<JAVA PROGRAMMING LAB -2장: 조건과 반복>

◇ 문제 및 요구하는 것

◆ 프로그램 실행결과



**1. Counter클래스를 생성하고, CounterTest클래스를 만든다. 이 Test클래스로 Counter의 필드와 메소드를 접근하는 프로그램을 만들어라**

◇ 제목

◇ 제목

◇ 제목

**public** **class** Counter {

 **int** counter;

 **void** up() {

 counter++;

 }

 **int** getCount(){

 **return** counter;

 }

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

 CounterTest a=**new** CounterTest();

 a.test();

 }

}

**class** CounterTest{

 **void** test(){

 Counter b= **new** Counter();

 b.counter=100;

 b.up();

 b.up();

 System.***out***.print(b.getCount());

 }

}

◇ 제목

◆ DESCRIPTION

◇ 제목

◆ 프로그램 설명

◇ 프로그램 소스

◆ DESCRIPTION

CounterTest객체를 만들고 a변수로 이 객체를 참조했다. a변수가 참조하는 객체의 test메소드를 호출한다. 그럼 이 메소드에서는 Counter객체를 만들고 b변수로 이 객체를 참조한다. 이후 b변수가 참조하는 객체의 counter필드를 100으로 초기화하고 up메소드를 2번 호출한다. Up메소드는 counter필드의 값을 1올리는 기능을 하는데 2번 호출했고 처음에 필드가 100으로 초기화되었으니 최종적으로 102가 저장될 것이다.

마지막으로 b변수가 참조하는 객체의 getCount메소드를 호출한다. 그러면 counter을 반환하게 되는데 이 반환 값을 다시 출력하니 102가 출력되게 된다.

< JAVA PROGRAMMING LAB -2장: 조건과 반복>

◇ 문제 및 요구하는 것

◆ 프로그램 실행결과



**2. 사각형을 나타내는 클래스 Rectangle을 설계하고 RectangleTest클래스를 작성하고 객체를 생성하여 테스트하라.**

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

Scanner객체를 만들고 input변수가 이 객체를 참조한다. 이후 Rectangle객체를 만들고 r변수가 이 객체를 참조한다. input변수가 참조하는 객체의 nextInt메소드를 이용해서 정수를 입력받은 후, r변수가 참조하는 객체의 w필드에 정수 값을 저장한다. 같은 방식으로 h필드와 x,y필드에도 정수 값을 저장한다. 그리고 r변수가 참조하는 객체의 getArea메소드를 호출한다. 이 메소드가 호출되면 w에 h를 곱한 값을 반환한다. 이 반환 값은 정수형 변수 n에 저장된다. 마지막으로 r변수가 참조하는 객체의 print메소드에 n을 보낸다. 그럼 print메소드에서는 필드의 값을 토대로 사각형에 대한 정보를 출력하게 될 것이다. 이처럼 클래스 내에 필드로 미리 지정되어 있으면, 메소드에서 필드를 자유롭게 가져올 수 있다. C언어처럼 매개변수로 일일이 변수를 보낼 필요가 없다는 것이다.

**import** java.util.\*;

**class** Rectangle {

 **int** x,y,w,h;

 **int** getArea() {

 **return** w\*h;

 }

 **void** print(**int** n) {

 System.***out***.println("사각형의 x,y좌표는"+"("+x+","+y+")입니다.");

 System.***out***.println("사각형의 가로길이는"+w+"입니다.");

 System.***out***.println("사각형의 세로길이는"+h+"입니다.");

 System.***out***.println("사각형의 넓이는"+n+"입니다.");

 }

}

**public** **class** RectangleTest{

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

 Scanner input= **new** Scanner(System.***in***);

 Rectangle r= **new** Rectangle();

 System.***out***.print("사각형의 가로 길이를 입력해주세요:");

 r.w=input.nextInt();

 System.***out***.print("사각형의 세로 길이를 입력해주세요:");

 r.h=input.nextInt();

 System.***out***.print("사각형의 x좌표를 입력해주세요:");

 r.x=input.nextInt();

 System.***out***.print("사각형의 y좌표를 입력해주세요:");

 r.y=input.nextInt();

 **int** n=r.getArea();

 r.print(n);

 }

}

◇ 제목

◇ 제목

◇ 프로그램 소스 및 설명

< JAVA PROGRAMMING LAB -2장: 조건과 반복>

◇ 문제 및 요구하는 것

◆ 프로그램 실행결과



**3. 날짜를 나타내는 Date클래스를 만들어라, Date는 연도,월,일 등을 받도록 해라. 이후 print1,print2메소드에는 날짜를 출력하는 소스를 작성하여라.**

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

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

 **int** year;

 **int** month;

 **int** day;

 **void** print1(){

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

 }

 **void** print2(){

 **switch**(month) {

 **case** 1:

 System.***out***.println("January "+day+","+year);

 **break**;

 **case** 2:

 System.***out***.println("February "+day+","+year);

 **break**;

 **case** 3:

 System.***out***.println("March "+day+","+year);

 **break**;

 **case** 4:

 System.***out***.println("April "+day+","+year);

 **break**;

 **case** 5:

 System.***out***.println("May "+day+","+year);

 **break**;

 **case** 6:

 System.***out***.println("June "+day+","+year);

 **break**;

 **case** 7:

 System.***out***.println("July "+day+","+year);

 **break**;

 **case** 8:

 System.***out***.println("August "+day+","+year);

 **break**;

 **case** 9:

 System.***out***.println("September "+day+","+year);

 **break**;

 **case** 10:

 System.***out***.println("November "+day+","+year);

 **break**;

 **case** 11:

 System.***out***.println("October "+day+","+year);

 **break**;

 **case** 12:

 System.***out***.println("December "+day+","+year);

