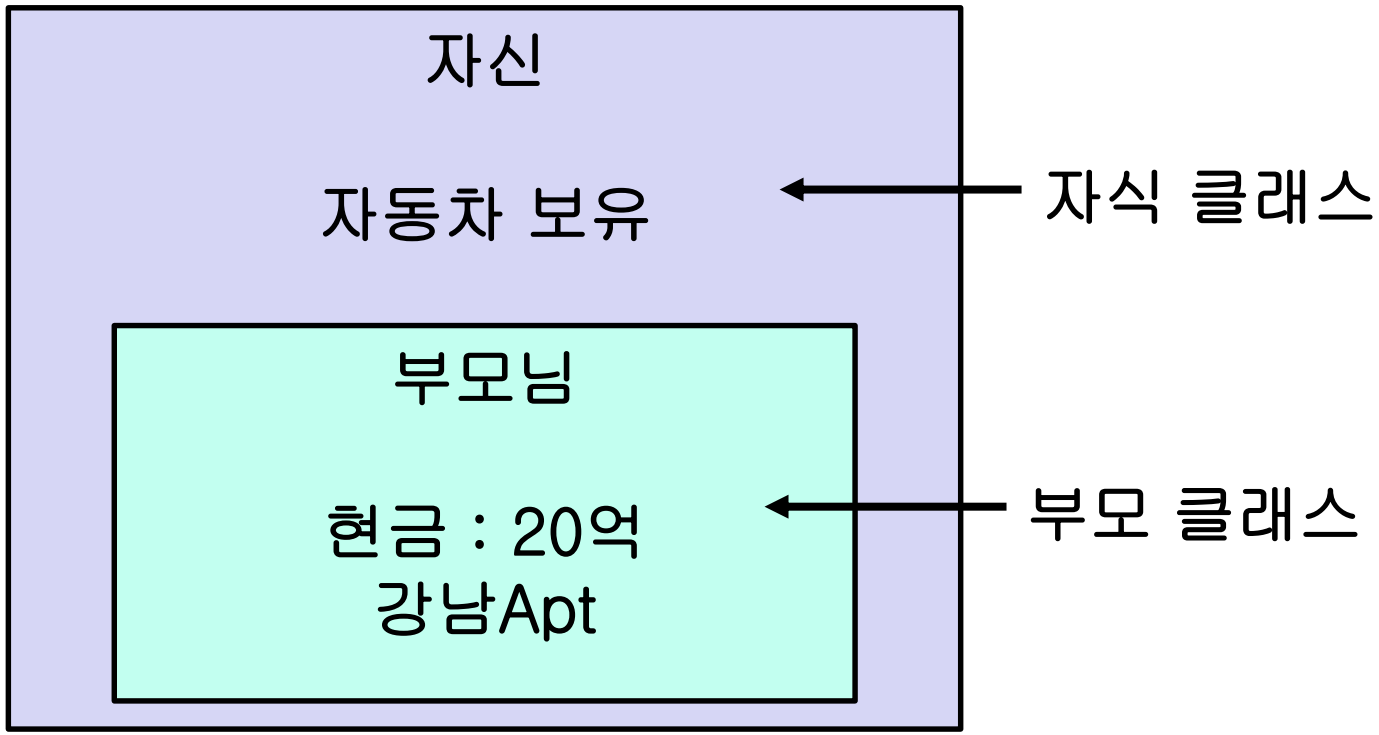
JAVA 프로그램 실습

Class 상속

경복대학교
소프트웨어융합과
배희호 교수

상속 예제 1

- 부모님(Parent)이 현금(money) 20억원과, 강남에 아파트(house)를 갖고 계신다.
- 나는 노원구의 전세에 살면서 자동차(car)를 갖고 있다.
- 상속의 개념으로 프로그램 해보자



상속 예제 1

- Parent Class

```
public class Parent {
    protected int money = 2000000000;
    String house = "아파트";
    public String address = "강남";
}
```

상속 예제 1

- Child Class

```
public class Child extends Parent {
    private String car = "소나타";
    public String address = "서울시 노원구";

    public String getCar() {
        return car;
    }
}
```

상속 예제 1

- Main Class

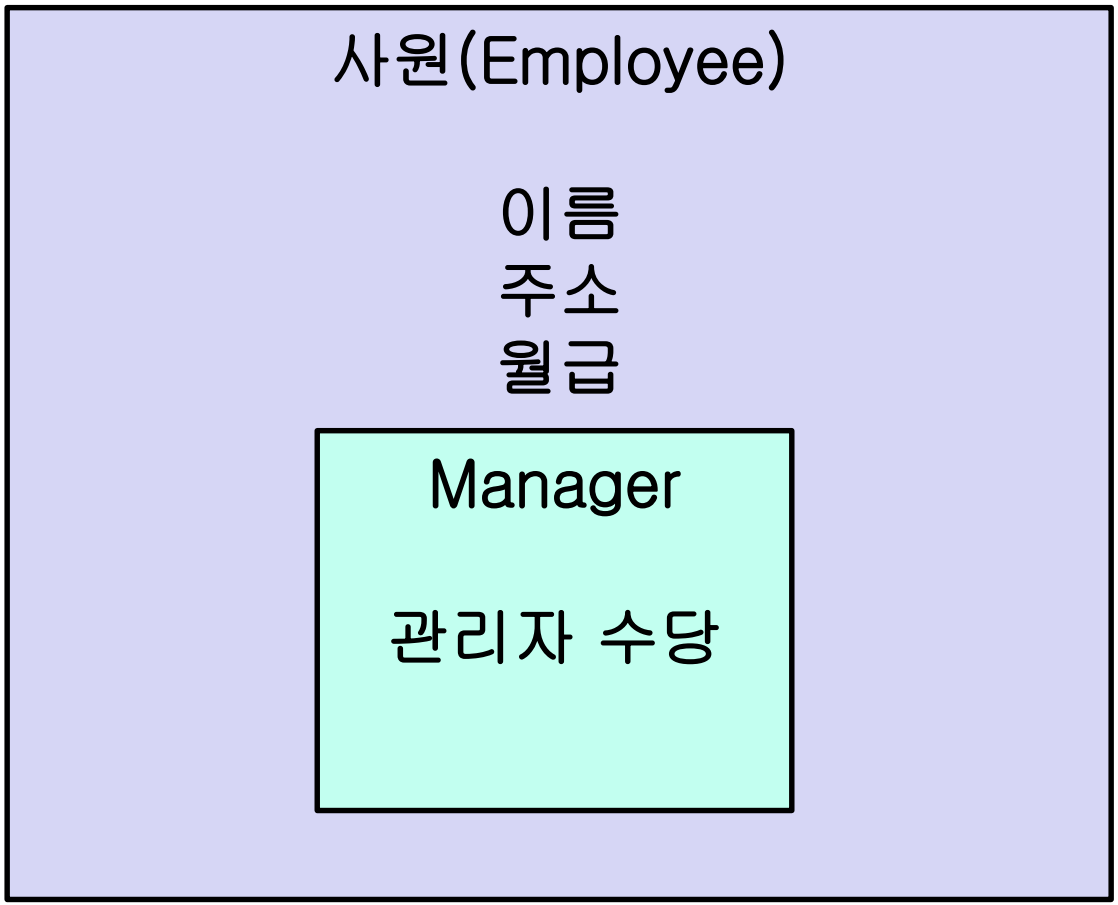
```
public static void main(String[] args) {
        Child child = new Child();
        if(child instanceof Parent) {
            System.out.println("상속 관계");
        }
        if(child instanceof Object) {
            System.out.println("상속 관계");
        }

        System.out.println(child.getCar());
        System.out.println(child.money);
        System.out.println(child.house);
        System.out.println(child.address);
        Parent parent = new Parent();
    //   System.out.println(parent.getCar());
        System.out.println(parent.money);
}
```

