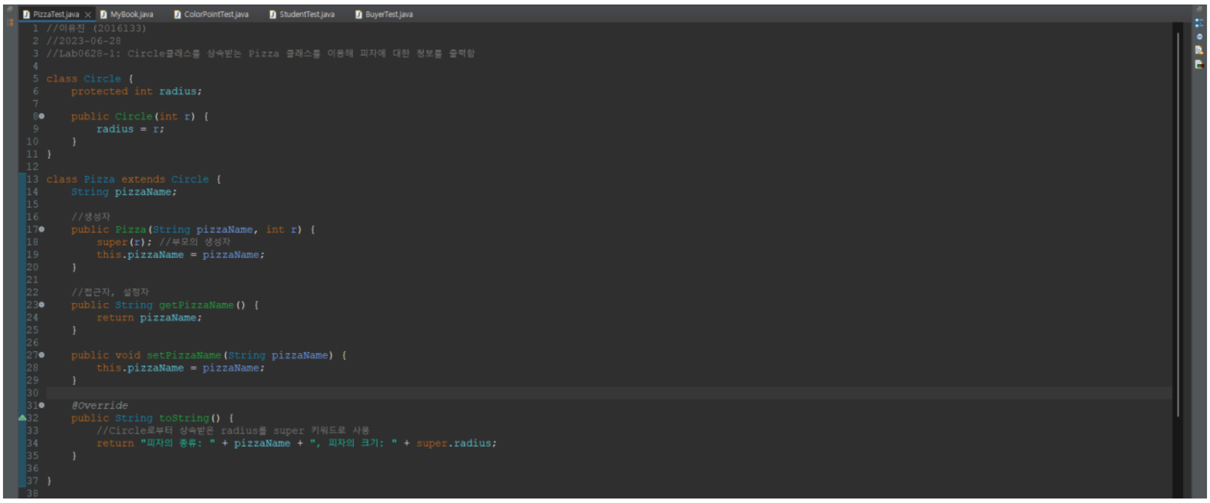
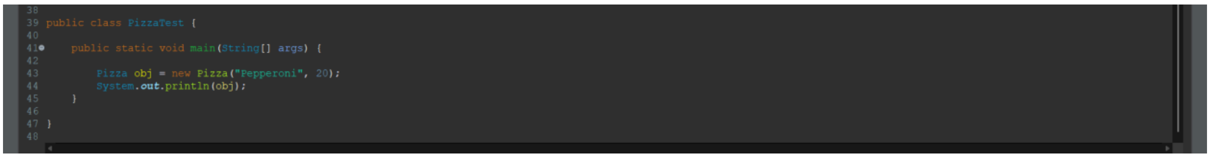
23-여름학기 자바프로그래밍(001) Lab0628
소프트웨어학부 컴퓨터과학전공 2016133 이유진

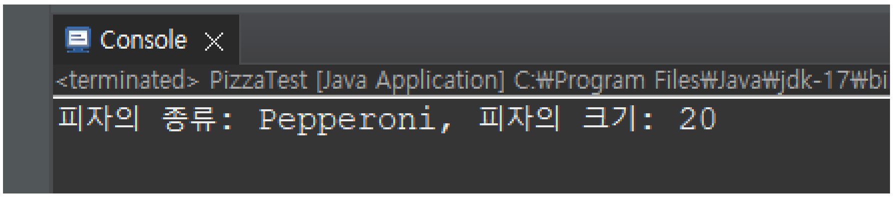
1. 실습 1 PizzaTest.java



```java
//이유진 (2016133)
//2023-06-28
//Lab0628-1: Circle클래스를 상속받는 Pizza 클래스를 이용해 피자에 대한 정보를 출력함

class Circle {
    protected int radius;

    public Circle(int r) {
        radius = r;
    }
}

class Pizza extends Circle {
    String pizzaName;

    //생성자
    public Pizza(String pizzaName, int r) {
        super(r); //부모의 생성자
        this.pizzaName = pizzaName;
    }

    //접근자, 설정자
    public String getPizzaName() {
        return pizzaName;
    }

    public void setPizzaName(String pizzaName) {
        this.pizzaName = pizzaName;
    }

    @Override
    public String toString() {
        //Circle로부터 상속받은 radius를 super 키워드로 사용
        return "피자의 종류: " + pizzaName + ", 피자의 크기: " + super.radius;
    }

}
```

