자바 프로그래밍 12장

|  |  |
| --- | --- |
| 분반 | 1분반 |
| 학번 | 2012136132 |
| 성명 | 최강석 |

296PAGE 1번문제

1. 문제 : 다음과 같은 인터페이스들을 정의하라

public interface Movable

{

void move(int dy, int dy);

}

본문에 등장하는 2차원 도형인 원, 사각형, 삼각형 등이 위의 인터페이스를 구현하도록 수정하라. move() 메소드는 도형의 기준점을 이동한다. main()에서 Movable 객체배열을 생성하고 배열의 각 원소에 대하여 move()를 호출하여 랜덤하게 객체를 이동시키는 프로그램을 작성하라

1. 알고리즘 :

Shape클래스를 만들어준다. (설정자, 접근자, 생성자등의 정의)

interface movable을 만들어준다.

shape을 상속받고 인터페이스 Movable을 이용한, 사각형, 원의 클래스를 정의해준다.

(설정자, 접근자, 생성자등의 정의)

각 도형의 클래스에 move에 대한 정의를 해준다. move에는 받은 매개변수로 shape에 있는 setRandom을 호출을 해 랜덤이동을 대입하고나서 도형의 좌표들을 출력하는 함수를 호출하게된다.

메인함수에서는 각 세개의 객체들을 배열에 만들어주고

세개의 배열에 있는 객체들을 각각

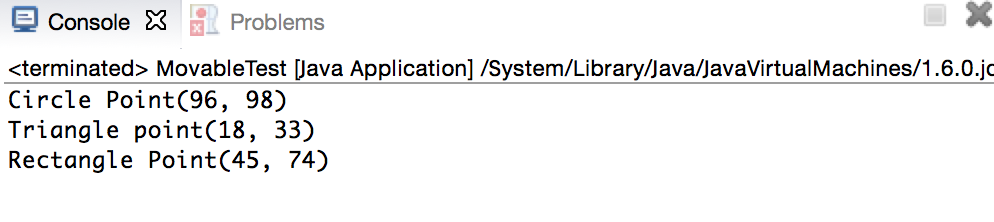
Math.random을 통해 move의 매개변수로 넣어주어준다.

그러면 좌표에 대한 정보가 출력이 된다.

3. 코드 :

|  |
| --- |
| **class** Shape {//Shape 클래스  **private** **int** x;//도형의 좌표 중 x.  **private** **int** y;//도형의 좌표 중 y.    **public** Shape(){//생성자.  **int** x= 0;  **int** y= 0;  }  **public** Shape(**int** x, **int** y){  //매개변수가 있을 경우의 생성자.  **this**.x=x;  **this**.y=y;  }    **public** **void** PrintXY() {  System.***out***.println("("+ **this**.x +", "+**this**.y +")" );  //좌표의 정보를 출력한다.  }  **public** **void** setRandom(**int** dx, **int** dy){  **this**.x+= dx;  **this**.y+= dy;  //램덤값으로 받은 dx와 dy를 기존에있는 좌표에 +=연산을 해주어서  //랜덤값만큼 이동하도록 나타내준다.  }    }  **interface** Movable {//문제에 나와있는 인터페이스.  **void** move(**int** dx, **int** dy);  }// 인터페이스  **class** Rectangle **extends** Shape **implements** Movable {  //사각형 클래스 다중상속을 받는다.  **private** **int** w, h;  **public** Rectangle(){//사각형의 설정자 매개변수가 없을경우.  **super**();  **this**.w=0;  **this**.h=0;  }  **public** Rectangle(**int** x, **int** y, **int** w, **int** h){  //사각형의 설정자 매개변수를 모두 받을경우.  **super**(x,y);  **this**.w=w;  **this**.h=h;  }    //오버라이드.  **public** **void** move(**int** dx, **int** dy) {  //Movable인터페이스의 move에 대한 함수를 정의해준다.  **super**.setRandom(dx,dy);//랜덤 값을 Shape의 x,y를 설정하는 함수로 넘겨준다.    System.***out***.print("Rectangle Point");  **super**.PrintXY();//좌표에 대한 정보를 출력해준다.  }  }  **class** Triangle **extends** Shape **implements** Movable {  //삼각형 클래스 다중상속을 받는다.  **int** a, b, c;//삼각형을 나타낼 세변의 변수.  **public** Triangle(){  //삼각형의 설정자  **super**();  **this**.a=0;  **this**.b=0;  **this**.c=0;  }  **public** Triangle(**int** x, **int** y, **int** a, **int** b, **int** c){  //삼각형의 설정자 매개변수를 모두 받을 경우.  **super**(x,y);  **this**.a=a;  **this**.b=b;  **this**.c=c;  }  @Override  **public** **void** move(**int** dx, **int** dy) {  //move  **super**.setRandom(dx, dy);  //도형의 좌표를 설정해주기위 Shape의 setRandom함수를  //호출해줍니다.  System.***out***.print("Triangle point");  **super**.PrintXY();  //현재 도형의 좌표를 출력을 해줍니다.  }  }  **class** Circle **extends** Shape **implements** Movable {  //원 클래스 다중상속을 받는다.  **private** **int** r;//반지름의 길이.    **public** Circle(){//매개변수가 없을 때의 생성자.  **super**();  **this**.r=0;  }  **public** Circle(**int** x, **int** y, **int** r){  //매개변수를 전부 받을 때의 생성자.  **super**(x,y);  **this**.r=r;  }  @Override  **public** **void** move(**int** dx, **int** dy) {  //도형의 좌표를 설정해주기위 Shape의 setRandom함수를  //호출해줍니다.  **super**.setRandom(dx, dy);  System.***out***.print("Circle Point");  **super**.PrintXY();  //현재 도형의 좌표를 출력을 해줍니다.  }  }  **public** **class** MovableTest {  **public** **static** **void** main(String[] args) {    Movable[] array = **new** Movable[3];// Movable 형 배열 할당.  array[0] = **new** Circle(); //원 객체.  array[1] = **new** Triangle();//삼각형 객체.  array[2] = **new** Rectangle();//사각형 객체.    **int** dx;//x에 랜덤값을 넣어줄 변수.  **int** dy;//y에 랜덤값을 넣어줄 변수.  **for** (**int** i = 0; i < array.length; i++) {  dx = (**int**) (Math.*random*()\*99 + 1);//x좌표의 더해줄 랜덤값  dy = (**int**) (Math.*random*()\*99 + 1);//y좌표의 더해줄 랜덤값    array[i].move(dx, dy);//move에 매개변수로 랜덤값들을 넣어준다.  }  }  } |

