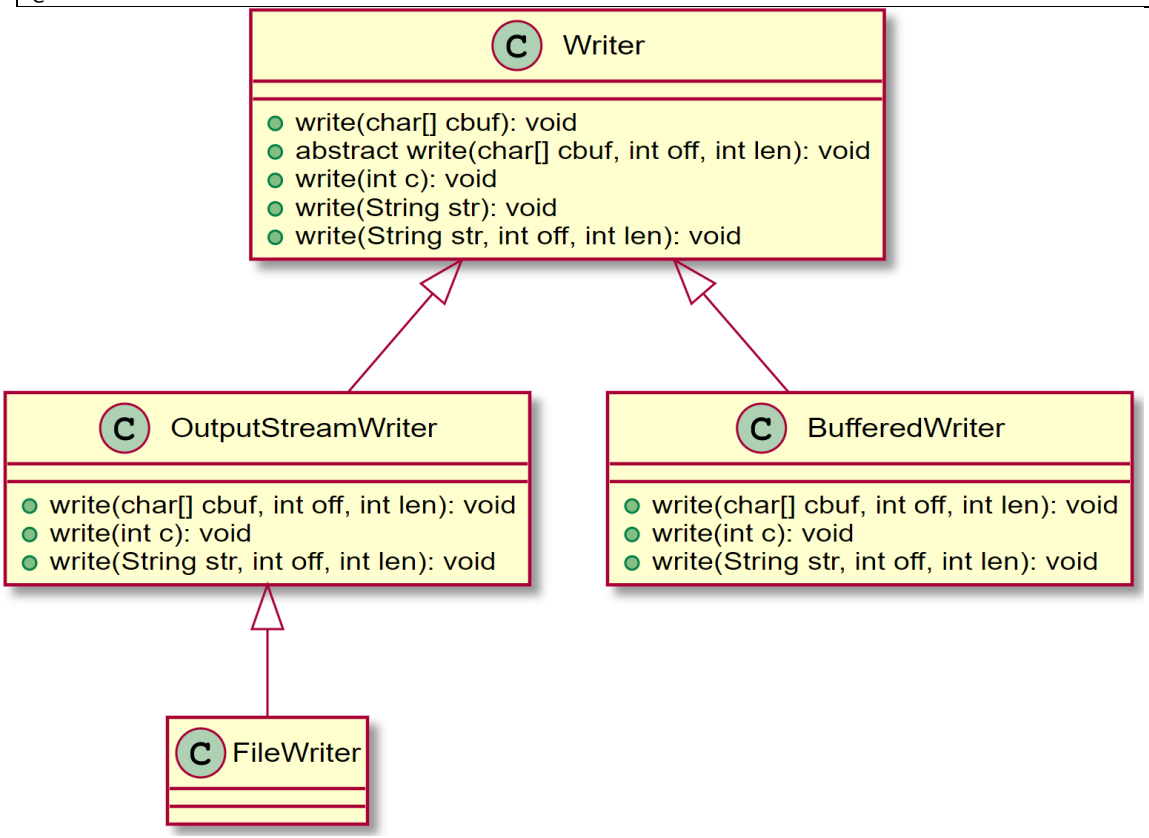

```
@startuml
class OutputStream {
    +write(byte[] b): void
    +write(byte[] b, int off, int len): void
    +abstract write(int b): void
}

class FileOutputStream {
    +write(byte[] b): void
    +write(byte[] b, int off, int len): void
    +abstract write(int b): void
}
class BufferedOutputStream {
    +write(byte[] b, int off, int len): void
    +write(int b): void
}
OutputStream <|-- FileOutputStream
OutputStream <|-- FilterOutputStream
FilterOutputStream *-- OutputStream
FilterOutputStream <|-- BufferedOutputStream
```

```
@enduml
```




```
@startuml
class Writer {
    +write(char[] cbuf): void
    +abstract write(char[] cbuf, int off, int len): void
    +write(int c): void
    +write(String str): void
    +write(String str, int off, int len): void
}
class OutputStreamWriter {
    +write(char[] cbuf, int off, int len): void
    +write(int c): void
    +write(String str, int off, int len): void
}
class FileWriter {
}
class BufferedWriter {
    +write(char[] cbuf, int off, int len): void
    +write(int c): void
```

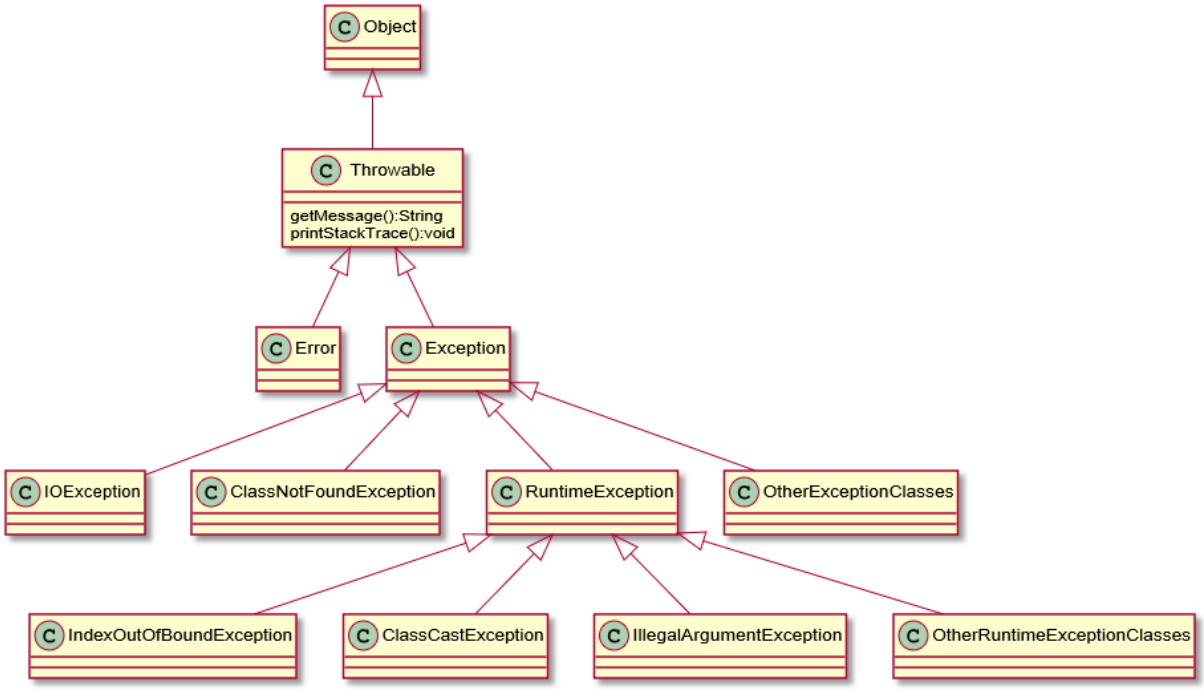
```
    +write(String str, int off, int len): void