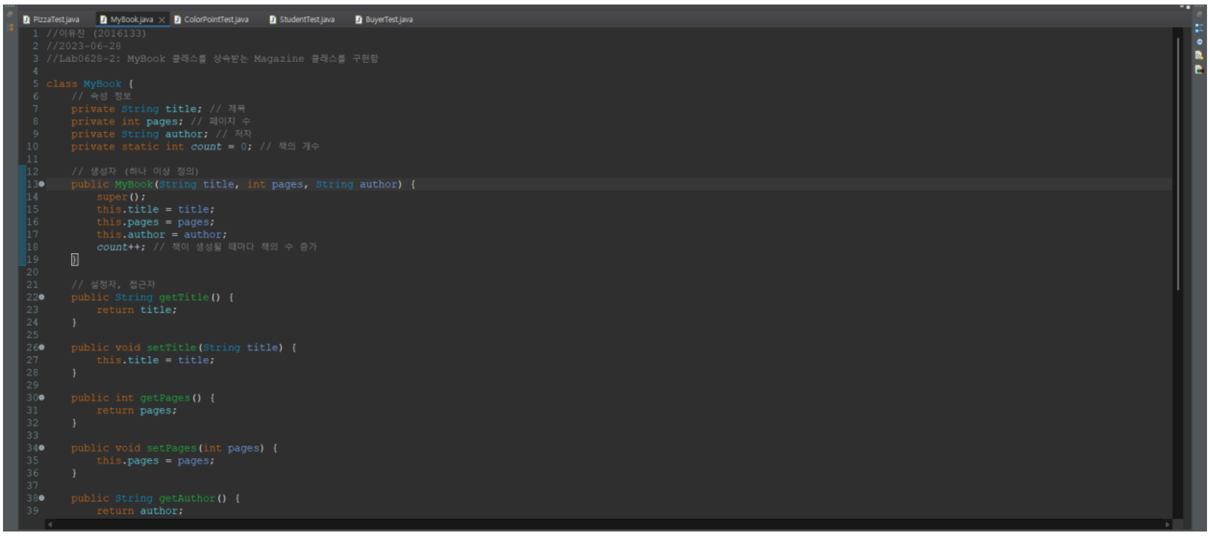


```java

public class PizzaTest {

    public static void main(String[] args) {

        Pizza obj = new Pizza("Pepperoni", 20);
        System.out.println(obj);
    }

}
```

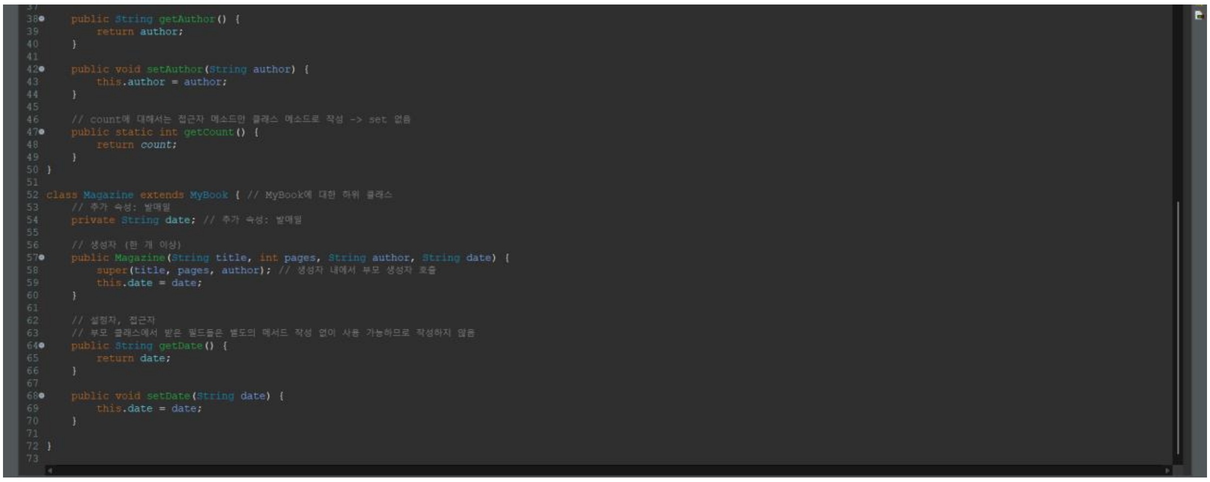


```
<terminated> PizzaTest [Java Application] C:₩Program Files₩Java₩jdk-17₩bi
피자의 종류: Pepperoni, 피자의 크기: 20
```

2. 실습 2 Mybook.java

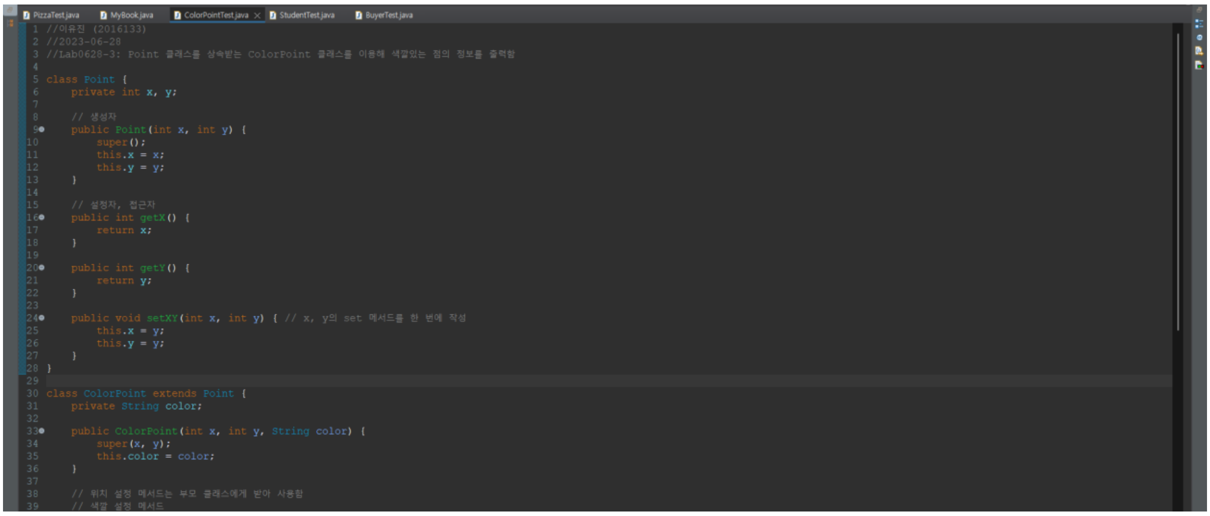


```java
//이유진 (2016133)
//2023-06-28
//Lab0628-2: MyBook 클래스를 상속받는 Magazine 클래스를 구현함

class MyBook {
    // 속성 정보
    private String title; // 제목
    private int pages; // 페이지 수
    private String author; // 저자
    private static int count = 0; // 책의 개수

    // 생성자 (하나 이상 정의)
    public MyBook(String title, int pages, String author) {
        super();
        this.title = title;
        this.pages = pages;
        this.author = author;
        count++; // 책이 생성될 때마다 책의 수 증가
    }

    // 설정자, 접근자
    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public int getPages() {
        return pages;
    }

    public void setPages(int pages) {
        this.pages = pages;
    }

    public String getAuthor() {
        return author;
```

