**Report**

**For java Programming**

**학과 :**

**학번 :**

**이름 :**

컴퓨터공학과

20184071

김도현

**문제 1번**

1. 프로그램 소스코드 및 설명과 결과

.

**public** **class** Counter { //카운터 클래스 생성

**int** counter; //카운터 필드 생성

**void** up() {

counter ++; //counter 클래스 내의 counter 필드 값 증가

}

**int** getCount() {

**return** counter; //counter 클래스 내의 counter 필드 반환하는 메소드

}

**public** **static** **void** main(String[] args) {

Counter cn = **new** Counter(); //cn 객체에 새로운 counter 클래스를 할당

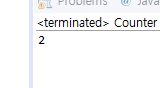
cn.up(); //counter 클래스의 up 메소드 사용 값은 1이됨, 초기화는 기본 0

cn.up(); //counter 클래스의 up 메소드 사용 값은 2가 된다

System.***out***.println(cn.getCount()); //cn객체가 가진 getCount 메소드를 사용하여 그 값 출력

}

}

실행결과 )) 

객체를 하나 더 할당후 up 메소드를 사용하고 출력하면 그 값은 1이 될것이다.

클래스는 각각 다른값을 담기때문이다.

**문제 2번**

1. 프로그램 소스코드 및 설명과 결과

**public** **class** Rectangle {

**int** x,y,w,h; //x y w h 필드 생성

**void** setX(**int** x) { //x 를 인자로 받는다 필드의 x 값 설정

**this**.x = x; //x 를 인자로 받았기에 필드 x 인지 인자 x 인지 구분하기위해 this 사용

}

**void** setY(**int** y) {//y 를 인자로 받는다 필드의 y 값 설정

**this**.y = y;//y 를 인자로 받았기에 필드 y 인지 인자 y 인지 구분하기위해 this 사용

}

**void** setW(**int** w) {//w 를 인자로 받는다 필드의 w 값 설정

**this**.w = w;//w 를 인자로 받았기에 필드 w 인지 인자 w 인지 구분하기위해 this 사용

}

**void** setH(**int** h) {//h 를 인자로 받는다 필드의 h 값 설정

**this**.h = h;//h 를 인자로 받았기에 필드 h 인지 인자 h 인지 구분하기위해 this 사용

}

**int** getArea() {//면적을 구하는 메소드

**return** **this**.w\***this**.h; //필드내의 w 와 h 를 곱하여 반환한다.

}

**void** print() {//사각형의 정보를 출력하는 메소드

System.***out***.println(String.*format*("가로 길이 :%d\n세로길이 : %d\n너비 : %d", w,h,w\*h));

//클래스내의 w h 필드를 사용하여 출력

System.***out***.println(String.*format*("x좌표 :%d\ny좌표 : %d", x,y));

//클래스 내의 x y 필드 출력

}

**public** **static** **void** main(String[] args) {

Rectangle rec = **new** Rectangle(); //Rectangle 클래스를 rec 객체에 할당

rec.setW(30); //rec 객체에 setW 를 사용하여 w 설정

rec.setH(50);//rec 객체에 setH를 사용하여 h 설정

rec.setX(5); //rec 객체에 setX를 사용하여 x 설정

rec.setY(6); //rec 객체에 setY를 사용하여 y 설정

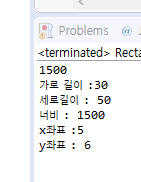
System.***out***.println(rec.getArea()); //rec 객체에 getArea 메소드는 반환만 되므로 그 값 출력

rec.print();//rec 객체가 가진 정보를 출력하는 print 메소드 호출

}

}

실행결과 ))



클래스는 빵틀과 같기 때문에 객체를 하나 더 생성하고 setW 등과 같은 설정메소드에 값을 바꾸면 rec 값과 새로 생성된 객체는 각각 다른값을 가지게 되고, 출력도 다르게 될 것 이다.

**문제 3번**

1. 프로그램 소스코드 및 설명과 결과

.

**public** **class** Date {

**int** year,month,day; //클래스의 필드 year month day 생성

**void** setYear(**int** year) { //year 를 인자로 받는다 필드의 year 값 설정

**this**.year = year; //필드의 year , 인자의 year 을 구분위해 this 명시

}

**void** setMonth(**int** month) {//month 를 인자로 받는다 필드의 month 값 설정

**this**.month = month;//필드의 month , 인자의 month 을 구분위해 this 명시

}

**void** setDay(**int** day) {//day 를 인자로 받는다 필드의 day 값 설정

**this**.day = day;//필드의 day, 인자의 day 구분을 위해 this 명시

}

**void** print1() {

System.***out***.println(**this**.year + "." + **this**.month + "." + **this**.day);

//Date 클래스의 year month day 필드 출력

}

**void** print2() {

String MonthEng = **null**; //switch Case 문에서 출력을 바꾸기 위한 String 변수 선언

**switch** (**this**.month) { //switch 문에서 month 필드를 가지고 판별한다.

**case** 1:

MonthEng = "January"; //1월이면 montheng 변경

**break**;

**case** 2:

MonthEng = "February "; //2월이면 montheng 변경, 마찬가지로 12월이면 까지 진행

**break**;

**case** 3:

MonthEng = "March";

