객체지향프로그래밍응용 과제03 컴퓨터소프트웨어공학과 2-YA 20202296 전채린

[예제 3-16]

[소스코드]

// 예제 3-16

// 추상 클래스와 추상 메소드

abstract public class Animal {

String name;

abstract void move();

}

[예제 3-17]

[소스코드]

// 예제 3-17

// 추상 클래스와 추상 메소드

public class Tiger extends Animal {

int age;

void move() {

System.out.println("네발로 이동한다.");

}

}

[예제 3-18]

[소스코드]

// 예제 3-18

// 추상 클래스와 추상 메소드

public class Eagle extends Animal {

String home;

void move() {

System.out.println("날개로 이동한다.");

}

}

[예제 3-19]

[소스코드]

// 예제 3-19

// 추상 클래스와 추상 메소드

public class Exam10 {

public static void main(String args[]) {

Tiger tiger1 = new Tiger();

Eagle eagle1 = new Eagle();

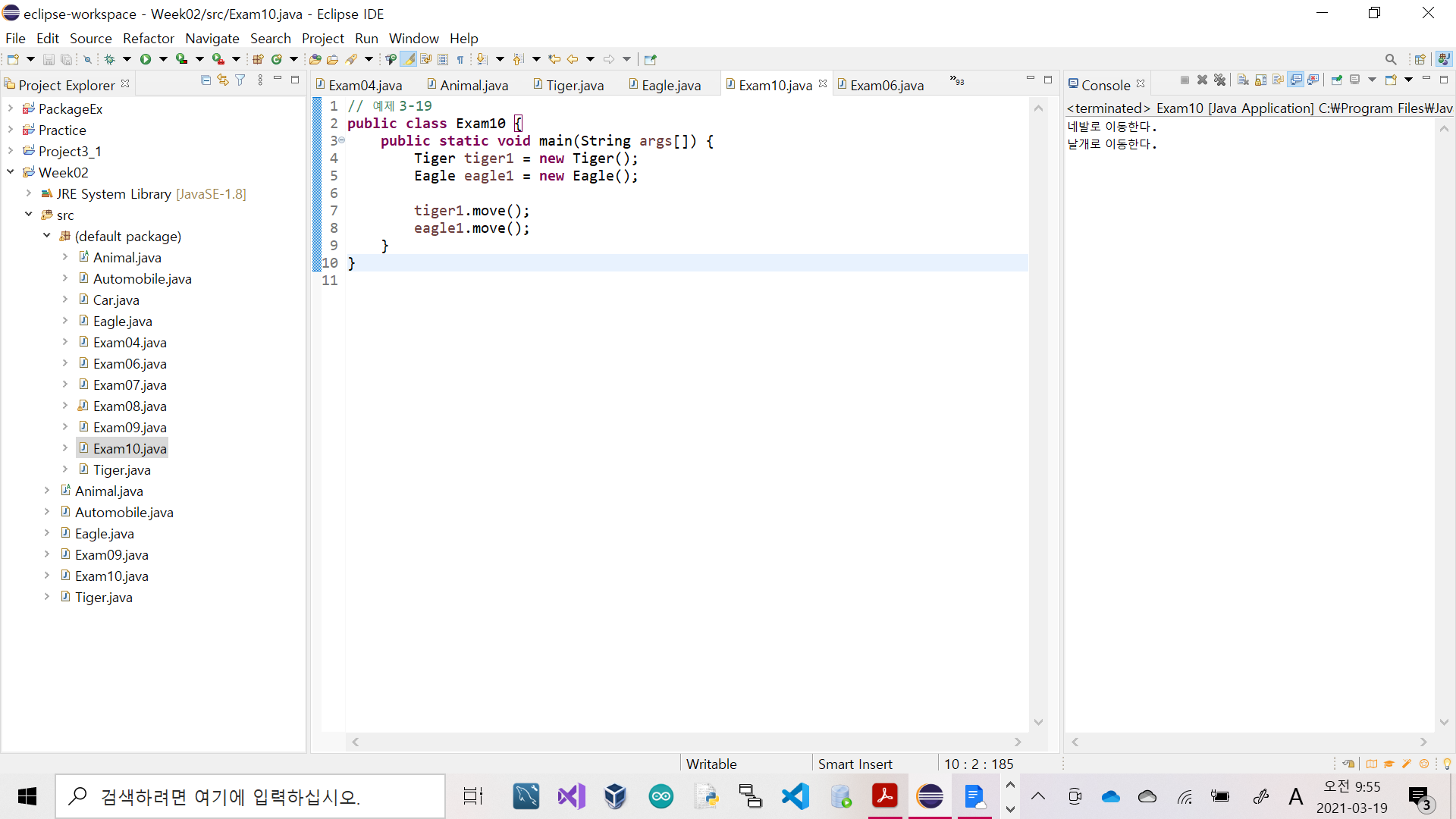
tiger1.move();

eagle1.move();