상속 예제 1

- 부모 클래스의 Member 변수는 자동으로 자식 클래스에 상속되어 들어감
- 부모 클래스가 가진 변수와 같은 이름의 변수를 자식 클래스에서 선언하면, 부모 클래스의 변수는 상속되지 않고, 자식 클래스에서 정의한 변수가 사용됨
- 자식 클래스는 부모 클래스 변수를 물려받아 사용할 수 있음
- private로 정의된 변수는 상속되긴 하지만 접근할 수 있는 권한이 없음

상속 예제 2

- 사원과 Manager 개념 정리



상속 예제 2

- 실행 결과

```
이름 : 홍길동, 주소 : 서울시 노원구, 급여 : 2560000
이름 : 김철수, 주소 : 경기도 남양주시, 급여 : 2100000, 보너스 : 670000


종료 코드 0(으)로 완료된 프로세스
```

상속 예제 2

- Employee 클래스

```
public class Employee {
    private String name;
    private String address;
    protected int salary;

    public Employee(String name, String address, int salary) {
        this.name = name;
        this.address = address;
        this.salary = salary;
    }

    public void setAddress(String address) {
        this.address = address;
    }
    @Override
    public String toString() {
        return  "이름 : " + name + ", 주소 : " + address + ", 급여 : " + salary;
    }
}
```

상속 예제 2

- Manager 클래스

```
public class Manager extends Employee{
    private int bonus;

    public Manager(String name, String address, int salary, int bonus) {
        super(name, address, salary);
        this.bonus = bonus;
    }

    @Override
    public String toString() {
        return super.toString() + ", 보너스 : " + bonus;
    }
}
```

상속 예제 2

- Main.JAVA

```
public static void main(String[] args) {
    Employee hong = new Employee("홍길동", "서울시 노원구", 2560000);
    Manager kim = new Manager("김철수", "경기도 가평군", 2100000, 670000);

    kim.setAddress("경기도 남양주시");
    System.out.println(hong);
    System.out.println(kim);
}
```

상속 예제 2

- 부모 클래스가 가지고 있는 메소드가 자식 클래스로 상속되어 자식 클래스에서 사용 가능
- 메소드의 Overriding
  - 부모 클래스의 메소드를 재사용하지 않고 새롭게 정의하여 사용하는 것
  - 부모 클래스가 가진 메소드는 상속되지 않음

상속 예제 2

- Overloading
  - 같은 클래스 내에 같은 이름의 생성자나 메소드를 사용하는 행위
  - 매개 변수의 개수와 타입이 달라야 함
- Overriding
  - 상속 관계에 있는 클래스들간에 같은 이름의 메소드를 정의하는 행위
  - 기존 클래스의 메소드 구현 부분만 약간 변화시켜 새로운 메소드를 생성할 수 있음
  - 매개 변수의 개수와 타입이 같아야 함
- Overloading과 Overriding은 객체지향 언어의 주요 개념인 Polymorphism(다형성)을 구현

상속 예제 2-1

- 실행 결과

```
이름 : 홍길동(생일 : 1992년 3월 20일)
주소 : 서울시, 급여 : 2560000
이름 : 김철수(생일 : 2001년 5월 20일)
주소 : 경기도, 급여 : 2100000, 보너스 : 670000

종료 코드 0(으)로 완료된 프로세스
```

상속 예제 2-1

- Date 클래스

```
public class Date {
    private int year;
    private int month;
    private int day;

    public Date(int year, int month, int day) {
        this.year = year;
        this.month = month;
        this.day = day;
    }

    @Override
    public String toString() {
        return year + "년 " + month + "월 " + day + "일";
    }
}
```

상속 예제 2-1

- Employee 클래스

```
public class Employee {
    public String name;
    private Date birthday;
    String address;
    protected int salary;

    public Employee(String name, Date birthday, String address, int salary) {
        this.name = name;
        this.birthday = birthday;
        this.address = address;
        this.salary = salary;
    }

    @Override
    public String toString() {
        return  "이름 : "  + name + "(생일 : " + birthday.toString() + ")₩n주소 : " +
                                                address + ", 급여 : " + salary;
    }
}
```

상속 예제 2-1

- Employee 클래스

```
public class Manager extends Employee{
    private int bonus;

    public Manager(String name, Date birthday, String address,
                                                    int salary, int bonus) {
        super(name, birthday, address, salary);
        this.bonus = bonus;
    }

    @Override
    public String toString() {
        return super.toString() + ", 보너스 : " + bonus;
    }
}
```

상속 예제 2-1

- Main.JAVA

```
public static void main(String[] args) {
        Employee hong = new Employee("홍길동",
                                        new Date(1992, 3, 20), "서울시", 2560000);
        Manager kim = new Manager("김철수",
                                    new Date(2001, 5, 20),"경기도", 2100000, 670000);

        System.out.println(hong);
        System.out.println(kim);
}
```

상속 예제 2-1

- Data 클래스를 논리적으로 점검하는 기능을 추가할 것

상속 예제 3

- 다음의 결과를 보자

```
대학생
이름 : 홍길동, 학년 : 3, 학점 : B
중학생
이름 : 이미숙, 학년 : 2, 점수 : 80
대학생
이름 : 경복대, 학년 : 2, 학점 : A
```

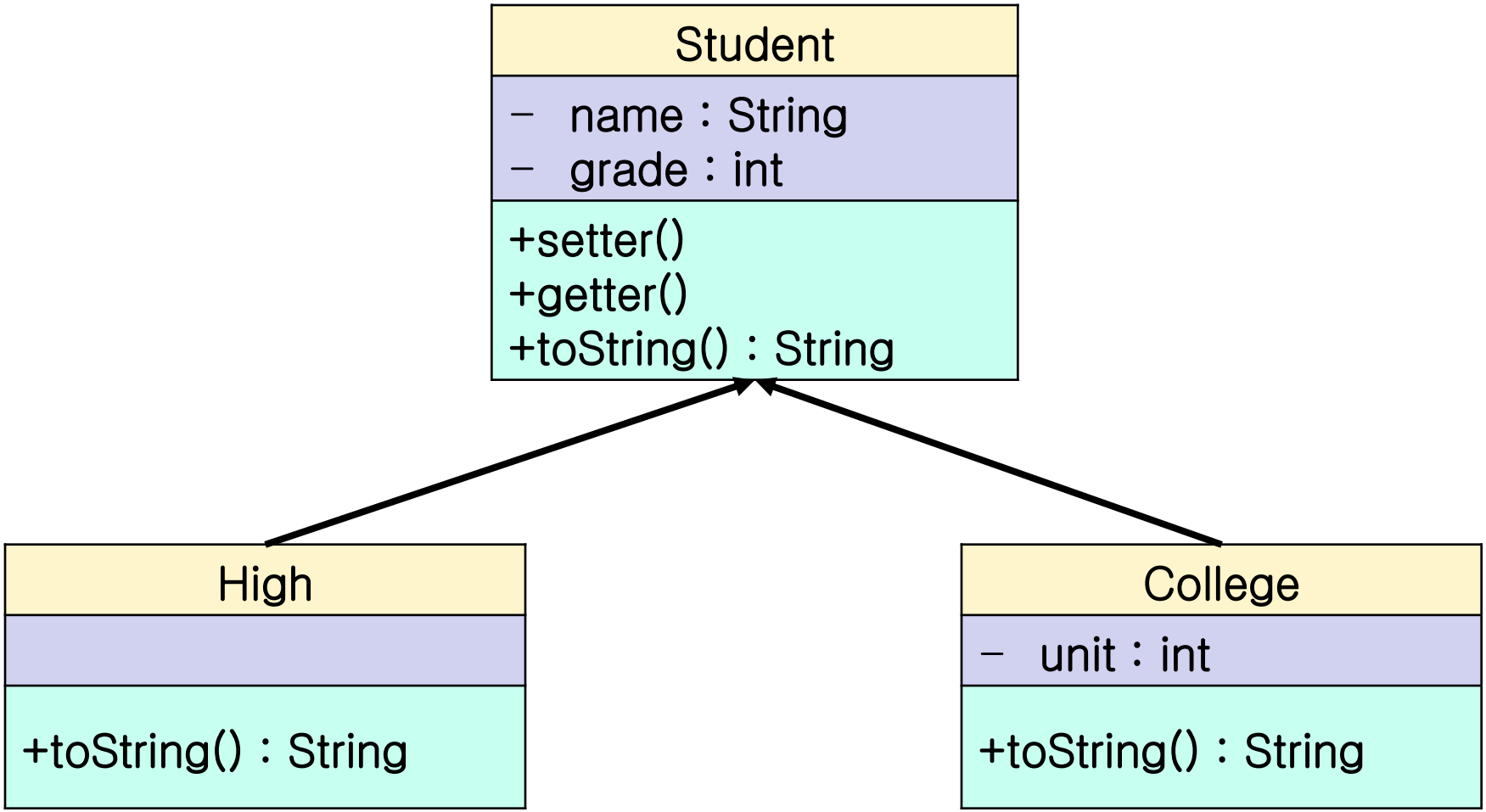
- 대학생 성적 처리와 고등학생 성적 처리를 따로 해야 하나?
  - 대학생과 고등학생의 공통점과 차이점 구분
  - 공통점을 상속의 개념에서 부모 클래스
  - 차이점은 각각의 자신 클래스