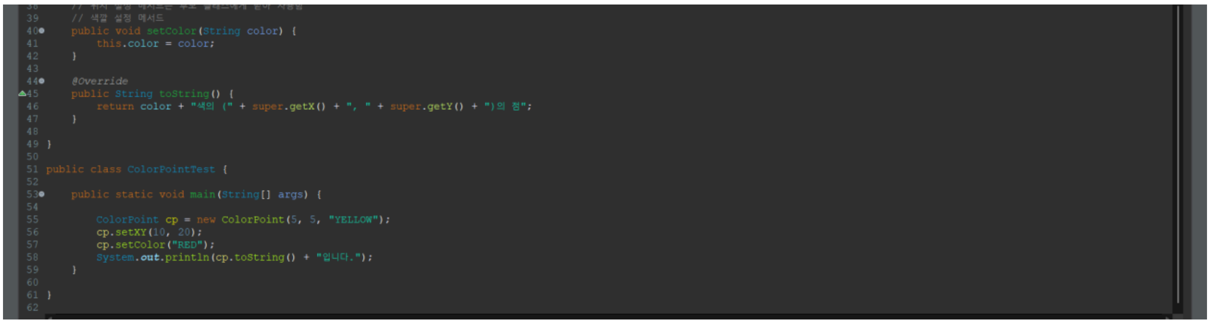


```java
    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    // count에 대해서는 접근자 메소드만 클래스 메소드로 작성 -> set 없음
    public static int getCount() {
        return count;
    }
}

class Magazine extends MyBook { // MyBook에 대한 하위 클래스
    // 추가 속성: 발매일
    private String date; // 추가 속성: 발매일

    // 생성자 (한 개 이상)
    public Magazine(String title, int pages, String author, String date) {
        super(title, pages, author); // 생성자 내에서 부모 생성자 호출
        this.date = date;
    }

    // 설정자, 접근자
    // 부모 클래스에서 받은 필드들은 별도의 메서드 작성 없이 사용 가능하므로 작성하지 않음
    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
    }

}
```

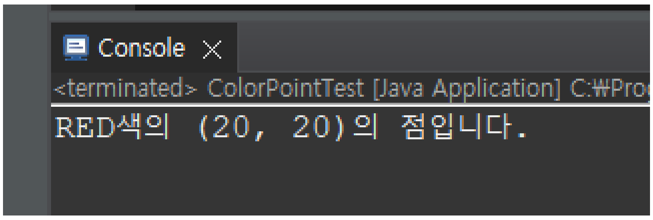
3. 실습 3 ColorPointTest.java



```java
//이유진 (2016133)
//2023-06-28
//Lab0628-3: Point 클래스를 상속받는 ColorPoint 클래스를 이용해 색깔있는 점의 정보를 출력함

class Point {
    private int x, y;

    // 생성자
    public Point(int x, int y) {
        super();
        this.x = x;
        this.y = y;
    }

    // 설정자, 접근자
    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }

    public void setXY(int x, int y) { // x, y의 set 메서드를 한 번에 작성
        this.x = y;
        this.y = y;
    }
}

class ColorPoint extends Point {
    private String color;

    public ColorPoint(int x, int y, String color) {
        super(x, y);
        this.color = color;
    }

    // 위치 설정 메서드는 부모 클래스에게 받아 사용함
    // 색깔 설정 메서드
```