}

}

[실행결과]



[예제 3-20]

[소스코드]

// 예제 3-20

// 클래스 변수의 다형성

public class Exam11 {

public static void main(String args[]) {

Animal animal;

animal = new Tiger();

animal.move();

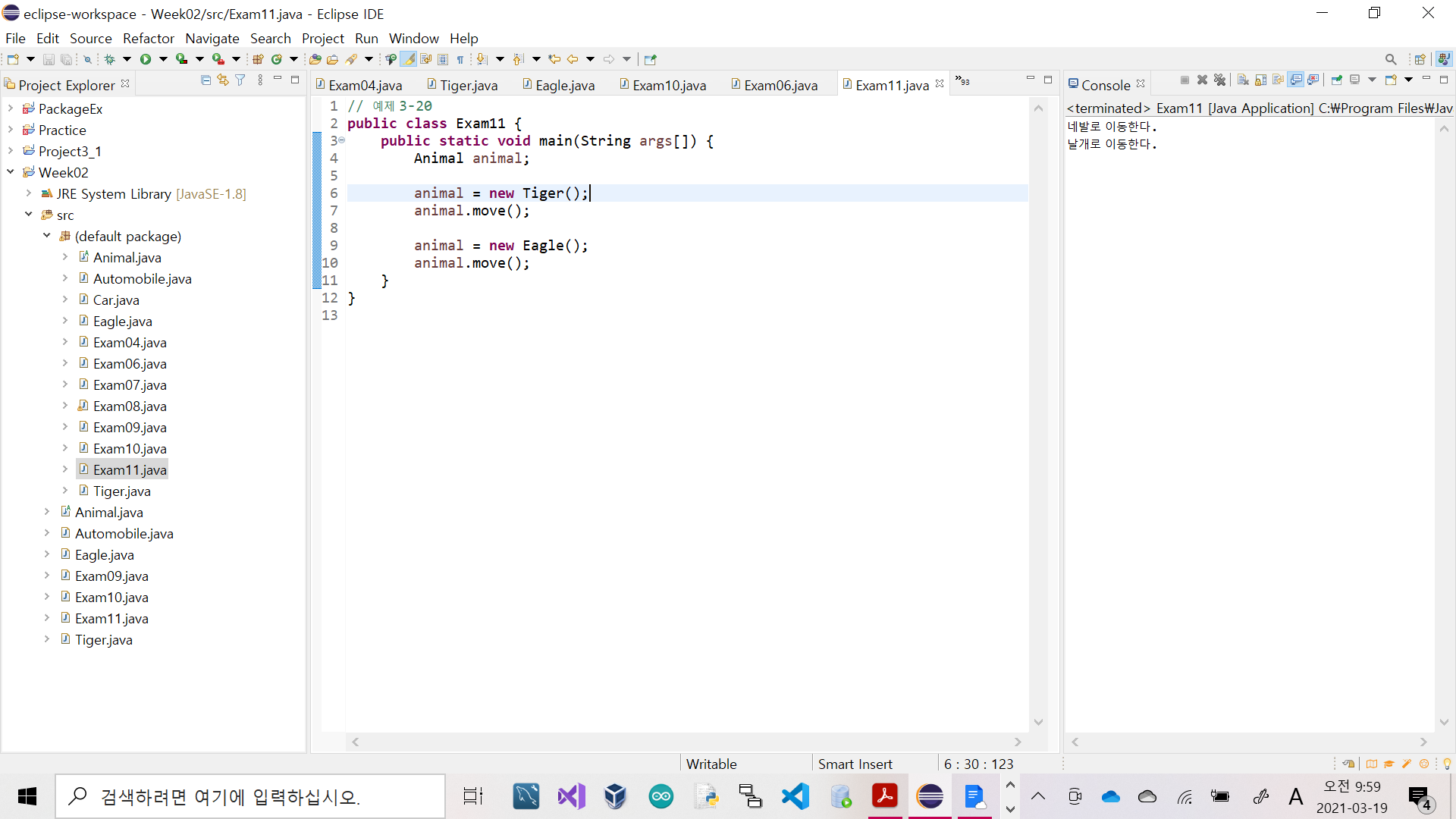
animal = new Eagle();

animal.move();

}

}

[실행결과]



[예제 3-21]

[소스코드]

// 예제 3-21

// 인터페이스와 다중 상속

interface iAnimal {

abstract void eat();

}

public class Exam12 {

public static void main(String args[]) {

iCat cat = new iCat();

cat.eat();

iTiger tiger = new iTiger();

tiger.move();

tiger.eat();