상속 예제 3

- 분석
  - 공통점 (Student 클래스)
    - 이름과 학년
  - 고등학생 (High 클래스)
    - 점수
  - 대학생 (College 클래스)
    - 등급

상속 예제 3

- 고등학생 is-a 학생, 대학생 is a 학생



상속 예제 3

- Student 클래스

```
public class Student {
    private String name;
    private int grade;

    public Student(String name, int grade) {
        this.name = name;
        this.grade = grade;
    }

    @Override
    public String toString() {
        return "이름 : " + name + ", 학년 : " + grade;
    }
}
```

상속 예제 3

- High 클래스

```
public class High extends Student{
    public High(String name, int grade) {
        super(name, grade);
    }

    @Override
    public String toString() {
        return super.toString();
    }
}
```

명시적인 부모
생성자 호출 시에는
반드시
첫 라인에서 호출

상속 예제 3

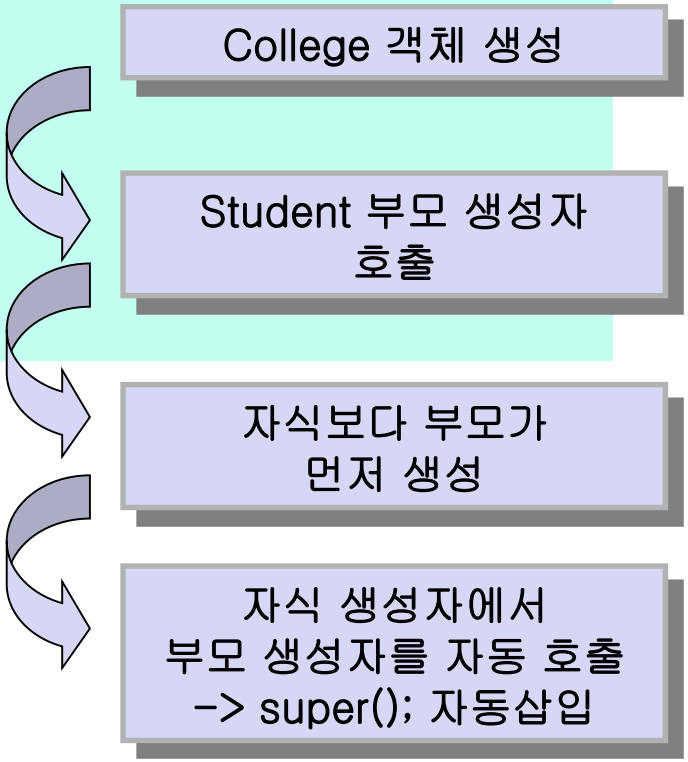
- College 클래스

```
public class College extends Student{
    private int unit;

    public College() {
        this("홍길동", 1, 20);
    }

     public College(String name, int grade, int unit) {
        super(name, grade);
        this.unit = unit;
    }

    @Override
    public String toString() {
        return super.toString() + ", 학점 : " + unit;
    }
}
```

상속 예제 3

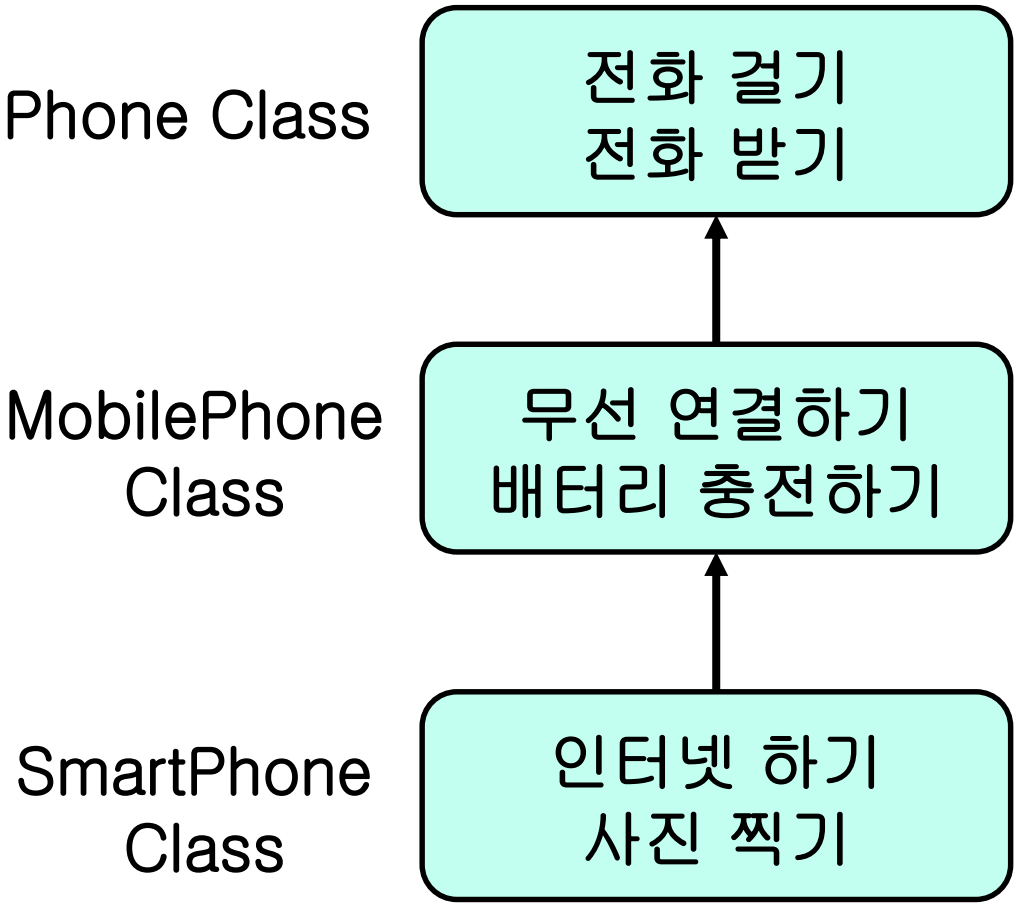
- Main.JAVA

```
public static void main(String[] args) {
    Student hong = new High("홍길동", 2);
    High kim = new High("김철수", 1);
    College lee = new College("이순신", 3, 24);

    System.out.println(hong);
    System.out.println(kim);
    System.out.println(lee);
}
```