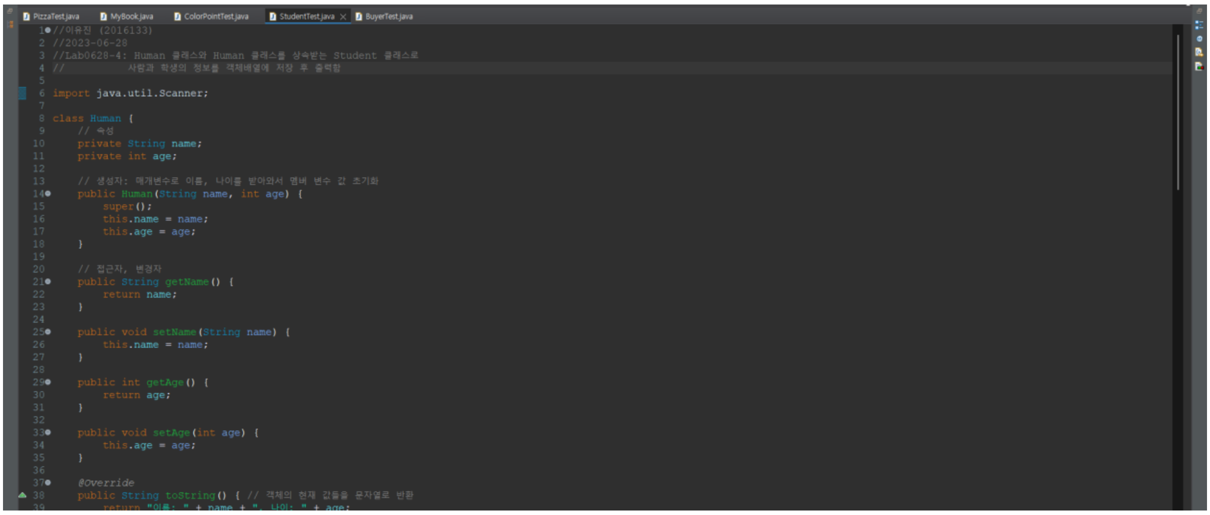


```java
    public void setColor(String color) {
        this.color = color;
    }

    @Override
    public String toString() {
        return color + "색의 (" + super.getX() + ", " + super.getY() + ")의 점";
    }

}

public class ColorPointTest {

    public static void main(String[] args) {

        ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
        cp.setXY(10, 20);
        cp.setColor("RED");
        System.out.println(cp.toString() + "입니다.");
    }

}
```

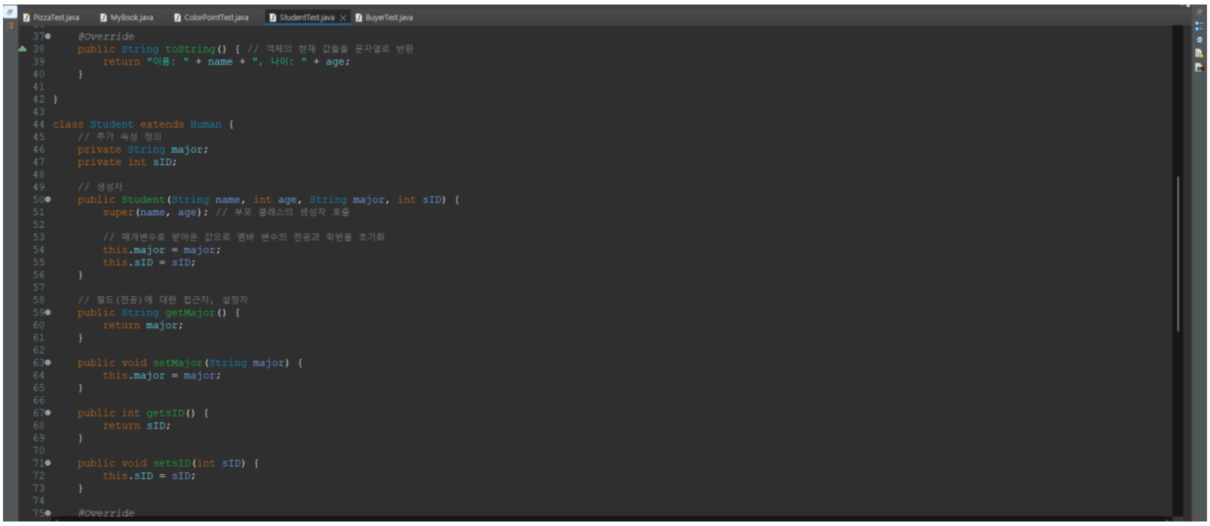


```
<terminated> ColorPointTest [Java Application] C:₩Pro
RED색의 (20, 20)의 점입니다.
```