}

static class iCat implements iAnimal {

public void eat() {

System.out.println("생선을 좋아한다.");

}

}

static class iTiger extends Animal implements iAnimal {

void move() {

System.out.println("네발로 이동한다.");

}

public void eat() {

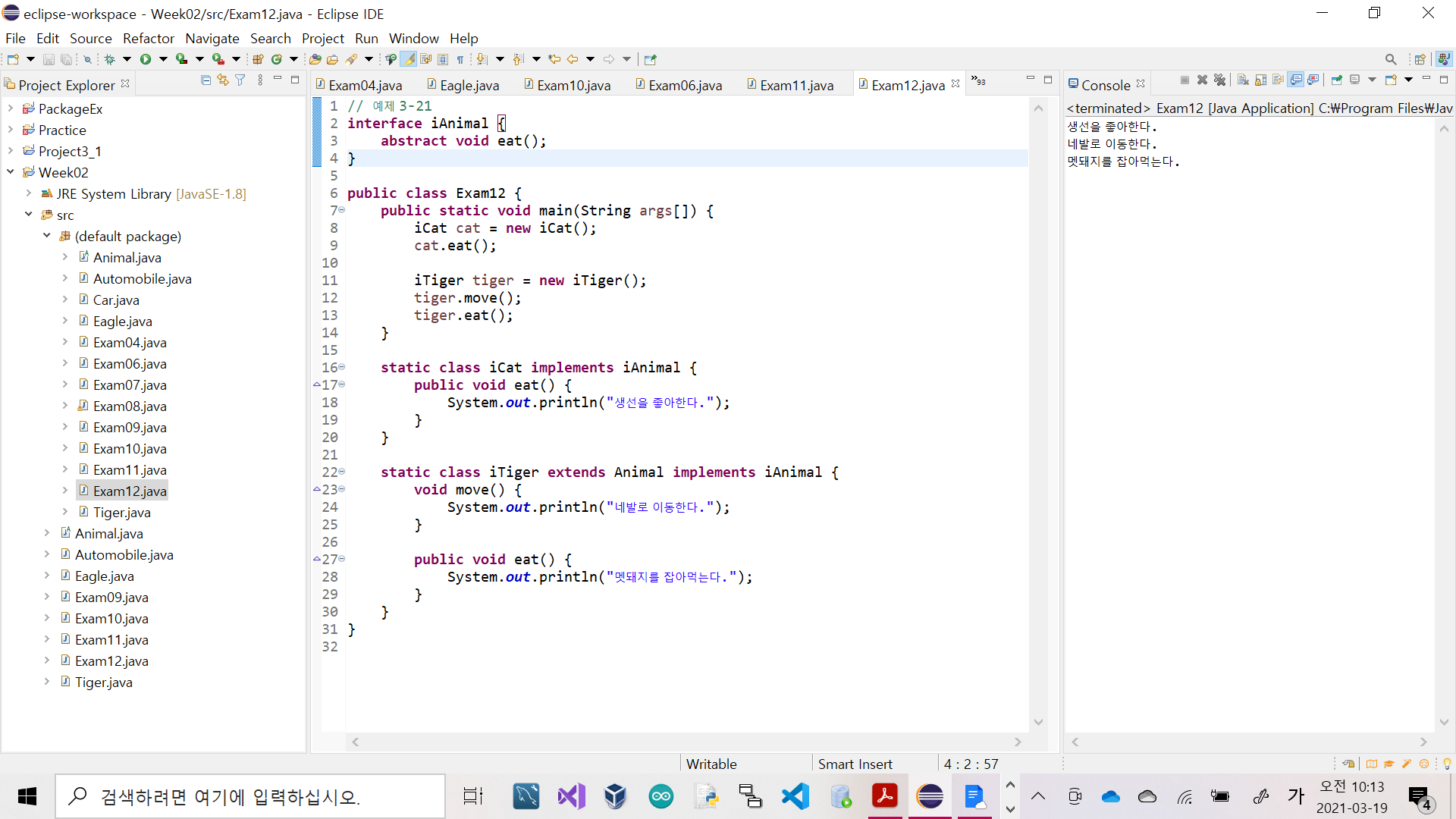
System.out.println("멧돼지를 잡아먹는다.");

}

}

}

[실행결과]



[예제 3-22]

[소스코드]

// 예제 3-22

// 익명 내부 클래스

interface clickListener {

public void print();

}

public class Exam13 {

public static void main(String args[]) {

clickListener listener =

(new clickListener() {

public void print() {

System.out.println("클릭 리스너입니다.");

