

# 자바 입문 과제\_2

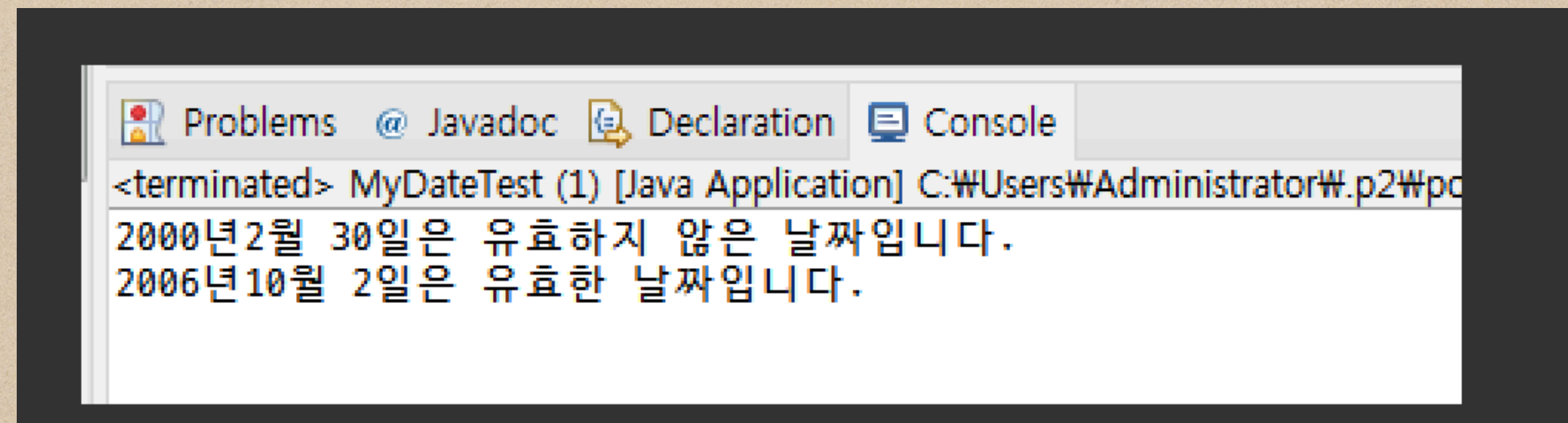
박은종



# 다음 코드가 수행되도록 MyDate 클래스를 구현하세요

```
public class MyDateTest {  
  
    public static void main(String[] args) {  
        MyDate date1 = new MyDate(30, 2, 2000);  
        System.out.println(date1.isValid());  
        MyDate date2 = new MyDate(2, 10, 2006);  
        System.out.println(date2.isValid());  
  
        // date1.day =10; 에러  
        // date1.month = 3;  
        // date1.year = 2020;  
  
    }  
}
```

#private #information hiding #윤년 계산하기  
#encapsulation #접근제어자 #getter/setter





# this

- ◆ 자바 프로그래밍에서 this 키워드는 크게 세 가지의 용도로 사용됩니다.
  - ◆ 자기 자신의 인스턴스 주소를 가리키는 this
  - ◆ 생성자에서 다른 생성자를 호출하는 this
  - ◆ 자기 자신을 반환하는 this
- ◆ 위 세가지 경우에 대한 예제 코드를 작성하세요

#this #생성자에서 this 사용할 때 주의 할 점 #this출력해보기



# 참조 자료형 변수

- ◆ 자바에서 변수는 자료형을 반드시 가져야 합니다.
- ◆ 자료형은 크게 다음과 같이 두 종류로 구분 할 수 있습니다.





# 참조 자료형 변수

- ◆ 다음과 같은 Student 클래스가 있을 때 학생이 수강하는 과목에 대한 정보 (과목 이름, 과목 점수)를 따로 클래스로 만들어서 Test 코드가 수행 되도록 하세요

```
public class Student {
```

```
    int studentID;
```

```
    String studentName;
```

```
    int koreaScore;
```

```
    int mathScore;
```

```
    String koreaSubject;
```

```
    String mathSubject;
```