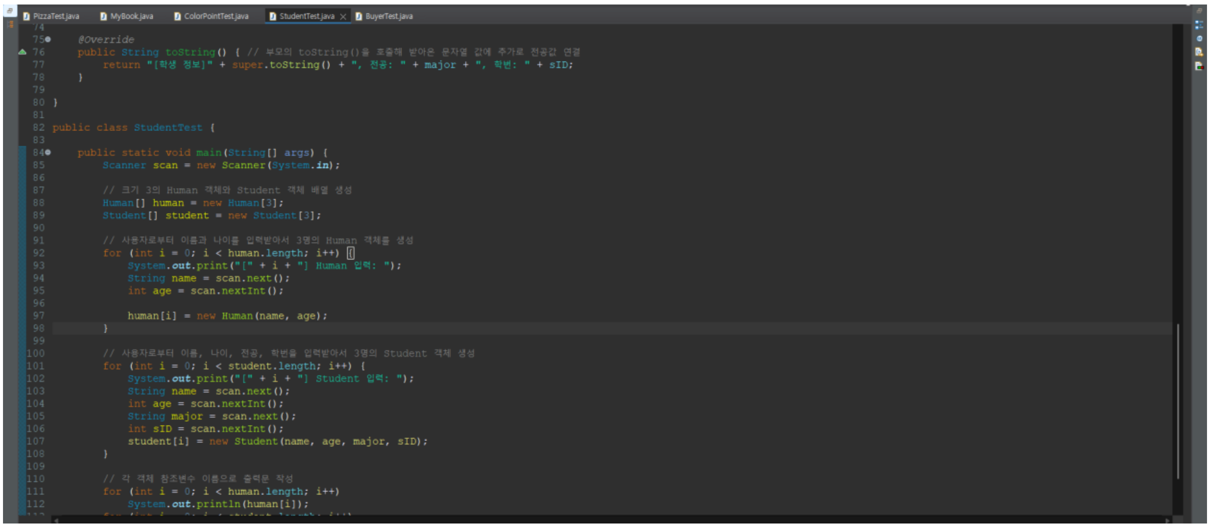
4. 실습 4 StudentTest.java



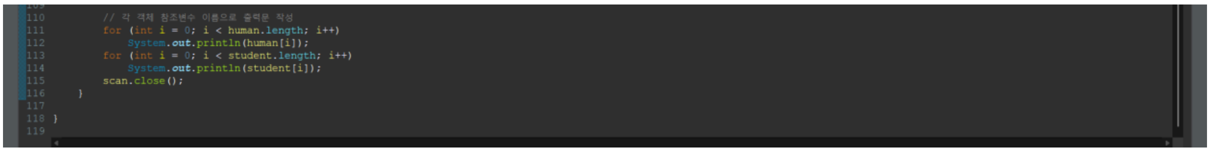
```java
//이유진 (2016133)
//2023-06-28
//Lab0628-4: Human 클래스와 Human 클래스를 상속받는 Student 클래스로
//           사람과 학생의 정보를 객체배열에 저장 후 출력함

import java.util.Scanner;

class Human {
    // 속성
    private String name;
    private int age;

    // 생성자: 매개변수로 이름, 나이를 받아와서 멤버 변수 값 초기화
    public Human(String name, int age) {
        super();
        this.name = name;
        this.age = age;
    }

    // 접근자, 변경자
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() { // 객체의 현재 값들을 문자열로 반환
        return "이름: " + name + ", 나이: " + age;
```



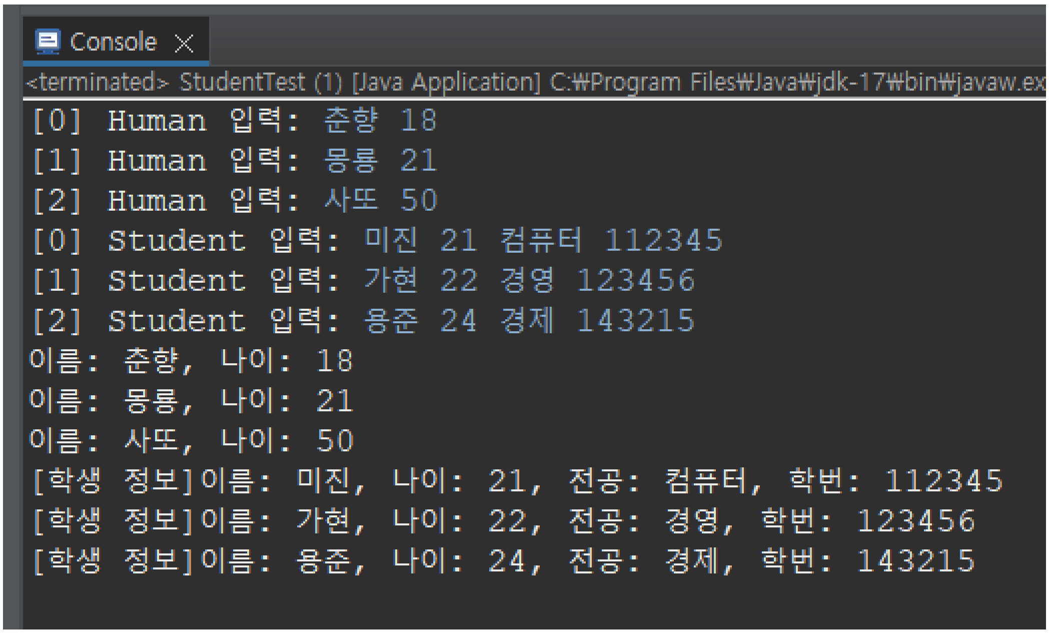
```java
    @Override
    public String toString() { // 객체의 현재 값들을 문자열로 반환
        return "이름: " + name + ", 나이: " + age;
    }

}

class Student extends Human {
    // 추가 속성 정의
    private String major;
    private int sID;

    // 생성자
    public Student(String name, int age, String major, int sID) {
        super(name, age); // 부모 클래스의 생성자 호출

        // 매개변수로 받아온 값으로 멤버 변수의 전공과 학번을 조기화
        this.major = major;
        this.sID = sID;
    }

    // 필드(전공)에 대한 접근자, 설정자
    public String getMajor() {
        return major;
    }

    public void setMajor(String major) {
        this.major = major;
    }

    public int getsID() {
        return sID;
    }

    public void setsID(int sID) {
        this.sID = sID;
    }

    @Override
```



```java
    @Override
    public String toString() { // 부모의 toString()을 호출해 받아온 문자열 값에 추가로 전공값 연결
        return "[학생 정보]" + super.toString() + ", 전공: " + major + ", 학번: " + sID;
    }

}

public class StudentTest {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        // 크기 3의 Human 객체와 Student 객체 배열 생성
        Human[] human = new Human[3];
        Student[] student = new Student[3];

        // 사용자로부터 이름과 나이를 입력받아서 3명의 Human 객체를 생성
        for (int i = 0; i < human.length; i++) {
            System.out.print("[" + i + "] Human 입력: ");
            String name = scan.next();
            int age = scan.nextInt();

            human[i] = new Human(name, age);
        }

        // 사용자로부터 이름, 나이, 전공, 학번을 입력받아서 3명의 Student 객체 생성
        for (int i = 0; i < student.length; i++) {
            System.out.print("[" + i + "] Student 입력: ");
            String name = scan.next();
            int age = scan.nextInt();
            String major = scan.next();
            int sID = scan.nextInt();
            student[i] = new Student(name, age, major, sID);
        }

        // 각 객체 참조변수 이름으로 출력문 작성
        for (int i = 0; i < human.length; i++)
            System.out.println(human[i]);
```