}

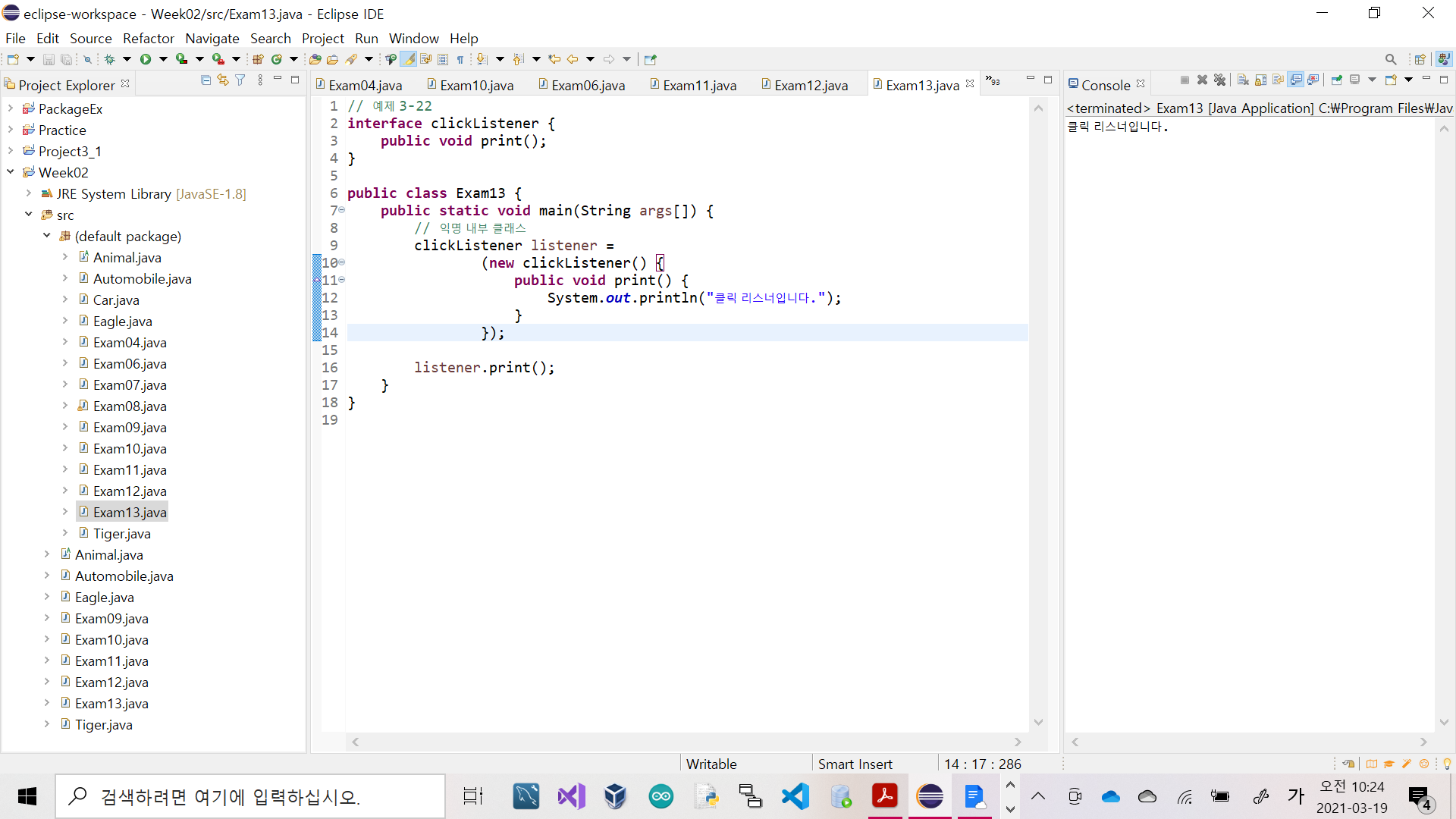
});

listener.print();

}

}

[실행결과]



[예제 7-2]

[소스코드]

// 예제 7-2

// Person 클래스에 toString() 작성

// Person 클래스에 Person 객체를 문자열로 리턴하는 toString() 메소드를 작성하라.

class Person {

String name, addr;

public Person(String name, String addr) {

this.name = name;

this.addr = addr;

}

@Override

public String toString() {

return "Person(" + name + "," + addr + ")";

}

}

public class ToStringEx {

public static void main(String args[]) {

Person p = new Person("아이유", "서울");

System.out.println(p.toString());

System.out.println(p); // p는 p.toString()으로 자동 변환

System.out.println(p + "입니다."); // p.toString() + "입니다"로 자동 변환

Person p2 = new Person("아이유", "서울");

System.out.println(p2.toString());

