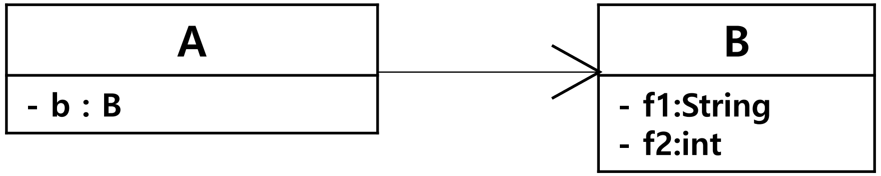
클래스 다이어그램 연습 문제 답

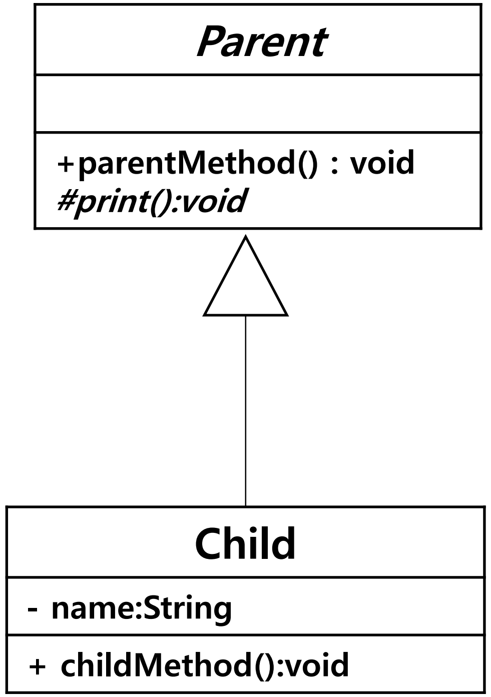
▶ 연습문제 1



```
public class A {
              private B b;
}
```

```
public class B {
              private String f1;
              private int f2;
}
```

연습문제 3



Parent

+parentMethod() : void
#print():void

Child

- name:String

+ childMethod():void

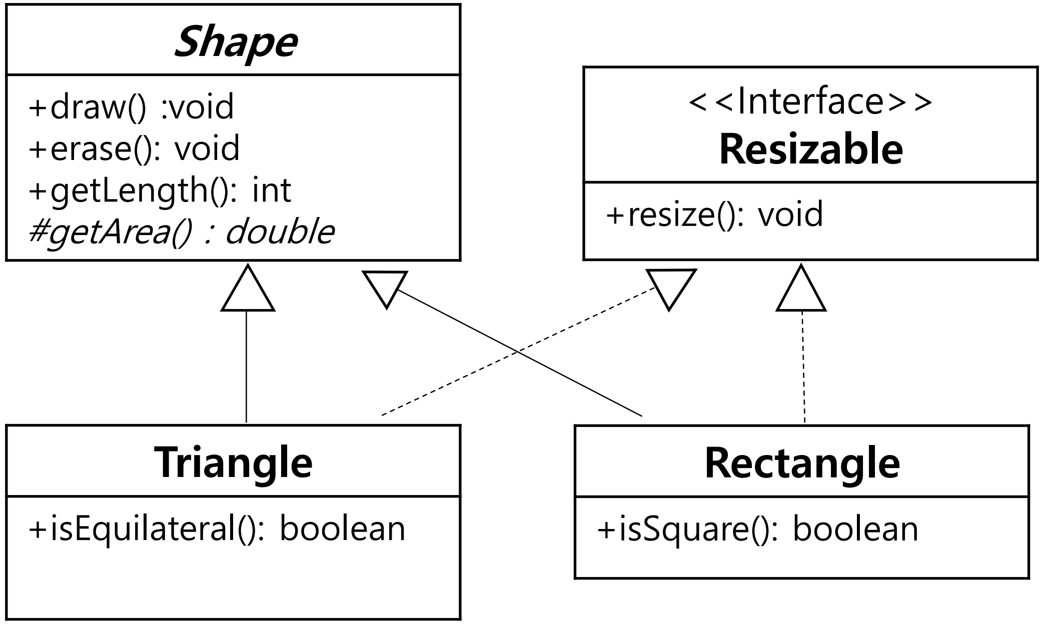
```
public abstract class Parent{
        public void parentMethod1(){ }
        protected abstract void print();
}
```

```
public class Child extends Parent{
        private String name;

        public void childMethod() {…}

        protectd void print(){…}
}
```