상속 예제 4

- Phone, MobilePhone, SmartPhone



상속 예제 4

```
class Phone {
    public void call() {   }
    public void answer() {   }
}
```

```
class MobilePhone extends Phone {
    public void wirelessconnect() {   }
    public void charge() {   }
}
```

```
class SmartPhone extends MobilePhone {
    public void internet() {   }
    public void takepicture() {   }
}
```

상속 예제 5

- 상속은 여러 단계로 이루어질 수 있음

Vehicle

Car

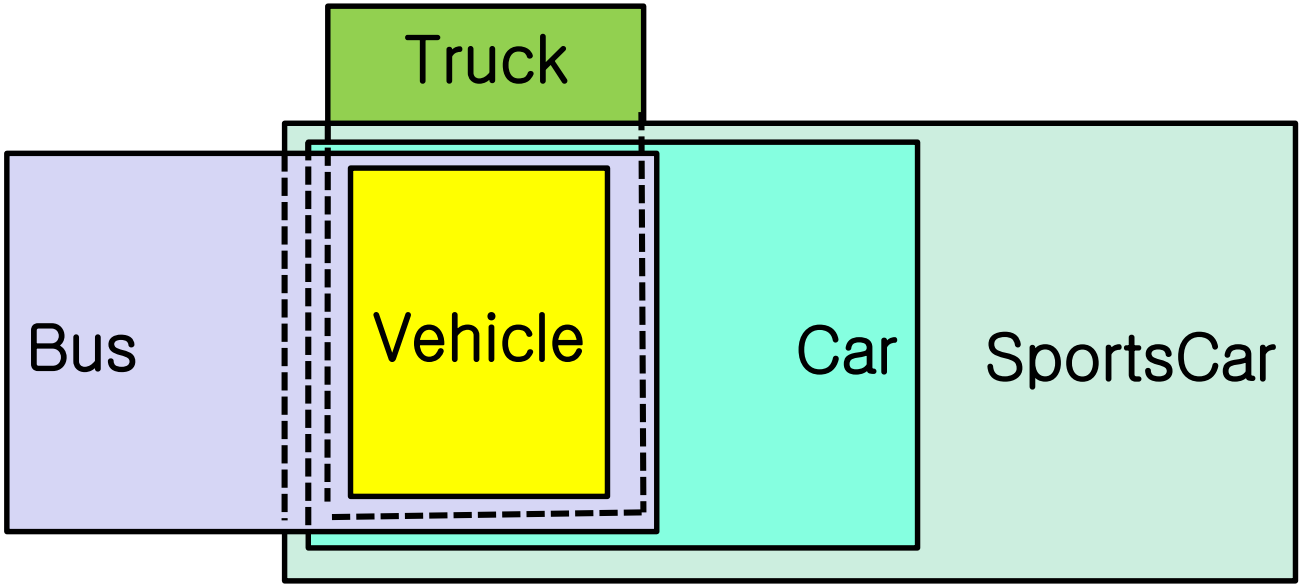
Truck

Bus

SportsCar

상속 예제 5

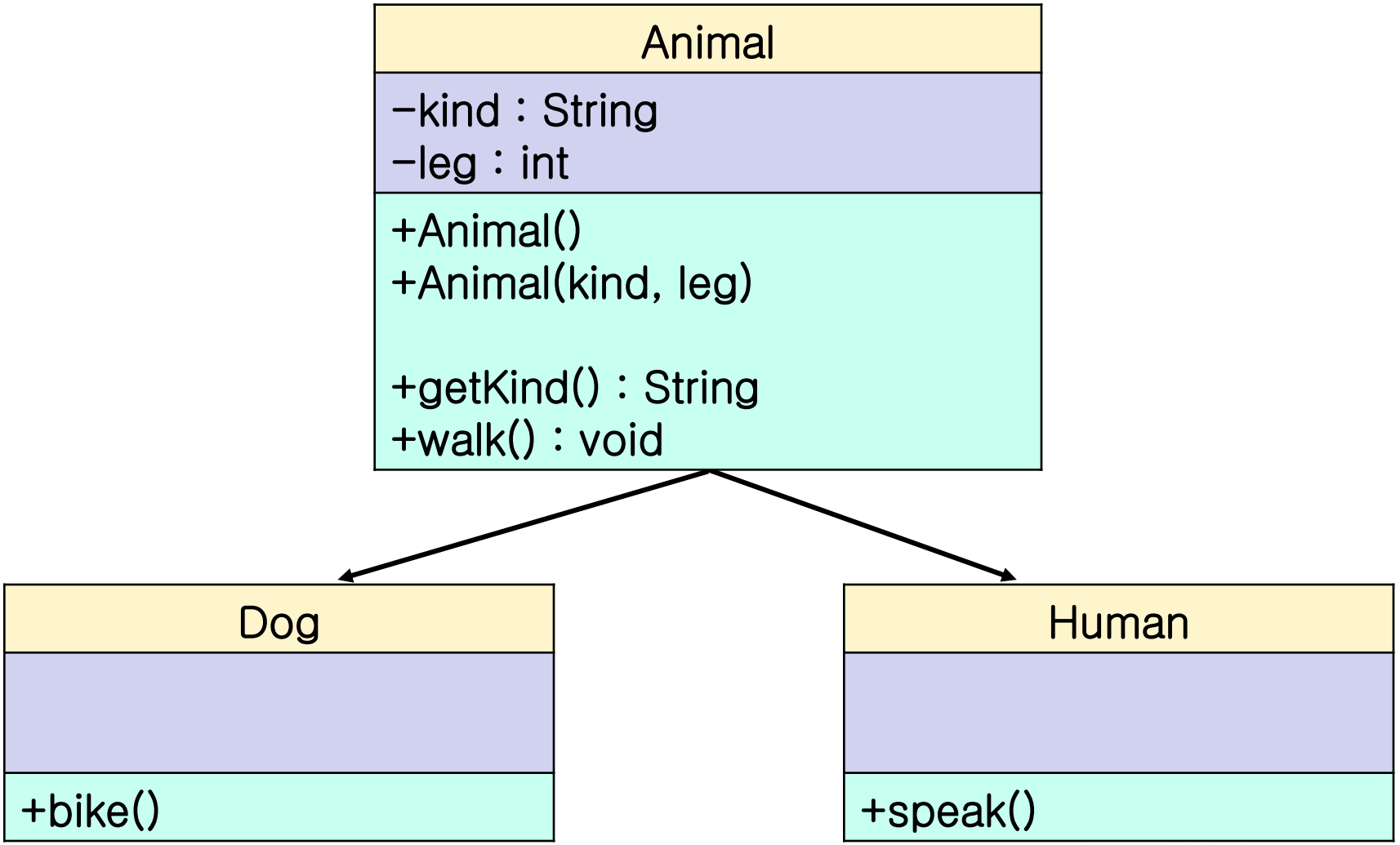
```
class Vehicle { }
class Car extends Vehicle { }
class SportsCar extends Car { }
class Truck extends Vehicle {  }
class Bus extends Vehicle {   }
```