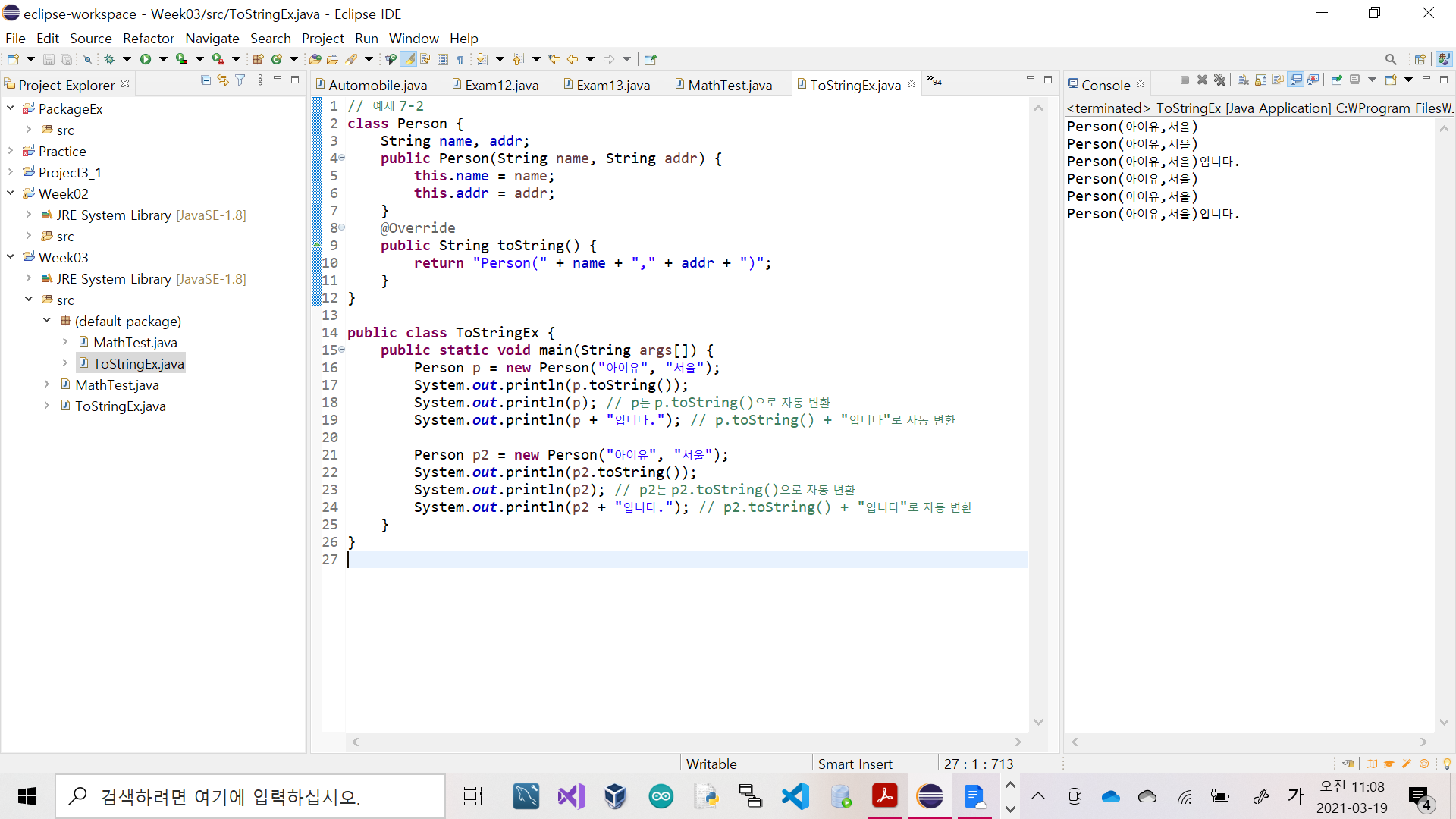
System.out.println(p2); // p2는 p2.toString()으로 자동 변환

System.out.println(p2 + "입니다."); // p2.toString() + "입니다"로 자동 변환

}

}

[실행결과]



[예제 7-5]

[소스코드]

// 예제 7-5

// Wrapper 클래스 활용

public class WrapperEx {

public static void main(String[] args) {

System.out.println(Character.toLowerCase('A')); // 'A'를 소문자로 변환

String x = "-123";

String y = "10";

System.out.print("x + y = ?");

System.out.println(Integer.parseInt(x) + Integer.parseInt(y));

Double d = Double.valueOf(3.14);