4. 결과



296PAGE 2번문제

1. 문제 : 다음과 같은 인터페이스들을 정의하라.

public interface Drawable

{

void draw();

}

본문에 등장하는 2차원 도형인 원, 사각형, 삼각형 등이 위의 인터페이스를 구현하도록 수정하라. draw() 메소드에서는 실제로 그리지는 않고 메시지만을 출력하라. main()에서 Drawable 객체배열을 생성하고 배열의 각 원소에 대하여 draw()를 호출하여 랜덤하게 객체를 그리는 프로그램을 작성하라

1. 알고리즘 :

1번 문제와 같이 shape과 각 도형에대한 정의를 해준다.

인터페이스 Drawable을 선언해주고 이번엔 각 도형의 클래스 안에

draw에 대한 정의를 해준다 (문자열로 출력)

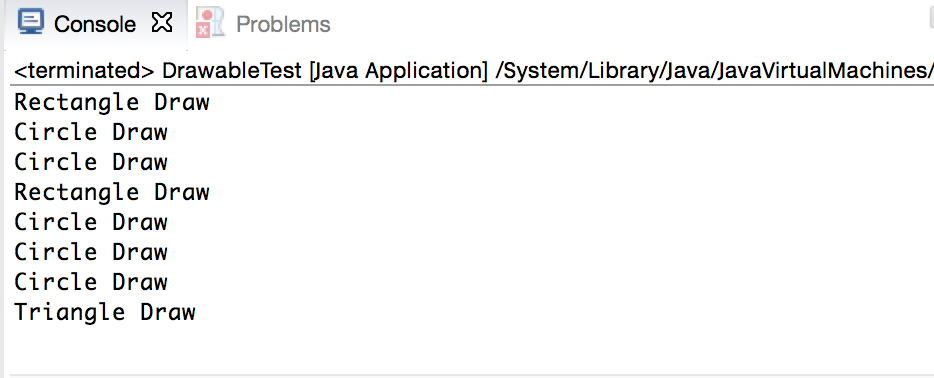
메인함수에는 세개의 배열을 만들어 세개의 객체들을 넣어준다.

for문을 이용해서 10번의 반복을 돌리고 Math.random()\*3+0 을 이용해서

각 루프마다 배열이 랜덤으로 출력을 하게 해준다.