상속 예제 6

- 동물(Animal)과 개(Dog)와 사람(Human)이란 클래스를 서로 상속이란 개념을 도입해서 설계해보자
- 슈퍼 클래스로는 동물(Animal)을 두고, 슈퍼 클래스에는 멤버 변수로는 어떤 종인지의 구분을 위해서 kind와 다리의 개수를 저장하기 위한 leg를 지정
- 슈퍼 클래스의 멤버 메소드로는 getKind()와 walk()를 생성
  - getKind() 메소드는 어떤 동물인지를 알려주는 메소드
  - walk() 메소드로 어떻게 걷는지를 알려주는 메소드

상속 예제 6



상속 예제 6

```
class Car {
   public static String brand = "Hyundai";
   int maxSpeed;
   private String assembler;
}

class Avante extends Car {
   public String getBrand() {    // public 이기 때문에 접근 가능
        return brand;
   }

   public String getAssembler() { // private 이기 때문에 접근 불가
        return assembler;
   }
}
```

상속 예제 6

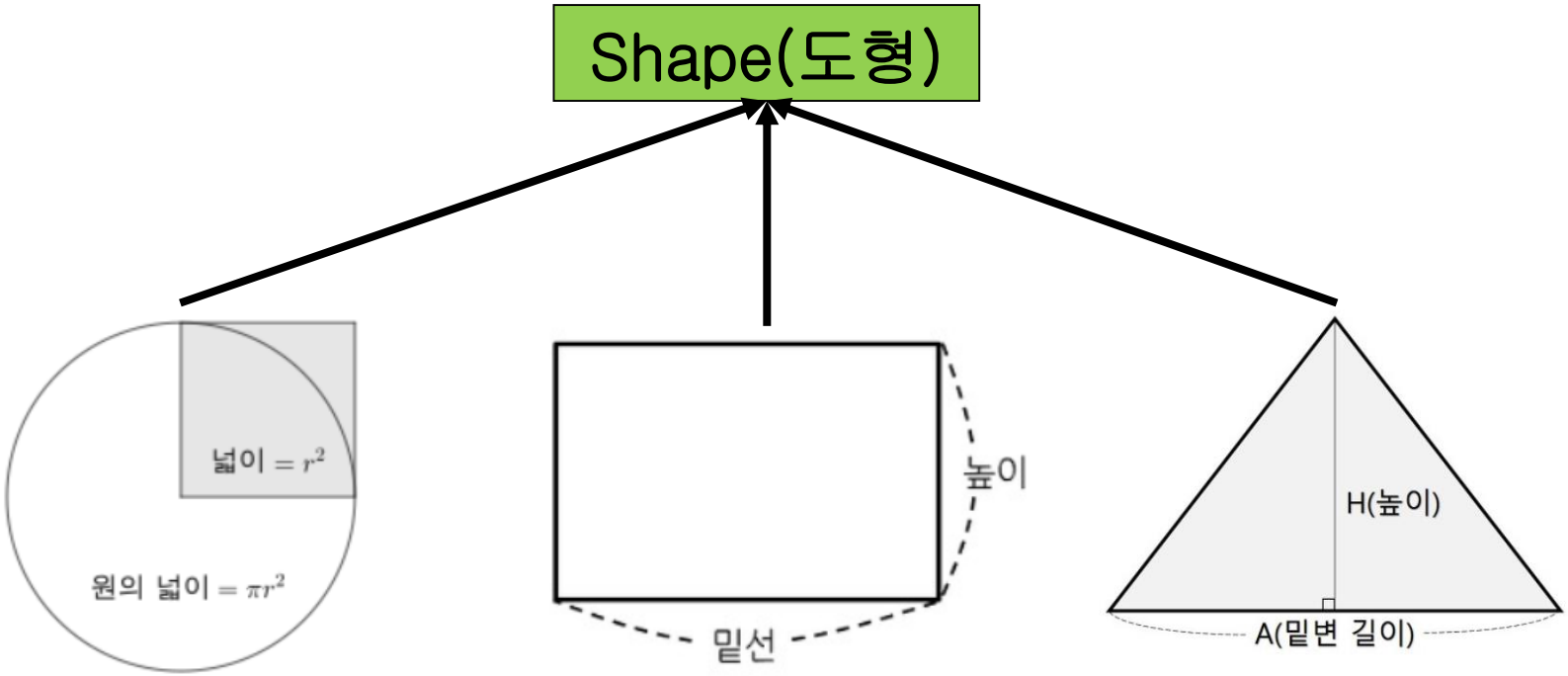
```
class Car {
    public static String brand = "Hyundai";
    protected int maxSpeed;
    private String assembler;
}

class Avante extends Car {
    public int getMaxSpeed() { // protected 는 접근 가능
        return super.maxSpeed;
    }
}
```

상속 예제 6

```
class Car {
   public static String brand = "Hyundai";
   int maxSpeed;

   public Car() {
        System.out.println("생성");
   }
}

class Avante extends Car {
   public Avante() {
      super();
   }

   public String getBrand() { // Error // 생성자에서만 호출해야 함
      super();
   }
}
```

상속 예제 7

- 원(Circle), 사각형(Rectangle)과 삼각형(Triangle) 도형의 면적을 구하는 Program을 만들어보자



상속 예제 7

- Shape.JAVA

```
public class Shape {
    private int length;
    private double width;

    public Shape(int length) {
        this.length = length;
        width = 0.0f;
    }

    public Shape(double width) {
        this.width = width;
        length = 0;
    }

    public int getLength() {
        return length;
    }
```

상속 예제 7

- Shape.JAVA

```
    public double getWidth() {
        return width;
    }

    public void setLength(int length) {
        this.length = length;
    }

    public void setWidth(double width) {
        this.width = width;
    }
}
```

상속 예제 7

- Circle.JAVA

```
public class Circle extends Shape{

    public Circle(double radius) {
        super(radius);
        super.setLength(0);
    }

    public Circle(int radius) {
        super(radius);
        super.setWidth(0.0);
    }

    public double area(boolean flag) {
        if (flag)
            return Math.PI * super.getWidth() * super.getWidth();
        else
            return Math.PI * super.getLength() * super.getLength();
    }
```

상속 예제 7

- Circle.JAVA

```
    @Override
    public String toString() {
        String result;
        if (super.getWidth() == 0.0) {
            result = String.format("원의 반지름 : %d Cm₩n", super.getLength());
            result += String.format("원의 면적 : %.2f Cm₩n", area(false));
        } else {
            result = String.format("원의 반지름 : %.2f Cm₩n", super.getWidth());
            result += String.format("원의 면적 : %.2f Cm₩n", area(true));
        }
        return result;
    }
}
```

상속 예제 7

- Rectangle.JAVA

```
public class Rectangle extends Shape{
    private double height;
    private int height1;

    public Rectangle(double width, double height) {
        super(width);
        this.height = height;
        height1 = 0;
    }

    public Rectangle(int width, int height) {
        super(width);
        this.height = 0.0;
        this.height1 = height;
    }
    public double area() {
        return super.getWidth() * height;
    }
```

상속 예제 7

- Rectangle.JAVA

```
    public int area1() {
        return super.getLength() * height1;
    }
    @Override
    public String toString() {
        String result;
        if (height == 0.0) {
            result = String.format("가로 길이 : %d Cm₩n", super.getLength());
            result += String.format("세로 길이 : %d Cm₩n", height1);
            result += String.format("사각형의 면적 : %d Cm₩n", area1());
        } else {
            result = String.format("가로 길이 : %.2f Cm₩n", super.getWidth());
            result += String.format("세로 길이 : %.2f Cm₩n", height);
            result += String.format("사각형의 면적 : %.2f Cm₩n", area());
        }
        return result;
    }
}
```