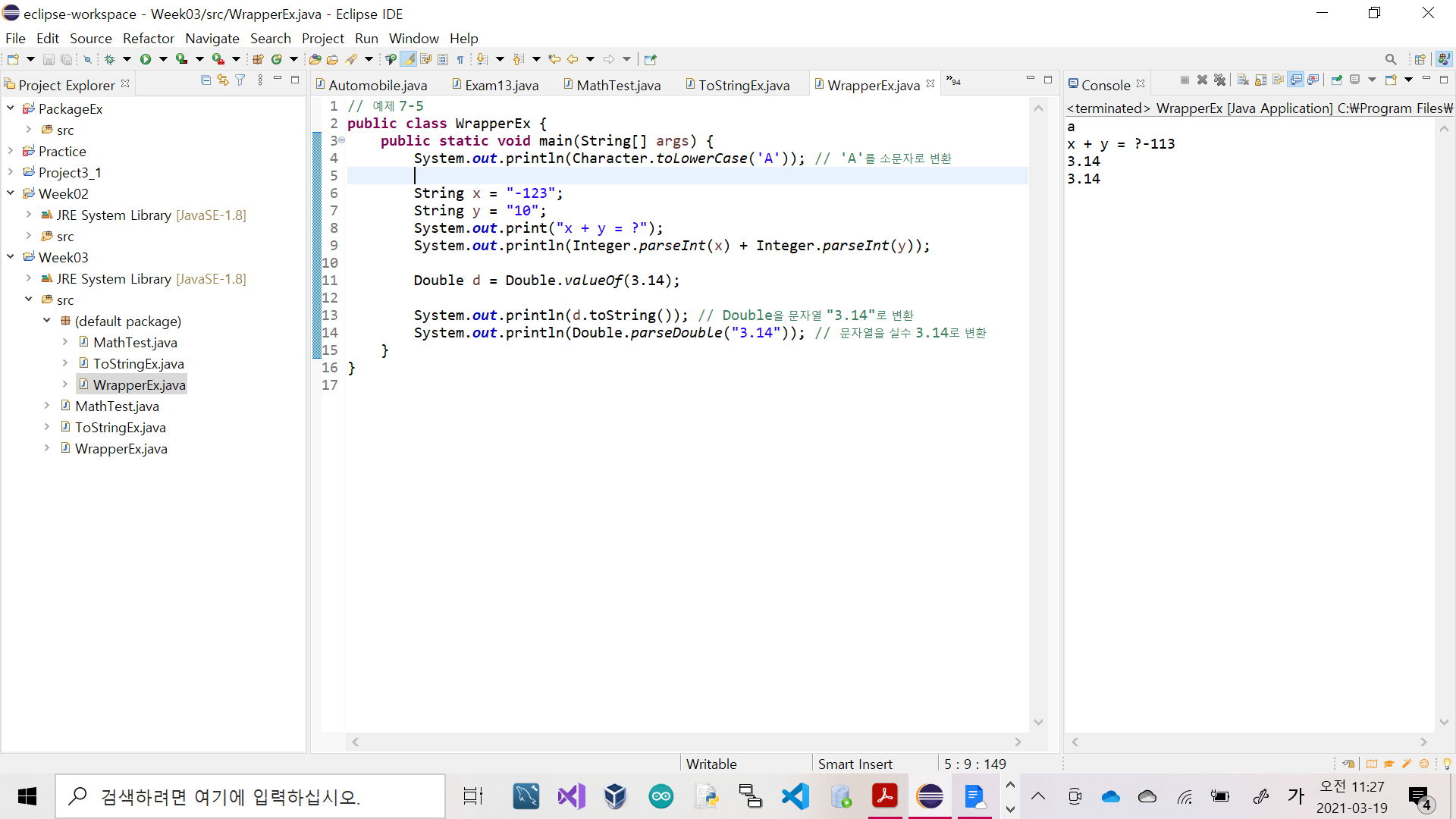
System.out.println(d.toString()); // Double을 문자열 "3.14"로 변환

System.out.println(Double.parseDouble("3.14")); // 문자열을 실수 3.14로 변환

}

}

[실행결과]



[예제 7-7]

[소스코드]

// 예제 7-7

// String 클래스 메소드 활용

public class StringEx {

public static void main(String[] args) {

String a = new String(" C#");

String b = new String(",C++");

System.out.println(a + "의 길이는 " + a.length()); // 문자열의 길이(문자 개수)

a = a.trim(); // 문자열 앞 뒤의 공백 제거

System.out.println(a + b);

String s[] = a.split(","); // 문자열 분리

for(int i=0; i<s.length; i++)

System.out.println("분리된 문자열" + i + ": " + s[i]);