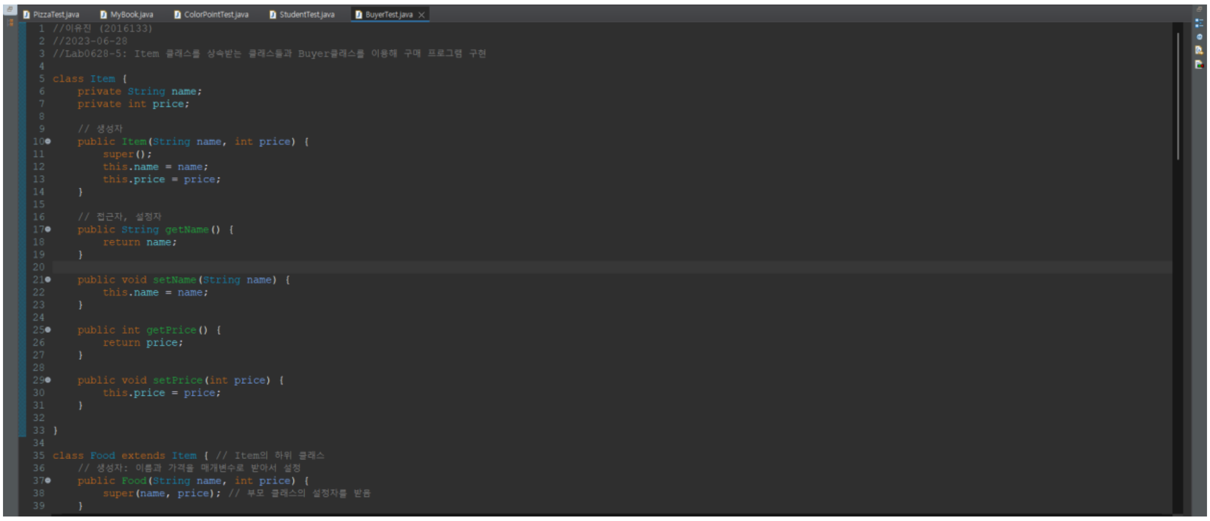


```java
        // 각 객체 참조변수 이름으로 출력문 작성
        for (int i = 0; i < human.length; i++)
            System.out.println(human[i]);
        for (int i = 0; i < student.length; i++)
            System.out.println(student[i]);
        scan.close();
    }

}
```

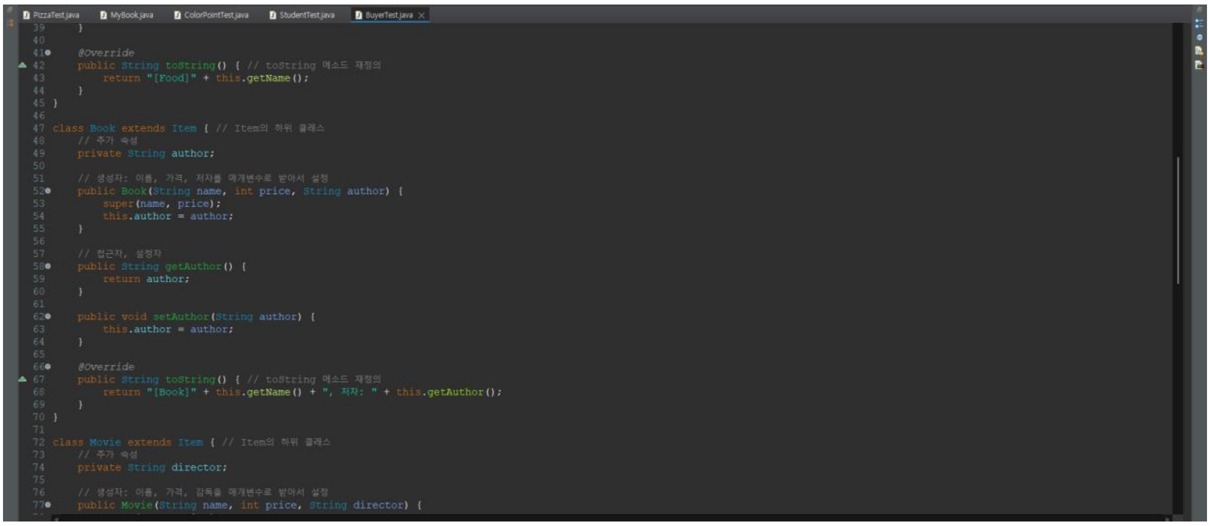


```
<terminated> StudentTest (1) [Java Application] C:₩Program Files₩Java₩jdk-17₩bin₩javaw.ex
[0] Human 입력: 춘향 18
[1] Human 입력: 몽룡 21
[2] Human 입력: 사또 50
[0] Student 입력: 미진 21 컴퓨터 112345
[1] Student 입력: 가현 22 경영 123456
[2] Student 입력: 용준 24 경제 143215
이름: 춘향, 나이: 18
이름: 몽룡, 나이: 21
이름: 사또, 나이: 50
[학생 정보]이름: 미진, 나이: 21, 전공: 컴퓨터, 학번: 112345
[학생 정보]이름: 가현, 나이: 22, 전공: 경영, 학번: 123456
[학생 정보]이름: 용준, 나이: 24, 전공: 경제, 학번: 143215
```