상속 예제 7

- Triangle.JAVA

```
public class Triangle extends Shape{
    private int height;
    private float height1;

    public Triangle(float length, float height) {
        super(length);
        this.height = 0;
        this.height1 = height;
    }

    public Triangle(int length, int height) {
        super(length);
        height1 = 0.0f;
        this.height = height;
    }
    public double area() {
        return height * super.getLength() / 2.0;
    }
```

상속 예제 7

- Main.JAVA

```
public static void main(String[] args) {
    Circle circle = new Circle(3);
    System.out.println(circle);
    circle = new Circle(3.5);
    System.out.println(circle);

    Triangle triangle = new Triangle(3, 4);
    System.out.println(triangle);
    triangle = new Triangle(3.5f, 4.5f);
    System.out.println(triangle);

    Rectangle rectangle = new Rectangle(3, 4);
    System.out.println(rectangle);
    rectangle = new Rectangle(3.5, 4.5);
    System.out.println(rectangle);
}
```

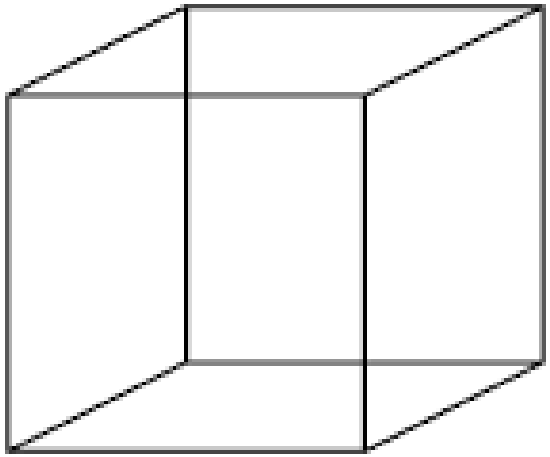
상속 예제 7

- Triangle.JAVA

```
    public double area1() {
        return super.getWidth() * height1 / 2.0;
    }
    @Override
    public String toString() {
        String result;
        if (height1 == 0.0f) {
            result = String.format("밑변 길이 : %d Cm₩n", super.getLength());
            result += String.format("높이 길이 : %d Cm₩n", height);
            result += String.format("삼각형의 면적 : %.2f Cm₩n", area());
        } else {
            result = String.format("밑변 길이 : %.2f Cm₩n", super.getWidth());
            result += String.format("높이 길이 : %.2f Cm₩n", height1);
            result += String.format("삼각형의 면적 : %.2f Cm₩n", area1());
        }
        return result;
    }
}
```

Cube 부피 계산하기

- 사각 기둥(Cube)의 부피(Volume)을 계산해보자



Cube
– width : int – length : int – height : int
+volume( ) : int +toString() : String

- 사각기둥을 위한 Cube Class를 만들어보자
- 사각기둥은 사각형(Rectangle)을 상속 받아 Cube Class를 만들어보자

Cube 부피 계산하기(I)

- Cube.JAVA

```
public class Cube {
    private int width;
    private int length;
    private int height;

    public Cube(int width, int length, int height) {
        this.width = width;
        this.length = length;
        this.height = height;
    }

    public int volume() {
        return width * length * height;
    }
```

Cube 부피 계산하기(I)

- Cube.JAVA

```
    @Override
    public String toString() {
        return (((width == length) && (width == height)) ? "정" : "직") +
                "사각기둥₩n 가로길이 : " + width +
                "₩n 세로길이 : " + length +
                "₩n 높이 : " + height + "₩n부피 : " + volume();
    }
}
```

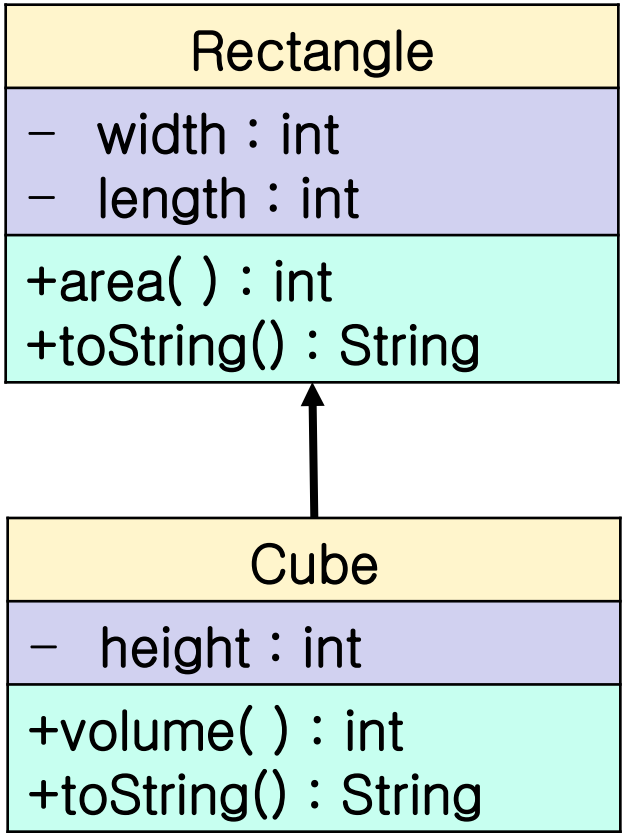
Cube 부피 계산하기(I)

- Main.JAVA

```
public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("사각기둥의 가로 길이 입력 : ");
        int width = keyboard.nextInt();
        System.out.print("사각기둥의 세로 길이 입력 : ");
        int length = keyboard.nextInt();
        System.out.print("사각기둥의 높이 길이 입력 : ");
        int height = keyboard.nextInt();

        Cube cube = new Cube(width, length, height);
        System.out.println(cube);
}
```

Cube 부피 계산하기(II)

- Inheritance



Cube 부피 계산하기(II)

- Rectangle.JAVA

```
public class Rectangle {
    private int width;
    private int lenght;

    public Rectangle(int width, int length) {
        this.width = width;
        this.length = length;
    }

    public int getWidth() {
        return width;
    }

    public int getLength() {
        return length;
    }
```

Cube 부피 계산하기(II)

- Rectangle.JAVA

```
    public int area(){
        return width * length;
    }

    @Override
    public String toString() {
        return ((width == length) ? "정" : "직") +
                "사각형의 가로길이 = " + width +
                ", 세로길이 = " + length + "₩n" +
                " 면적 = " + area() + "₩n";
    }
}
```

Cube 부피 계산하기(II)

- Cube.JAVA

```
public class Cube extends Rectangle{
    private int height;

    public Cube(int width, int length, int height) {
        super(width, length);
        this.height = height;
    }

    public int volume() {
        return super.area() * height;
    }
```

Cube 부피 계산하기(II)