}
Writer <|-- BufferedWriter
Writer <|-- OutputStreamWriter
OutputStreamWriter <|-- FileWriter
@enduml
```



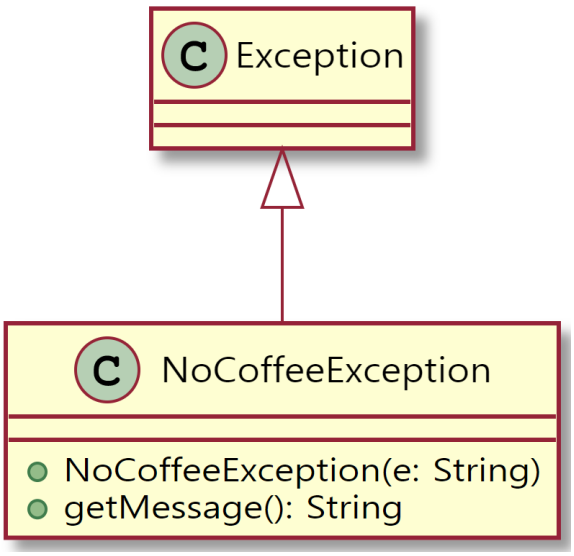
10 장

```
@startuml
class Throwable {
    getMessage():String
    printStackTrace():void
}
class Error
class Exception
class IOException
class ClassNotFoundException
class RuntimeException
class ArrayIndexOutOfBoundException
```

```
class ClassCastException
Object <|-- Throwable
Throwable <|-- Error
Throwable <|-- Exception
Exception <|-- RuntimeException
Exception <|-- IOException
Exception <|-- ClassNotFoundException
Exception <|-- OtherExceptionClasses
RuntimeException <|-- ArrayIndexOutOfBoundException
RuntimeException <|-- ClassCastException
RuntimeException <|-- IllegalArgumentException
RuntimeException <|-- OtherRuntimeExceptionClasses
@enduml
```




```
@startuml
class NoCoffeeException extends Exception {
    +NoCoffeeException(e: String)
    +getMessage(): String
}
@enduml
```




```
@startuml
class CoffeeMachine {
    -amountOfCoffeeBeans: int
    +Coffee(beans: int)
    +getAmountOfCoffeeBeans(): int
    +addCoffeeBeans(beans: int): void
    +brew() throws NoCoffeeException: void
}
@enduml
```