String t[] = b.split(","); // 문자열 분리

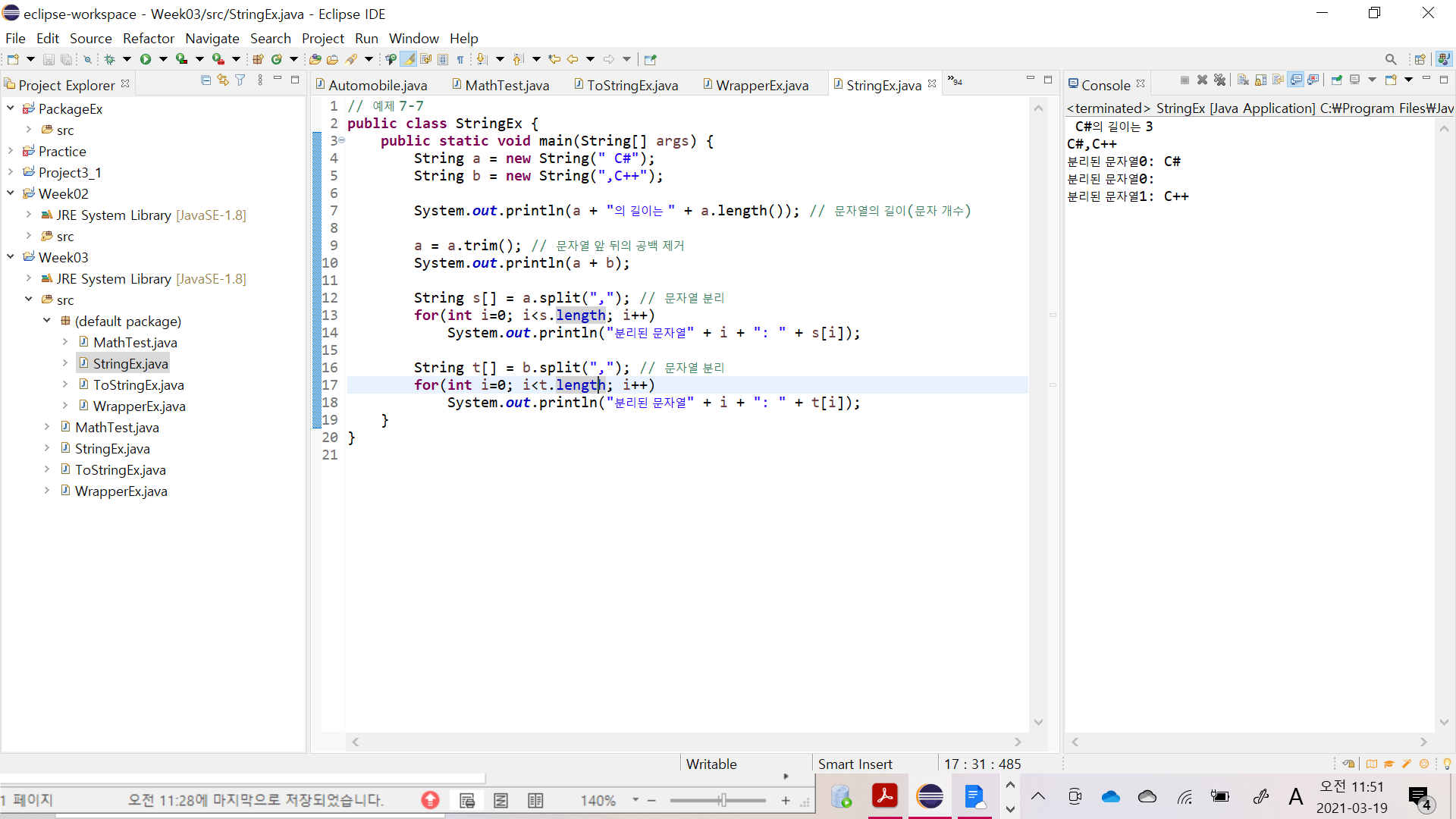
for(int i=0; i<t.length; i++)

System.out.println("분리된 문자열" + i + ": " + t[i]);

}

}

[실행결과]



[예제 7-10]

[소스코드]

// 예제 7-10

// Math 클래스 메소드 활용

public class MathEx {

public static void main(String[] args) {

System.out.println(Math.PI); // 원주율 상수 출력

System.out.println(Math.sqrt(4)); // 제곱근

System.out.println(Math.round(3.14)); // 반올림

// [1, 45] 사이의 정수형 난수 5개 발생

System.out.print("이번주 행운의 번호는 ");