**break**;

**case** 4:

MonthEng = "April";

**break**;

**case** 5:

MonthEng = "May";

**break**;

**case** 6:

MonthEng = "June";

**break**;

**case** 7:

MonthEng = "July";

**break**;

**case** 8:

MonthEng = "August";

**break**;

**case** 9:

MonthEng = "September";

**break**;

**case** 10:

MonthEng = "October";

**break**;

**case** 11:

MonthEng = "November";

**break**;

**case** 12:

MonthEng = "December";

**break**;

}

System.***out***.println(MonthEng + " " +**this**.day +","+ **this**.year); //switch 문이 끝났으면

//monthEng 변수와 day 필드 year 필드 출력

}

**public** **static** **void** main(String[] args) {

Date dt= **new** Date(); //dt 객체에 Date 클래스 할당

dt.setDay(12); //dt 객체가 가진 setDay 메소드 사용하여 day 필드를 12 로 변경

dt.setMonth(12);//dt 객체가 가진 setMonth 메소드 사용하여 month 필드를 12 로 변경

dt.setYear(2019);//dt 객체가 가진 setYear 메소드 사용하여 year 필드를 22019 로 변경

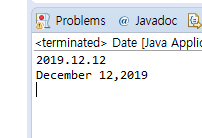
dt.print1();//dt 객체가 가진 print1 메소드 호출

dt.print2();//dt 객체가 가진 print2 메소드 호출

}

}

실행결과 ))



Set 메소드를 통해 값이 잘 설정되었고 month 에 따라 december 와 같이 나온다.

**문제 4번**

1. 프로그램 소스코드 및 설명과 결과

**import** javax.swing.JFrame;

**public** **class** FrameTest2 {

**public** **static** **void** main(String[] args) {

JFrame frm1 = **new** JFrame("프레임 1 번"); //frm1 에 JFrame 값 할당, 생성자가 프레임명이 필요하므로

//생성할때 생성자에 들어가는 인자를 넣어준다.

**int** x1,y1,x2,y2; //창을 움직이는데 필요한 x1 y1 x2 y2 변수 선언

x1 = 30;//기본값 선언부

y1 = 20;

x2 = 30;

y2 = 280;

frm1.setSize(300,200); //frm1 객체의 크기를 정해주고

frm1.setLocation(x1,y1); //위치 또한 정해준다

frm1.setVisible(**true**); //그 후 보여지게 설정한다.

JFrame frm2 = **new** JFrame("프레임 2번");//frm2 에 JFrame 값 할당, 생성자가 프레임명이 필요하므로

//생성할때 생성자에 들어가는 인자를 넣어준다.

frm2.setSize(300,200); //frm1 과 다르므로 setSize 메소드를 통해 크기 설정

frm2.setLocation(x2, y2); //위치도 설정해준다.

frm2.setVisible(**true**); //보여지게 설정한다

**try** { //thread 가 sleep 상태에서 오류가 날수 있으므로 try catch 문 로 오류를 방여한다.

**while** (x1 < 900 || x2 < 900) { //각 창의 좌표 2개가 900이 넘지 않을때까지 반복한다.

x1 +=(**int**)(Math.*random*()\*40) +1; //math의random 메소드를 사용후 \*40 을 한후 1 을더해

//1~41 까지 난수를 설정한다. 또한 math.random 의 반환값이 double 이므로 int 로 형변환을 해준다.

x2 += (**int**)(Math.*random*()\*40)+ 1;

Thread.*sleep*(100);//0.1초 쉰다

System.***out***.println(x1 +","+x2); //디버그용 x1 x2 좌표 출력

frm1.setLocation(x1, y1); //값이 바뀌면 frm1 의 값을 바꾼다

frm2.setLocation(x2,y2); //마찬자지로 frm2 의 x 값을 바꾼다.

}

// frm1.setVisible(false);

// frm2.setVisible(false);

//System.out.println(x1>x2);

System.***out***.println((x1 > x2 ? "1번 프레임" : "2번 프레임") + "승리"); //while 이 끝나면 출력 3항 연산자를 사용하여 불필요한 소스를 제거한다.

} **catch** (InterruptedException e) {

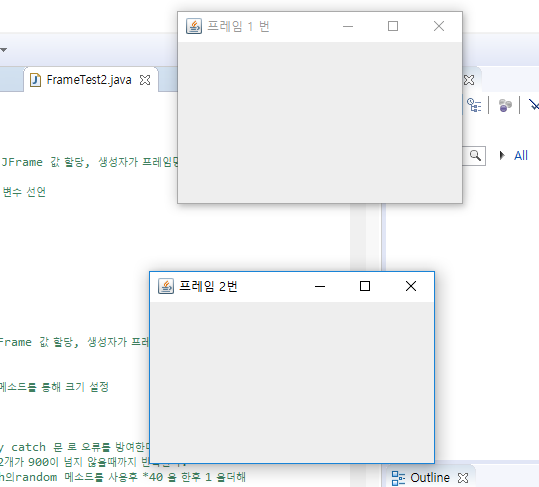
e.printStackTrace();

}

}

}

실행결과 ))





프레임이 0.1 초 단위로 움직이고, 몇번 프레임이 이겼는지 출력이 된다. 또한 디버그용 프린트도 잘 출력된다.