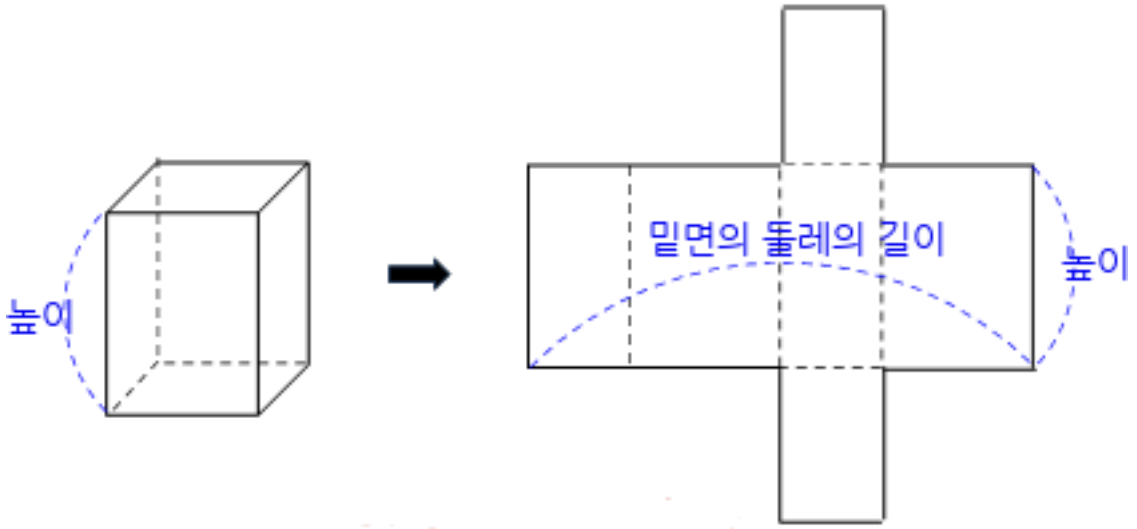
- Cube.JAVA

```
    @Override
    public String toString() {
        return ((super.getWidth() == super.getLength() &&
                                        super.getWidth() == height) ? "정" : "직") +
                "사각기둥₩n 가로길이 : " + super.getWidth() +
                "₩n 세로길이 : " +super.getLength() +
                "₩n 높이 : " + depth +"₩n부피 : " + volume();
    }
}
```

Cube 겉넓이 계산하기[심화]

- 사각기둥의 겉넓이를 구해보자

사각기둥의 겉넓이 = 2 × (밑 넓이) + (옆 넓이)



- 앞의 문제에서 만든 Rectangle 클래스를 이용하면 좀더 편리하지 않을까?

Cube 겉넓이 계산하기[심화]

- Cube.JAVA

```
public class Cube extends Rectangle{
    private int height;

    public Cube(int width, int length, int height) {
        super(width, length);
        this.height = height;
    }

    public int volume() {
        return super.area() * height;
    }

    public int area() {
        int 밑넓이 = super.area();
        int 옆넓이 = (super.getWidth() + super.getLength()) * height;
        return 2 * (밑넓이 + 옆넓이);
    }
```

Cube 겉넓이 계산하기[심화]

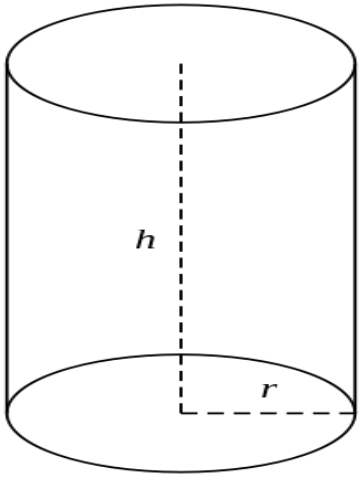
- Cube.JAVA

```
    @Override
    public String toString() {
        return ((super.getWidth() == super.getLength() &&
                                        super.getWidth() == height) ? "정" : "직") +
                "사각기둥₩n 가로길이 : " + super.getWidth() +
                "₩n 세로길이 : " +super.getLength() +
                "₩n 높이 : " + height +"₩n 부피 : " + volume() + “₩u33A4” +
                "₩n 표면적 : " + area() +”₩u33A0”;
    }
}
```

```
사각기둥의 가로 길이 입력 : 2
사각기둥의 세로 길이 입력 : 3
사각기둥의 높이 길이 입력 : 4
직사각기둥
 가로길이 : 2
 세로길이 : 3
 높이 : 4
 부피 : 24㎥
 표면적 : 52㎡
```

Cylinder 부피 구하기

- 원기둥 밑면의 반지름이 r, 높이가 h일 때
- 원기둥의 부피 = (밑 넓이) × (높이) = 2πrh



Cylinder
– radius : float – height : float
+volume( ) : double +toString() : String

- 원기둥 클래스인 Cylinder Class를 만들어보자
- 원기둥 클래스인 Cylinder Class를 Circle Class로부터 상속 받아 만들어보자

Cylinder 부피 구하기(I)

- Cylinder.JAVA

```
public class Cylinder {
    private float radius;
    private float height;

    public Cylinder(float radius, float height) {
        this.radius = radius;
        this.height = height;
    }
    private double volume() {
        return (Math.PI * radius * radius) * height;
    }
    @Override
    public String toString() {
        return "원기둥의 반지름 : " + radius +
                "₩n원기둥의 높이 : " + height +
                "₩n원기둥의 부피 : " + volume();
    }
}
```

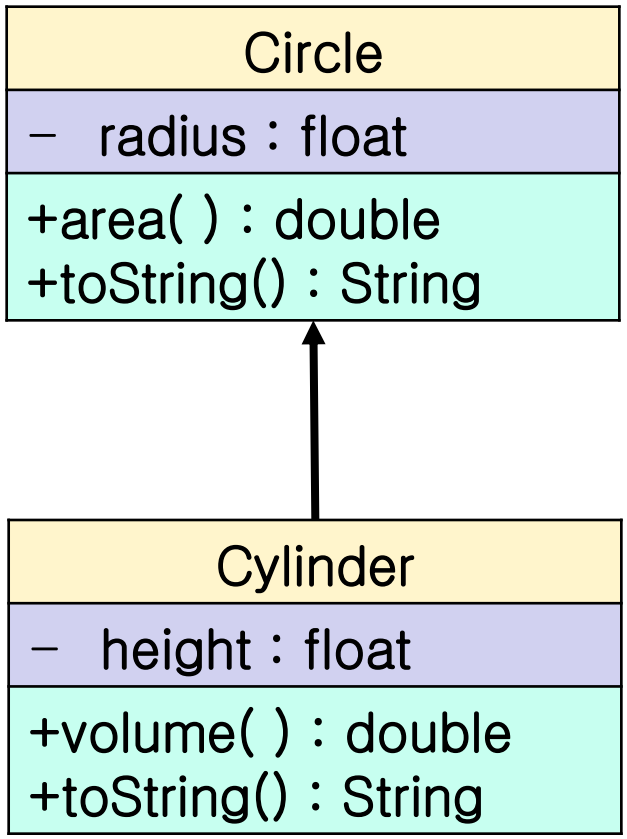
Cylinder 부피 구하기(I)

- Main.JAVA

```
public class Main {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        System.out.print("원기둥의 반지름 입력 : ");
        float radius = keyboard.nextFloat();
        System.out.print("원기둥의 높이 입력 : ");
        float height = keyboard.nextFloat();

        Cylinder cylinder = new Cylinder(radius, height);
        System.out.println(cylinder);
    }
}
```

Cylinder 부피 구하기(상속)

- Inheritance



Cylinder 부피 구하기(II)

- Circle.JAVA

```
public class Circle {
    float radius;

    public Circle(float radius) {
        this.radius = radius;
    }

    public double area() {
        return Math.PI * radius * radius;
    }

    @Override
    public String toString() {
        return "원의 반지름 : " + radius +
                "₩n원의 면적 : "  + area();
    }
}
```