 **break**;

 **default**:

 **break**;

 }}

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

 Scanner input= **new** Scanner(System.***in***);

 System.***out***.print("연도를 입력해주세요:");

 Date d=**new** Date();

 d.year=input.nextInt();

 System.***out***.print("월을 입력해주세요:");

 d.month=input.nextInt();

 System.***out***.print("일을 입력해주세요:");

 d.day=input.nextInt();

 d.print1();

 d.print2(); }}

◇ 제목

◇ 제목

◇ 프로그램 소스

자바는 main메소드에서 처음 시작한다. 이 main메소드는 어느 클래스에 있든 상관이 없으며, 어느 클래스에 소속되었다는 느낌은 주지 않을 것이다. 왜냐하면 main이 속한 Date클래스의 year, month, day필드를 접근하기 위해 인스턴스화를 진행하고 있기 때문이다. 보통의 메소드는 클래스의 필드를 자유롭게 접근할 수 있지만 main은 이처럼 특별하다.

그렇다면 왜 main의 접근지정자는 public이며 static이 옆에 붙는 것인가? 일단 main은 소스 파일에서 없어서는 안될 주요 메소드이므로, 다른 라이브러리에서 현재 java파일을 가져올 때도 충분히 사용될 수 있다. 다른 라이브러리에서 가져올 때 같은 패키지의 파일을 가져온다고 보장할 수가 없으니 public이 붙는 것이 옳다.

static은 정적으로 사용한다는 의미보단, 객체 생성 없이 사용할 수 있다고 생각하면 편하다. 자바 프로그램이 실행되면 static 부분은 바로 메모리에 올라가 상주하게 된다. 그렇기 때문에 main이 클래스에 속해 있다고 보기는 애매하고 java형식 상 클래스 안에 있다고 보는 것이 맞다.

Main에 대한 설명은 이쯤 하고 main메소드 내의 소스부터 살펴보자. 먼저 Scanner객체를 만들고 input변수로 이 객체를 참조했다. 보통 이럴 때 Scanner객체의 이름을 input이라고 한다. 같은 클래스에서 여러 객체가 생성될 수 있고, 그 객체들은 다르게 동작된다. 즉 객체의 이름을 참조 변수로 하지 않으면 여러 객체가 생길 때 혼동이 생길 수 있다. 다음 소스들을 보면 input객체의 nextint메소드를 통해 정수를 입력받는다. 이 정수들은 d객체의 필드에 차곡차곡 저장된다.

이후 d객체의 print1메소드를 호출한다. 그러면 단순히 입력한 정수들을 출력하게 된다.

또 d객체의 print2메소드도 호출한다. 그러면 switch문으로 가게 되는데 month필드에 저장된 값에 따라 다른 문장을 출력하게 된다.

Print2메소드의 리턴 값은 없도록 만들어졌으니 메소드에서 return자체가 불가능하고

Break로 끝나는 모습을 볼 수 있다.

◇ 제목

◇ 제목

◇ 프로그램 설명

< JAVA PROGRAMMING LAB -2장: 조건과 반복>

◇ 문제 및 요구하는 것

◆ 프로그램 실행결과



4. 두 개의 윈도우를 만들고 두 윈도우가 경주하는 프로그램을 만들어라.

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

◇ 제목

Swing패키지의 모든 java파일을 가져왔다. 이 패키지에는 GUI윈도우를 구성하게 하는 라이브러리 파일이 존재한다. 그리고 그 파일 안에 있는 윈도우를 구성하는 핵심 클래스가 바로 ‘JFrame’이다. 그래서 JFrame을 객체로 만들고 f변수가 이 객체를 참조시켰다. 윈도우로 경쟁을 해야 하니 두 개의 객체를 만들 필요가 있다. 그래서 f2객체도 만들게 된다. 다음은 util패키지에 있는 Random클래스를 객체로 만들었다. 이 객체를 r객체라고 부르겠다. For문 안 쪽 소스를 확인해보자. R객체의 nextint메소드에 인수를 넣어서 호출하면 그 인수 값 이하인 정수 랜덤 값을 자동으로 생성한다. 이 랜덤값을 ran1에 넣고 같은 방식으로 ran2에도 저장하였다. 이후 f객체의 setsize메소드로 윈도우 크기를 정하고 setlocation메소드로 윈도우 위치를 정하고 setvisible메소드로 윈도우를 보이게 한다. 이게 f객체로부터 윈도우를 생성하는 과정이다. f2객체도 마찬가지 방식으로 윈도우를 생성시킨다. 근데 문제에서는 경주가 목적이다. 그러니 f객체 setlocation의 첫 번째 인수(x좌표)에 l1(ran1+l1)을 넣고, f2에도 l2값을 넣으면 랜덤 값에 따라 윈도우 위치가 다른 것을 볼 수 있다. 이 반복을 400번하면 l1,l2값에 랜덤 값이 계속 누적되고, 두 윈도우가 앞으로 나가며 경쟁하는 것 처럼 보이게 된다.

**import** java.util.\*;

**import** javax.swing.\*;

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

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

 **int** l1=30,l2=20;

 JFrame f=**new** JFrame("Frame Test");

 JFrame f2=**new** JFrame("Frame Test");

 Random r=**new** Random();

 **for**(**int** i=0;i<400;i++) {

 **int** ran1=r.nextInt(5);

 **int** ran2=r.nextInt(5);

 f.setSize(300,200);

 f.setLocation(l1,20);

 f.setVisible(**true**);

 f2.setSize(300,200);

 f2.setLocation(l2,220);

 f2.setVisible(**true**);

 l1+=ran1;

 l2+=ran2;

 }

 }

}

◇ 제목

◇ 제목

◇ 프로그램 소스 및 설명