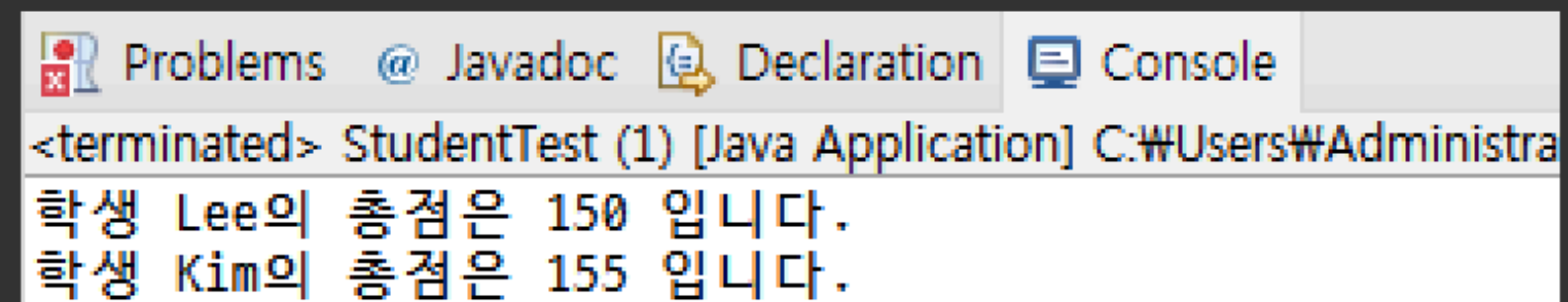
```
}
```

⇒ Subject 클래스로 분리



# 참조 자료형 변수

```
public class StudentTest {  
  
    public static void main(String[] args) {  
        Student studentLee = new Student(1001, "Lee");  
  
        studentLee.setKorean("국어", 100);  
        studentLee.setMath("수학", 50);  
  
        Student studentKim = new Student(1002, "Kim");  
  
        studentKim.setKorean("국어", 70);  
        studentKim.setMath("수학", 85);  
  
        studentLee.showStudentInfo();  
        studentKim.showStudentInfo();  
    }  
}
```



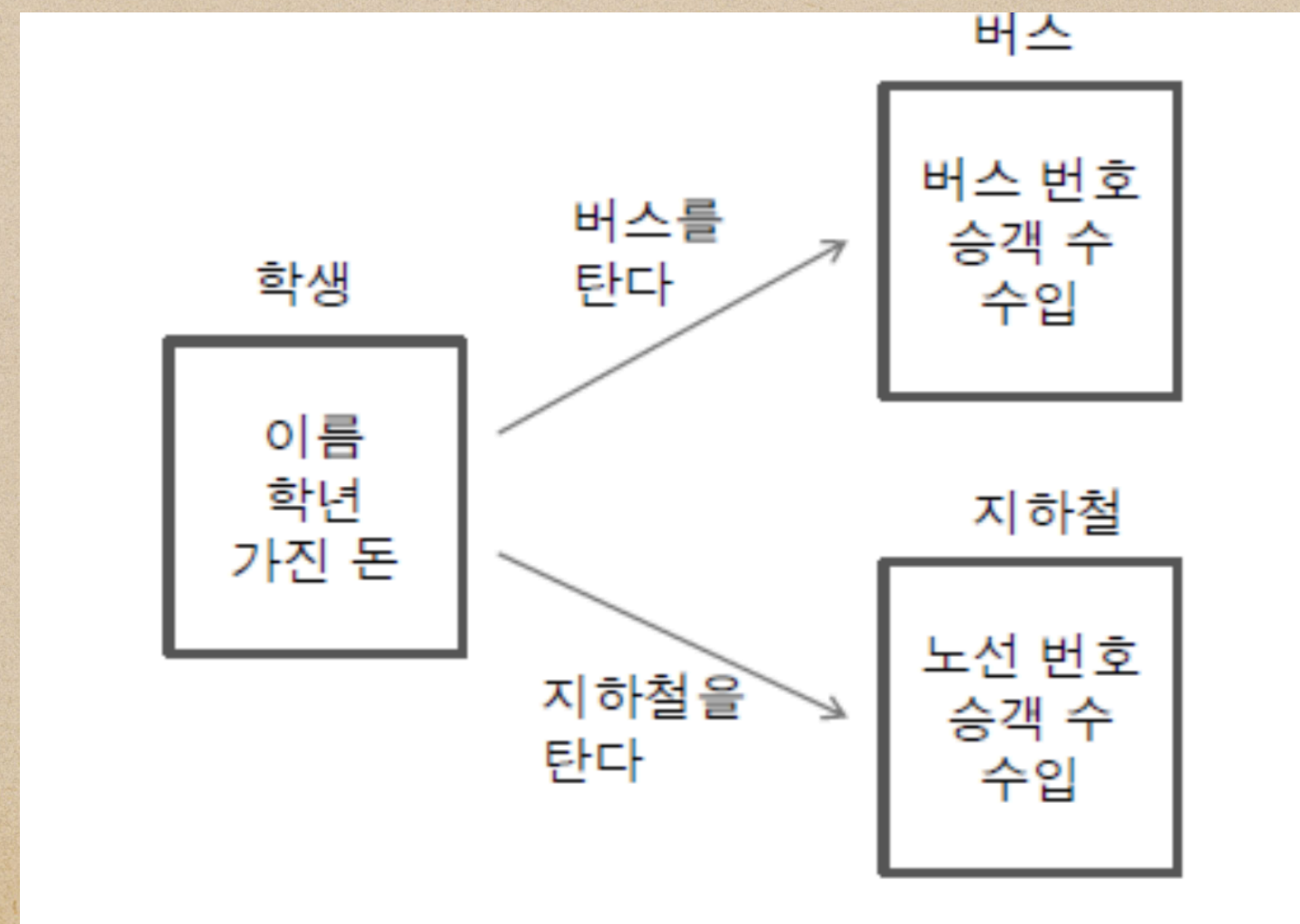
The screenshot shows an IDE console window with the following tabs: Problems, Javadoc, Declaration, and Console. The console output is as follows:

```
<terminated> StudentTest (1) [Java Application] C:\Users\Administra  
학생 Lee의 총점은 150 입니다.  
학생 Kim의 총점은 155 입니다.
```