1. 코드 :

|  |
| --- |
| **class** Shape {//Shape 클래스  **private** **int** x;//도형의 좌표 중 x.  **private** **int** y;//도형의 좌표 중 y.    **public** Shape(){//생성자.  **int** x= 0;  **int** y= 0;  }  **public** Shape(**int** x, **int** y){  //매개변수가 있을 경우의 생성자.  **this**.x=x;  **this**.y=y;  }    }  **interface** Drawable {//문제에 나와있는 인터페이스.  **void** draw();  }// 인터페이스  **class** Rectangle **extends** Shape **implements** Drawable {  //사각형 클래스 다중상속을 받는다.  **private** **int** w, h;  **public** Rectangle(){//사각형의 설정자 매개변수가 없을경우.  **super**();  **this**.w=0;  **this**.h=0;  }  **public** Rectangle(**int** x, **int** y, **int** w, **int** h){  //사각형의 설정자 매개변수를 모두 받을경우.  **super**(x,y);  **this**.w=w;  **this**.h=h;  }  **public** **void** draw(){//사각형 draw 출력.  System.***out***.println("Rectangle Draw");  }    //오버라이드.    }  **class** Triangle **extends** Shape **implements** Drawable {  //삼각형 클래스 다중상속을 받는다.  **int** a, b, c;//삼각형을 나타낼 세변의 변수.  **public** Triangle(){  //삼각형의 설정자  **super**();  **this**.a=0;  **this**.b=0;  **this**.c=0;  }  **public** Triangle(**int** x, **int** y, **int** a, **int** b, **int** c){  //삼각형의 설정자 매개변수를 모두 받을 경우.  **super**(x,y);  **this**.a=a;  **this**.b=b;  **this**.c=c;  }  **public** **void** draw(){//삼각형 draw 출력.  System.***out***.println("Triangle Draw");  }  }  **class** Circle **extends** Shape **implements** Drawable {  //원 클래스 다중상속을 받는다.  **private** **int** r;//반지름의 길이.    **public** Circle(){//매개변수가 없을 때의 생성자.  **super**();  **this**.r=0;  }  **public** Circle(**int** x, **int** y, **int** r){  //매개변수를 전부 받을 때의 생성자.  **super**(x,y);  **this**.r=r;  }  **public** **void** draw(){//원 draw 출력.  System.***out***.println("Circle Draw");  }    }  **public** **class** DrawableTest {  **public** **static** **void** main(String[] args) {    **int** r; //0에서 2까지의 랜덤변수를 받을 변수입니다.(도형이 3가지)  Drawable[] array = **new** Drawable[3];  array[0]= **new** Circle();//0의 원소에는 원객체를 넣어준다.  array[1]= **new** Rectangle();//사각형을 넣어줌.  array[2]= **new** Triangle();//삼각형을 넣어줌.    **for**(**int** i= 0; i<10; i++ ){//10개정도 랜덤호출하기  r= (**int**) (Math.*random*()\*3+0);//r에는 0부터 2의 값이 랜덤으 대입이 됨.  array[r].draw();//배열이 랜덤으로 출력.  }  }  } |

1. 결과 

296PAGE 3번문제

1. 문제 : 비디오 플레이어는 play와 stop라는 조작을 할 수 있다. 따라서 인터페이스 Controllable을 아래와 같이 정의할 수 있다. Controllable 인터페이스를 구현하는 VideoPlayer 클래스를 Test 클래스의 main()안에 무명 내부 클래스로 작성하고 테스트하여 보자.

public interface Controllable {

void play();

void stop();

}

1. 알고리즘 :

Controllable인터페이스를 만들어 준다.

그안에는 시작을 하기위한 play(), stop()이 있다.

test 클래스안에 무명클래스 객체를 만들어 준다.

(Controllable c1 = new Controllable)

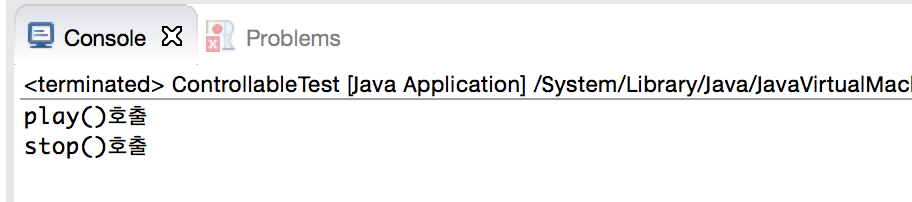
그안에 play와 stop을 호출하면 호출했다는 문자열을 넣어준다.

play와 stop을 호출해준다.