▶ 연습문제 4



Shape

+draw() :void
+erase(): void
+getLength(): int
#getArea() : double

<<Interface>>
Resizable

+resize(): void

Triangle

+isEquilateral(): boolean

Rectangle

+isSquare(): boolean

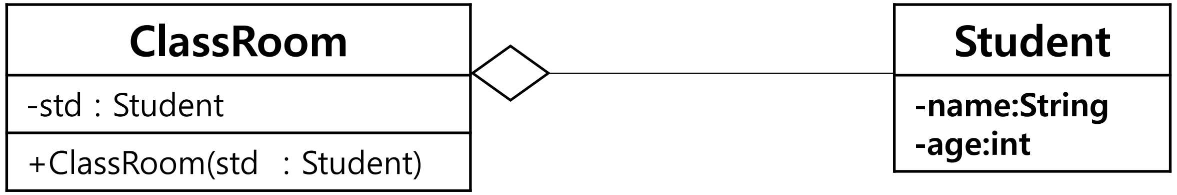
```
public abstract class Shape{
        public void draw(){…}
        public void erase(){…}
        public int getLength(){…}
        protected abstract double getArea();
}
```

```
public interface Resizable {
        void resize();
}
```

```
public class Triangle extends Shape
                        implements Resizable  {
        public boolean isEquilateral() {…}
        protected double getArea() {…}
        public void resize() {…}
}
```

```
public class Rectangle extends Shape
                        implements Resizable {
        public boolean isSquare() {…}
        protected double getArea() {…}
        public void resize() {…}
}
```

▶ 연습문제 5



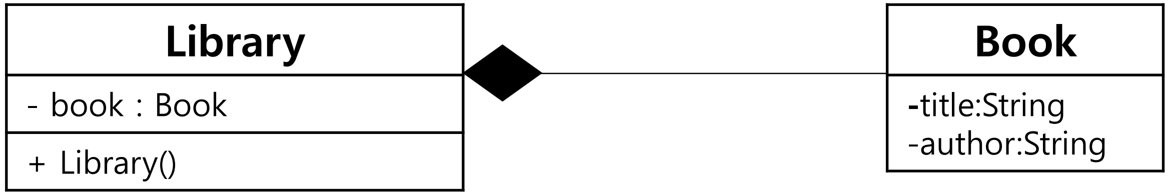
```
public class ClassRoom{
        private Student std;

        public ClassRoom(Student std){
                this.std  = std;
        }
}
```

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

}
```

▶ 연습문제 5



```
public class Library{
        private Book book;

        public Library(){
                this.book  = new book();
        }
}
```

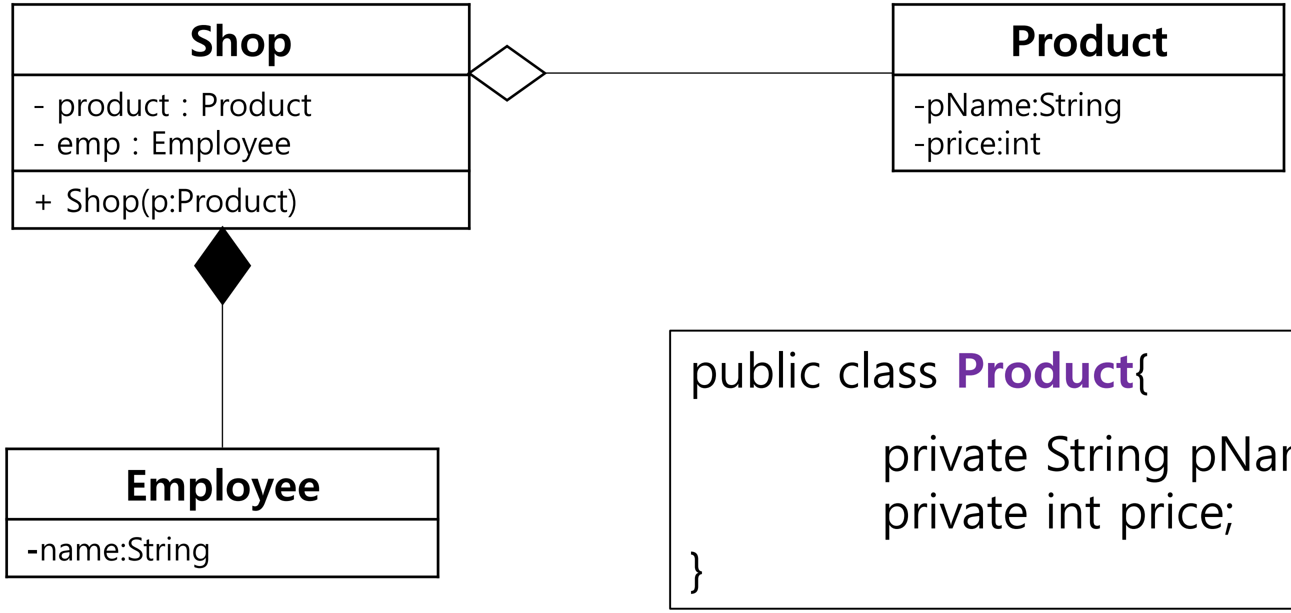
```
public class Book{
        private String title;
        private String author;
}
```

▶ 연습문제 6