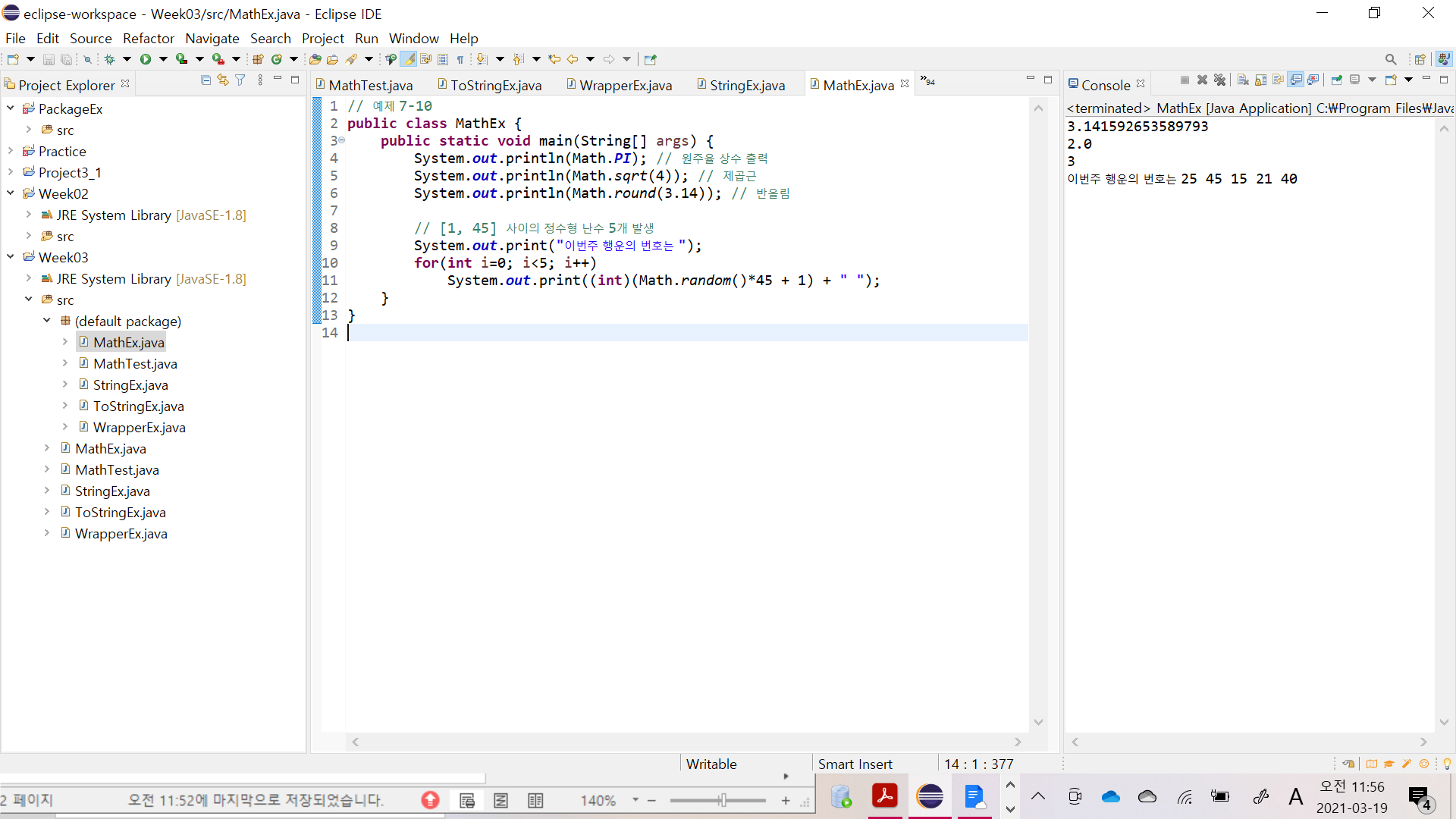
for(int i=0; i<5; i++)

System.out.print((int)(Math.random()\*45 + 1) + " ");

}

}

[실행결과]



[예제 7-11]

[소스코드]

// 예제 7-11

// Calendar를 이용하여 현재 날짜와 시간 알아내기 / 날짜 시간

import java.util.Calendar;

public class CalendarEx {

public static void main(String[] args) {

Calendar cal = Calendar.getInstance();

int year = cal.get(Calendar.YEAR);

int month = cal.get(Calendar.MONTH) + 1;

int day = cal.get(Calendar.DAY\_OF\_MONTH);

int dayOfWeek = cal.get(Calendar.DAY\_OF\_WEEK);

int hour = cal.get(Calendar.HOUR);

int hourOfDay = cal.get(Calendar.HOUR\_OF\_DAY);

int ampm = cal.get(Calendar.AM\_PM);

int minute = cal.get(Calendar.MINUTE);

System.out.print("현재" + year + "/" + month + "/" + day + "/");

System.out.print("(" + hourOfDay + "시)");

System.out.println(minute + "분");

}

}

[실행결과]