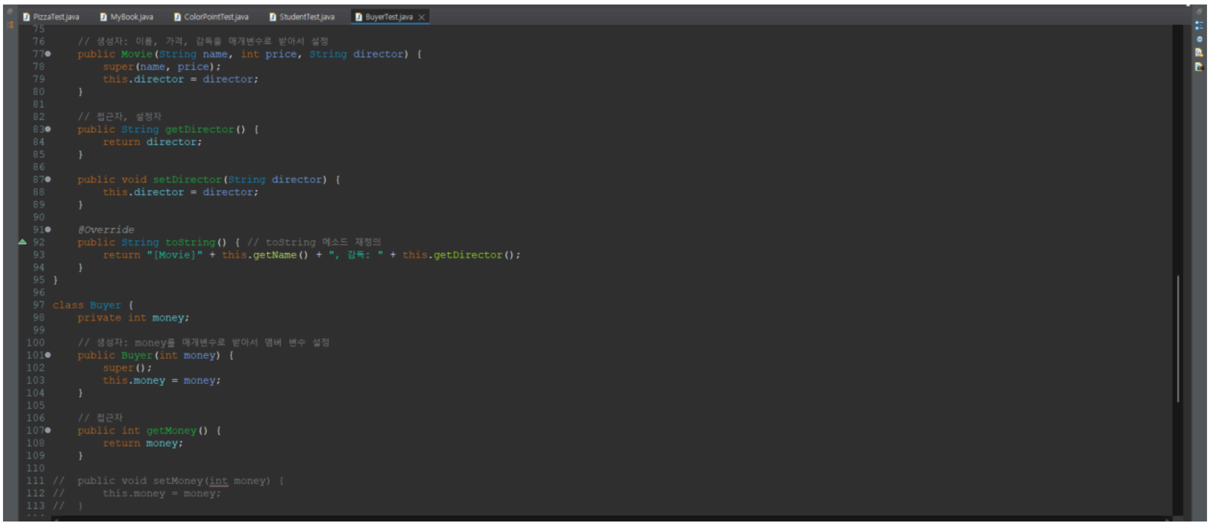
5. 실습 5 BuyerTest.java



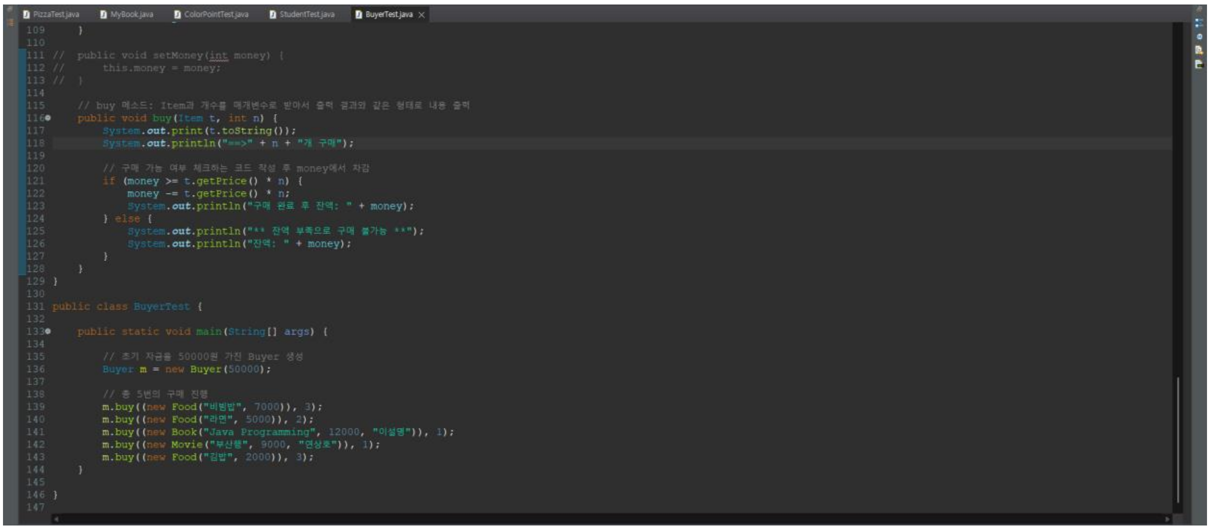
```java
//이유진 (2016133)
//2023-06-28
//Lab0628-5: Item 클래스를 상속받는 클래스들과 Buyer클래스를 이용해 구매 프로그램 구현

class Item {
    private String name;
    private int price;

    // 생성자
    public Item(String name, int price) {
        super();
        this.name = name;
        this.price = price;
    }

    // 접근자, 설정자
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getPrice() {
        return price;
    }

    public void setPrice(int price) {
        this.price = price;
    }

}

class Food extends Item { // Item의 하위 클래스
    // 생성자: 이름과 가격을 매개변수로 받아서 설정
    public Food(String name, int price) {
        super(name, price); // 부모 클래스의 설정자를 받음
    }
```