3. 코드 :

|  |
| --- |
| **interface** Controllable{ //Controllable 인터페이스.  **void** play(); //시작을 하기위한 함수.  **void** stop(); //정지를 하기위한 함수.  }  **class** ControllableTest {  **public** **static** **void** main(String[] args) {  Controllable c1 = **new** Controllable(){  //Test 메인안의 무명클래스  **public** **void** play(){ //시작을 하기위한 함수 정의.  System.***out***.println("play()호출");  }  **public** **void** stop(){ //정지를 하기위한 함수 정의.  System.***out***.println("stop()호출");  }  };  c1.play(); //play()호출;  c1.stop(); //stop()호출;  }  } |

4. 결과



296PAGE 4번문제

1. 문제 : Person이라는 클래스를 정의하라 Person은 이름, 키를 필드로 가진다. Perosn은 본문에 나오는 Comparable 인터페이스를 구현한다. 이 인터페이스를 이용하여서 가장 키 큰 사람의 이름을 반환하는 메소드 getMaximum(Person[] array)을 구현하고 테스트하라.
2. 알고리즘 :

interface Comparable을 만들어준다.

그안에는 반환형이 int인 객체를 비교하는 함수가 있다.

person클래스를 만들어주어 인터페이스 Comparable을 implements한다.

person은 이름과 키를 필드로 가지고 있고 이에대한 생성자와 설정자, 접근자를 정의해준다.

인터페이스 Comparable compareTo는 객체의 키를 비교해서 크면 1을 반환해주고

같으면 0을 반환 작은경우는 -1을 반환을 해주게 한다.

메인함수안에 반환형이 static String형인 getmaximum을 정의해 준다.

이 함수는 Person 배열을 매개변수로 받고 이 객체들을 삼항연산자를

통해 비교해주어 키가제일 큰 값인 객체의 이름을 반환한다.

메인에는 person 배열을 만들어 5개의 person객체를 만들어주고 이름과 키를

대입을 해준다. 그리고 getMaximum(array)호출을 통한 출력을 해준다.

1. 코드 :

|  |
| --- |
| **interface** Comparable {//인터페이스  **int** compareTo(Object other);  //객체를 비교하기위한 함수 선언.  }  **class** Person **implements** Comparable {  //Person클래스, Comparable인터페이스를 사용.  **private** String name;//이름.  **private** **int** height;//키.    **public** Person(String name, **int** height){  //생성자에 대한정의 필드값을 설정해준다.  **this**.name=name;  **this**.height=height;  }  **public** **int** getHeight(){  **return** height;//키를 반환해준다.  }  **public** String getName(){  **return** name;//이름을 반환해준다.  }  **public** **int** compareTo(Object other) {  **if** (**this**.height > ((Person) other).height)  **return** 1;//객체의 키를 비교해서 크면 1을 반환.  **else** **if** (**this**.height == ((Person) other).height)  **return** 0;//객체의 키를 비교해서 같으면 0을 반환.  **else**  **return** -1;//키를 비교해서 작은경우 -1을 반환.  }  }  **public** **class** PersonTest {  **public** **static** String getMaximum(Person[] array){  //메인함수가 static이므로 static String형으로 반환값을 설정.    Person max = array[0];//첫배열의 값을 제일 큰사람으로 설정.    **for**(**int** i =1; i<array.length; i++){  //객체들를 삼항연산자를 통해 비교해주어  //키가 가장 큰 사람의 객체를 max에 대입을 해준다.  max = (max.getHeight() < array[i].getHeight() ? array[i] : max);  }    **return** max.getName();//가장 큰 객체의 이름을 반환해준다.  }  **public** **static** **void** main(String[] args) {    Person[] array = **new** Person[5];//5개의 사람객체를 만들기 위한 배열할당.  //각 배열의 원소에 대해 이름과, 키를 매개변수로 하는 person클래스 객체를 만들어준다.  array[0] = **new** Person("김구십", 190);  array[1] = **new** Person("최칠십", 170);  array[2] = **new** Person("홍칠일", 171);  array[3] = **new** Person("박육공", 160);  array[4] = **new** Person("김구삼", 193);      System.***out***.println("키가 제일 큰 사람은 " + *getMaximum*(array) + " 입니다");  //getMaximum에서는 가장 큰 사람이 반환하게 된다.  }  } |

1. 결과