```
public class Shop{

        private Product product;
        private Employee emp;

        public Shop(Product p){
                this.product = p;
                this.emp = new Employee();
        }
}
```



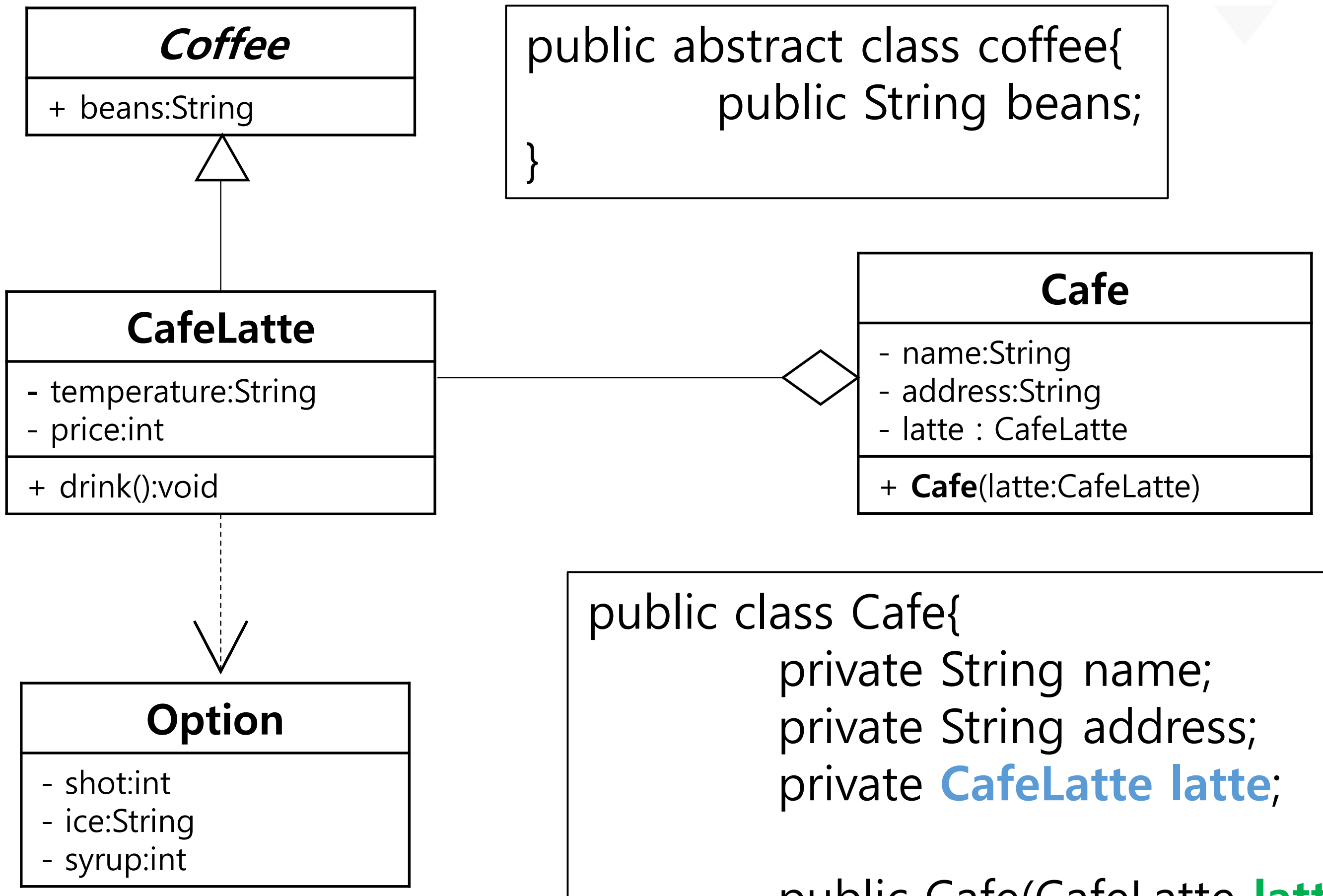
```
public class Product{

        private String pName;
        private int price;
}
```

```
public class Employee{

        private String name;
}
```

▶ 연습문제 7



```
public abstract class coffee{
            public String beans;
}
```

```
public class CafeLatte extends coffee{
            private temperature;
            private int price;

            public void drink(){
                        Option opt = new Option();
            }
}
```

```
public class Option{
            private int shot;
            private String ice;
            private int syrup;
}
```

```
public class Cafe{
            private String name;
            private String address;
            private CafeLatte latte;

            public Cafe(CafeLatte latte){
                        this.latte = latte;
            }
}
```