**Report**

**For java Programming**

**학과 :**

**학번 :**

**이름 :**

컴퓨터공학과

20184071

김도현

**문제 1번**

1. 프로그램 소스코드 및 설명과 결과

import java.awt.Graphics;

import java.util.ArrayList;

import javax.swing.JFrame;

class Vehicle extends JFrame { //JFrame 을 상속받는 Vehicle 생성

public Vehicle() { //vehicle 클래스의 생성자를 선언한다 > 상속을 받으면 상속받은 클래스도 이 생성자 호출

super(); //JFrame 의 생성자를 명시적으로 호출해준다.

setSize(200, 120); //상속받아온 setSize 메소드를 사용하여 윈도우 크기 조절

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true); //윈도우에 보여주기를 시작한다.

}

@Override //부모 클래스인 Jframe 이 가진 paint 메소드를 재정의 한다.

public void paint(Graphics g) {

super.paint(g); //부모클래스의 paint 를 호출하고

g.drawRect(50, 20, 100, 50); //paint 뿐 아니라 사각형을 그리는 메소드도 호출한다.

}

}

class Bus extends Vehicle { //Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Bus 클래스 생성

@Override //Vehicle 클래스가 가진 paint 메소드를 재정의함을 표기

public void paint(Graphics g) {

super.paint(g); //부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawArc(90 + 30, 60, 20, 20, 0, 360); //인자로 받은 g 객체에 바퀴를 그려준다.

g.drawArc(30 + 30, 60, 20, 20, 0, 360); //인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그려준다.

g.drawRect(20 + 30 + 80, 30, 20, 20); //인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(60 + 30, 40, 20, 30);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(60 + 30, 40, 10, 30);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawString("BUS !", 50 + 30, 100);//인자로 받은 g 객체에 drawString 메소드를 사용하여 문자를 그려준다.

}

}

class Car extends Vehicle {//Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Car 클래스 생성

@Override/ / Vehicle 클래스가 가진 paint 메소드를 재정의함을 표기

public void paint(Graphics g) {//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

super.paint(g);//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawRect(150, 45, 25, 25);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(25, 45, 25, 25);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawArc(60, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그린다.

g.drawArc(122, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그린다.

g.drawString("CAR !", 50 + 30, 100); //인자로 받은 g 객체에 drawString 메소드를 사용하여 Car! 를 그려준다,

}

}

class Truck extends Vehicle {//Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Truck 클래스 생성

@Override

public void paint(Graphics g) {

super.paint(g);//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawLine(120 + 30, 70, 145 + 30, 70); //인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawLine(145 + 30, 70, 145 + 30, 50);//인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawLine(145 + 30, 50, 120 + 30, 40);//인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawArc(40 + 30, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 을 사용하여 바퀴를 그려준다,

g.drawArc(122 + 30, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 을 사용하여 바퀴를 그려준다,

g.drawString("Truck !", 50 + 30, 100); //인자로 받은 g 객체에 drawString 을 사용하여 Truck 을 그려준다.

g.drawString("Express", 50 + 30, 50);//인자로 받은 g 객체에 drawString 을 사용하여 Express 을 그려준다.

}

}

public class RacingCarFrame {

public static void main(String[] args) {

Vehicle truck = new Truck(); //truck 인스턴스에 Truck 객체를 할당

Vehicle car = new Car(); //car 인스턴스에 Car 객체를 할당

Vehicle bus = new Bus(); //bus 인스턴스에 Bus 객체릃 할당.

//3개 모두 Vehicle 을 상속받았으므로 Vehicle 타입의 인스턴스로 하면 된다.

Vehicle[] list = { truck,car,bus }; //Vehicle 인스턴스를 담는 list 를 생성, 인자는 truck car bus 인스턴스 들어감.

int y = 50; //프레임의 위치를 정해줄 y 좌표의 초기값 설정

for (Vehicle V : list) { //Vehicle 인스턴스인 V 변수에 list 각각 인덱스 할당 > for each 문의 특징이다.

V.setLocation(50, y); //출발점은 모두 같으므로 50 ,y 로 해준다.

y += 150; //y 좌표를 150 씩 증가하여 보기 좋게 만들어준다.

}

try { //thread 의 sleep 메소드가 작동하지 않을 경우를 위해 try catch 문 사용

boolean Start = true; //while 문 실행 유무를 위해 정해준다.

do {

Thread.sleep(100); //100 밀리세컨드 쉰다.

for (Vehicle V : list) {//Vehicle 인스턴스인 V 변수에 list 각각 인덱스 할당 > for each 문의 특징이다.

int Random = (int)(Math.random() \* 50) + 1; //Math 클래스의 random 메소드를 사용하여 1~51 까지 랜덤 숫자 생성

V.setLocation(V.getLocation().x + Random, V.getLocation().y);

//for 문을 돌리며 할당받은 Vehicle 인스턴스인 V 에 setLocation 메소드를 사용하여 위치를 증가해준다.

//x 좌표가 누적해서 덧셈해야 되므로 V.getLocation메소드의 x 좌표를 하여 더하기전 X 좌푤 구해온다.

//그 x 좌표에 Random 을 더해주고, y 좌표는 V.getLocation 메소드의 y 값을 가져와서 할당해준다.

if (V.getLocation().x > 1500) { //만약 더한 값이 1500보다 클 경우

Start = false; //while 문을 종료시키기 위해 start 는 false 로 바꾼다.

break; //for 문을 종료시키면, while 문으로 넘어가서 종료식 검사를 수행하게 된다

}

}

} while (Start); //do while 문을 사용하여 1회 이상 수행시킨다.