# 객체 협력 (cooperation)

- ◆ 객체 지향 프로그래밍에서는 객체 간의 협력이 이루어집니다.
- ◆ 협력을 위해서는 필요한 메시지를 전송하고 이를 처리하는 기능을 구현합니다.
- ◆ 매개 변수로 객체가 전달합니다.
- ◆ 객체 협력의 예



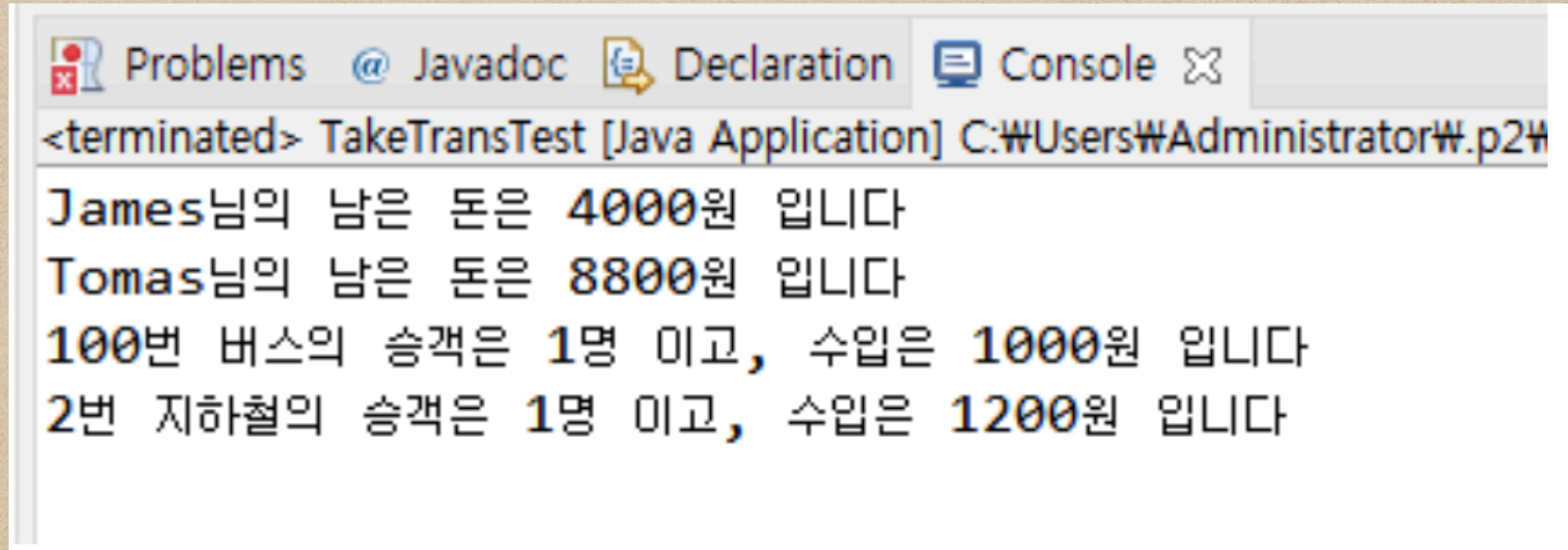


# 다음 기술된 사항을 구현하여 테스트 코드가 실행되도록 하세요

James와 Tomas는 각각 버스와 지하철을 타고 학교에 갑니다.  
James는 5000원을 가지고 있었고, 100번 버스를 타면서 1000원을 지불합니다.  
Tomas는 10000원을 가지고 있었고, 초록색 지하철을 타면서 1200원을 지불합니다.

두 학생이 버스와 지하철을 타는 상황을 구현해 봅시다.

```
public class TakeTransTest {  
  
    public static void main(String[] args) {  
        Student studentJ = new Student("James", 5000);  
        Student studentT = new Student("Tomas", 10000);  
  
        Bus bus100 = new Bus(100);  
  
        Subway subwayGreen = new Subway(2);  
  
        studentJ.takeBus(bus100);  
        studentT.takeSubway(subwayGreen);  
  
        studentJ.showInfo();  
        studentT.showInfo();  
  
        bus100.showBusInfo();  
  
        subwayGreen.showSubwayInfo();  
    }  
}
```



```
Problems @ Javadoc Declaration Console  
<terminated> TakeTransTest [Java Application] C:\Users\Administrator\p2\  
James님의 남은 돈은 4000원 입니다  
Tomas님의 남은 돈은 8800원 입니다  
100번 버스의 승객은 1명 미고, 수입은 1000원 입니다  
2번 지하철의 승객은 1명 미고, 수입은 1200원 입니다
```