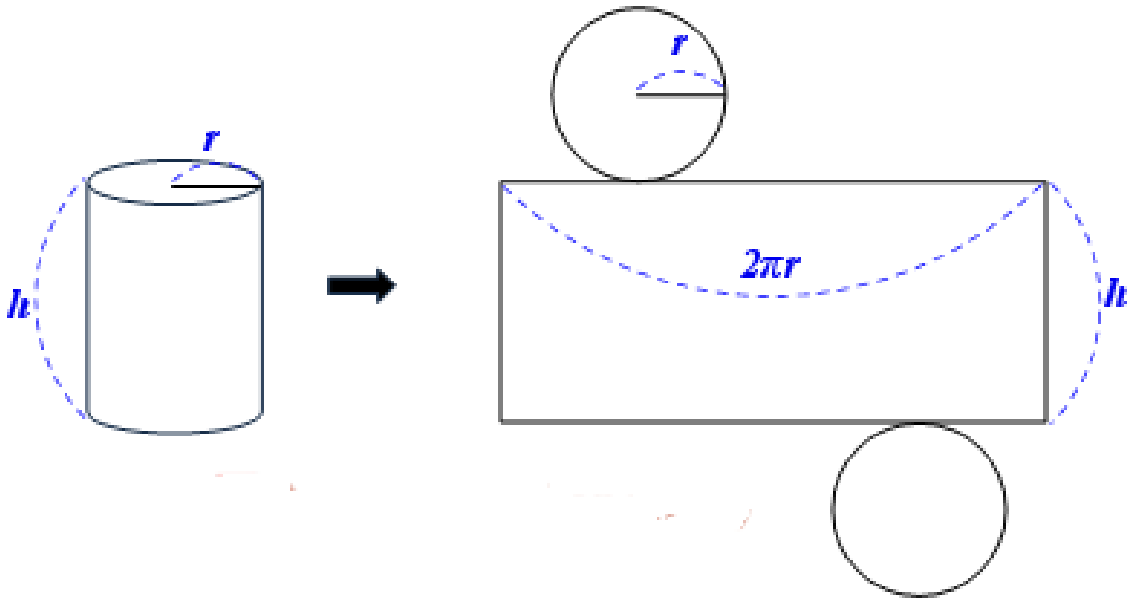
Cylinder 부피 구하기(II)

- Cylinder.JAVA

```
public class Cylinder extends Circle {
    private float height;

    public Cylinder(float radius, float height) {
        super(radius);
        this.height = height;
    }

    private double volume() {
        return super.area() * height;
    }

    @Override
    public String toString() {
        return "원기둥의 반지름 : " + radius +
               "₩n원기둥의 높이 : " + height +
               "₩n원기둥의 부피 : " + volume();
    }
}
```

Cylinder 표면적 구하기[심화]

- 원기둥 밑면의 반지름이 r, 높이가 h일 때
- 원기둥의 겉넓이 = 2 × (밑넓이) + (옆넓이) = $2\pi r^2 + 2\pi rh$



- 원기둥 클래스인 Cylinder Class를 Circle Class로부터 상속 받아 여러분이 만들어보자

Cylinder 표면적 구하기[심화]

- Cylinder.JAVA

```
public class Cylinder extends Circle {
    private float height;

    public Cylinder(float radius, float height) {
        super(radius);
        this.height = height;
    }

    private double volume() {
        return super.area() * height;
    }

    public double area() {
        double 원넓이 = super.area();
        double 옆면넓이 = Math.PI * radius * height * 2;
        return (원넓이 * 2) + 옆면넓이;
    }
```

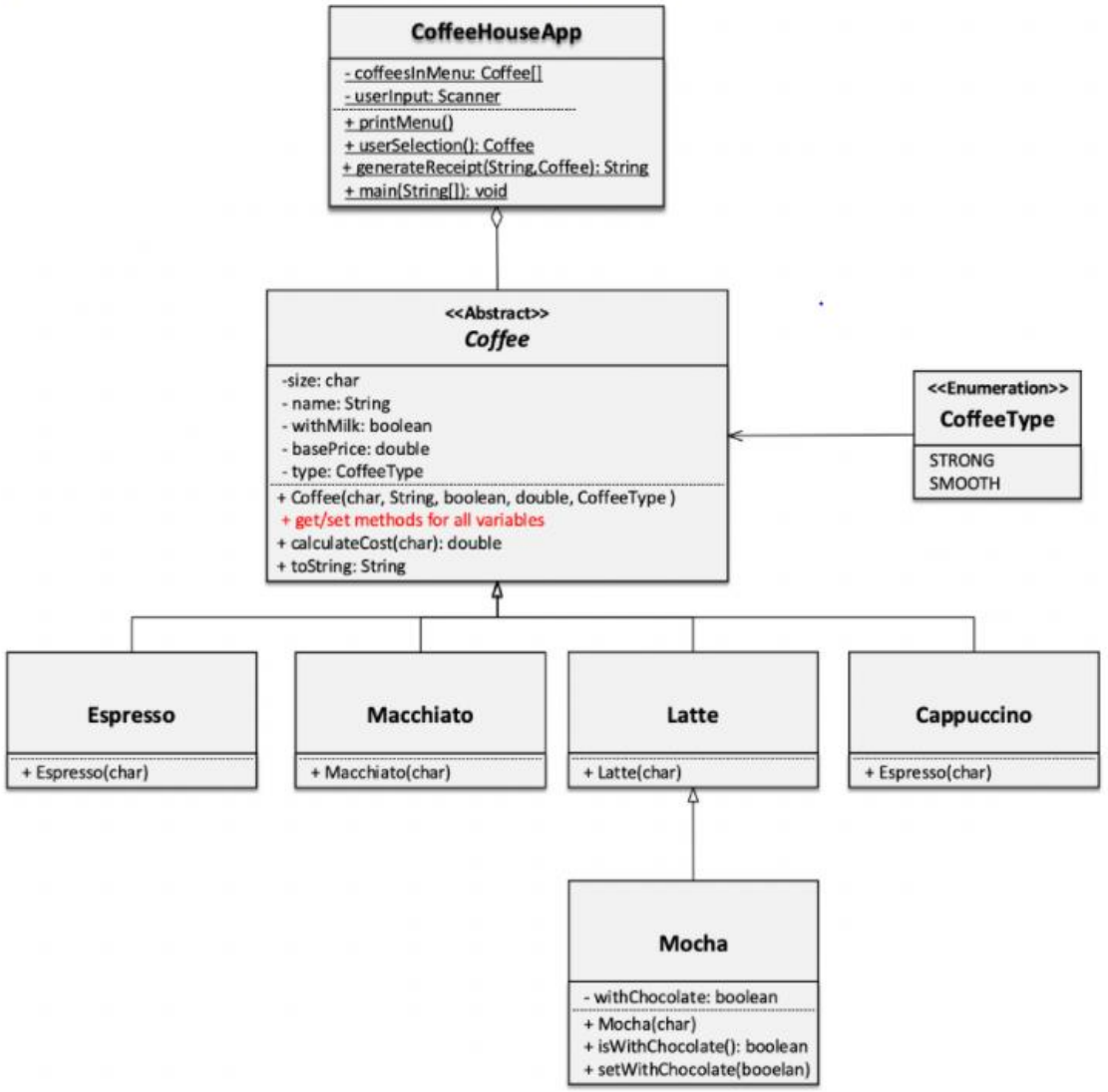
Cylinder 표면적 구하기[심화]

- Cylinder.JAVA

```
    @Override
    public String toString() {
        return "원기둥의 반지름 : " + radius +
                "₩n원기둥의 높이 : " + height +
                "₩n원기둥의 부피 : " + volume() +
                "₩n원기둥의 표면적 : " + area();
    }
}
```

CoffeeHouse

Write the appropriate codes (classes, methods, etc.) to implement the following UML diagram.



상속 예제 5

- 클래스 상속으로 코드 재사용하기

로봇(Robot) 클래스를 상속받는 청소로봇(CleanRobot) 클래스를 만들어보자
청소 로봇은 로봇이 갖는 기능을 상속받아 사용할 수 있다