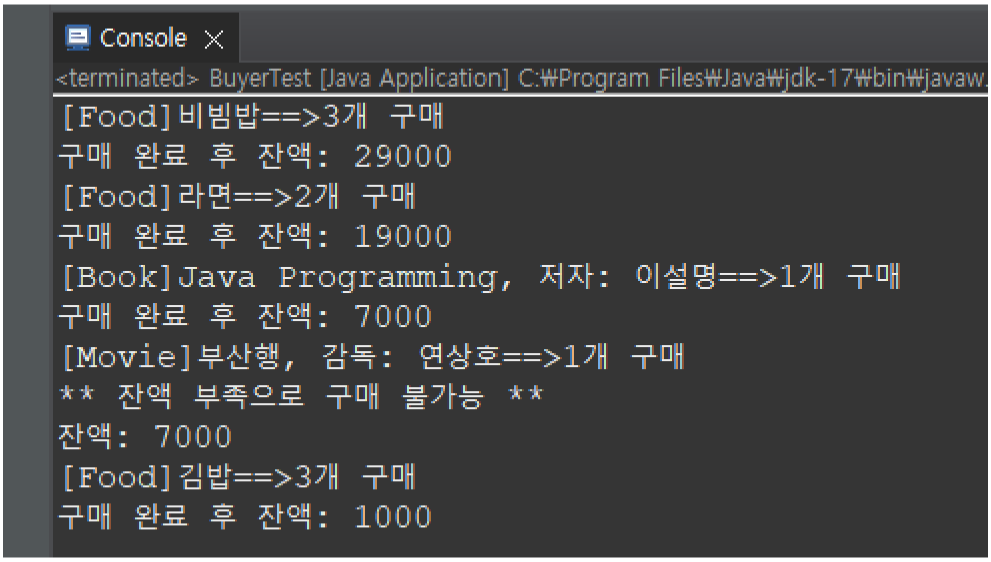
```java
    }

    @Override
    public String toString() { // toString 메소드 재정의
        return "[Food]" + this.getName();
    }
}

class Book extends Item { // Item의 하위 클래스
    // 추가 속성
    private String author;

    // 생성자: 이름, 가격, 저자를 매개변수로 받아서 설정
    public Book(String name, int price, String author) {
        super(name, price);
        this.author = author;
    }

    // 접근자, 설정자
    public String getAuthor() {
        return author;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    @Override
    public String toString() { // toString 메소드 재정의
        return "[Book]" + this.getName() + ", 저자: " + this.getAuthor();
    }
}

class Movie extends Item { // Item의 하위 클래스
    // 추가 속성
    private String director;

    // 생성자: 이름, 가격, 감독을 매개변수로 받아서 설정
    public Movie(String name, int price, String director) {
```



```java

    // 생성자: 이름, 가격, 감독을 매개변수로 받아서 설정
    public Movie(String name, int price, String director) {
        super(name, price);
        this.director = director;
    }

    // 접근자, 설정자
    public String getDirector() {
        return director;
    }

    public void setDirector(String director) {
        this.director = director;
    }

    @Override
    public String toString() { // toString 메소드 재정의
        return "[Movie]" + this.getName() + ", 감독: " + this.getDirector();
    }
}

class Buyer {
    private int money;

    // 생성자: money를 매개변수로 받아서 멤버 변수 설정
    public Buyer(int money) {
        super();
        this.money = money;
    }

    // 접근자
    public int getMoney() {
        return money;
    }

//  public void setMoney(int money) {
//      this.money = money;
//  }
```



```java
    }

//  public void setMoney(int money) {
//      this.money = money;
//  }

    // buy 메소드: Item과 개수를 매개변수로 받아서 출력 결과와 같은 형태로 내용 출력
    public void buy(Item t, int n) {
        System.out.print(t.toString());
        System.out.println("==>" + n + "개 구매");

        // 구매 가능 여부 체크하는 코드 작성 후 money에서 차감
        if (money >= t.getPrice() * n) {
            money -= t.getPrice() * n;
            System.out.println("구매 완료 후 잔액: " + money);
        } else {
            System.out.println("** 잔액 부족으로 구매 불가능 **");
            System.out.println("잔액: " + money);
        }
    }
}

public class BuyerTest {

    public static void main(String[] args) {

        // 초기 자금을 50000원 가진 Buyer 생성
        Buyer m = new Buyer(50000);

        // 총 5번의 구매 진행
        m.buy((new Food("비빔밥", 7000)), 3);
        m.buy((new Food("라면", 5000)), 2);
        m.buy((new Book("Java Programming", 12000, "이설명")), 1);
        m.buy((new Movie("부산행", 9000, "연상호")), 1);
        m.buy((new Food("김밥", 2000)), 3);
    }

}
```



```
<terminated> BuyerTest [Java Application] C:₩Program Files₩Java₩jdk-17₩bin₩javaw.
[Food]비빔밥==>3개 구매
구매 완료 후 잔액: 29000
[Food]라면==>2개 구매
구매 완료 후 잔액: 19000
[Book]Java Programming, 저자: 이설명==>1개 구매
구매 완료 후 잔액: 7000
[Movie]부산행, 감독: 연상호==>1개 구매
** 잔액 부족으로 구매 불가능 **
잔액: 7000
[Food]김밥==>3개 구매
구매 완료 후 잔액: 1000
```