}

catch (Exception e) {

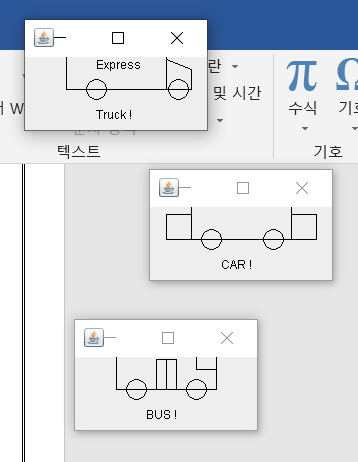
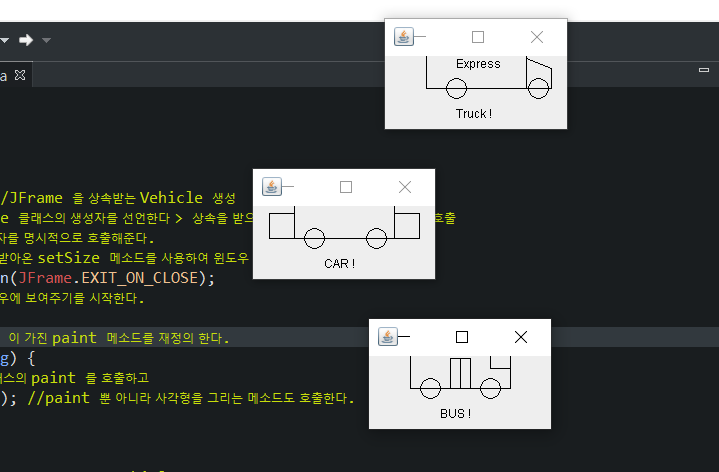
e.printStackTrace(); //오류가 날 경우, 그 오류가 무엇인지 콘솔에 찍어준다.

}

}

}

실행결과 ))



각각 상속받은 가장 큰 네모를 가지고 바퀴 등을 그려 달리기 게임을 수행한다.

X 픽셀이 1500이 넘어가면 레이스는 종료 된다.

**문제 2번**

1. 프로그램 소스코드 및 설명과 결과

import java.awt.Graphics;

import java.util.ArrayList;

import javax.swing.JFrame;

class Vehicle extends JFrame { //JFrame 을 상속받는 Vehicle 생성

public Vehicle() { //vehicle 클래스의 생성자를 선언한다 > 상속을 받으면 상속받은 클래스도 이 생성자 호출

super(); //JFrame 의 생성자를 명시적으로 호출해준다.

setSize(200, 120); //상속받아온 setSize 메소드를 사용하여 윈도우 크기 조절

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true); //윈도우에 보여주기를 시작한다.

}

@Override //부모 클래스인 Jframe 이 가진 paint 메소드를 재정의 한다.

public void paint(Graphics g) {

super.paint(g); //부모클래스의 paint 를 호출하고

g.drawRect(50, 20, 100, 50); //paint 뿐 아니라 사각형을 그리는 메소드도 호출한다.

}

}

class Bus extends Vehicle { //Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Bus 클래스 생성

@Override //Vehicle 클래스가 가진 paint 메소드를 재정의함을 표기

public void paint(Graphics g) {

super.paint(g); //부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawArc(90 + 30, 60, 20, 20, 0, 360); //인자로 받은 g 객체에 바퀴를 그려준다.

g.drawArc(30 + 30, 60, 20, 20, 0, 360); //인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그려준다.

g.drawRect(20 + 30 + 80, 30, 20, 20); //인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(60 + 30, 40, 20, 30);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(60 + 30, 40, 10, 30);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawString("BUS !", 50 + 30, 100);//인자로 받은 g 객체에 drawString 메소드를 사용하여 문자를 그려준다.

}

}

class Car extends Vehicle {//Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Car 클래스 생성

@Override/ / Vehicle 클래스가 가진 paint 메소드를 재정의함을 표기

public void paint(Graphics g) {//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

super.paint(g);//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawRect(150, 45, 25, 25);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawRect(25, 45, 25, 25);//인자로 받은 g 객체에 drawRect 메소드를 사용하여 사각형을그린다.

g.drawArc(60, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그린다.

g.drawArc(122, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 메소드를 사용하여 바퀴를 그린다.

g.drawString("CAR !", 50 + 30, 100); //인자로 받은 g 객체에 drawString 메소드를 사용하여 Car! 를 그려준다,

}

}

class Truck extends Vehicle {//Jframe 을 상속받는 Vehicle 클래스를 상속받아오는 Truck 클래스 생성

@Override

public void paint(Graphics g) {

super.paint(g);//부모클래스인 Vehicle 의 paint 메소드를 호출한다.

g.drawLine(120 + 30, 70, 145 + 30, 70); //인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawLine(145 + 30, 70, 145 + 30, 50);//인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawLine(145 + 30, 50, 120 + 30, 40);//인자로 받은 g 객체에 drawLine 을 사용하여 대각선을 그려준다,

g.drawArc(40 + 30, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 을 사용하여 바퀴를 그려준다,

g.drawArc(122 + 30, 60, 20, 20, 0, 360);//인자로 받은 g 객체에 drawArc 을 사용하여 바퀴를 그려준다,

g.drawString("Truck !", 50 + 30, 100); //인자로 받은 g 객체에 drawString 을 사용하여 Truck 을 그려준다.

g.drawString("Express", 50 + 30, 50);//인자로 받은 g 객체에 drawString 을 사용하여 Express 을 그려준다.

}

}

public class RacingCarFrame {

public static void main(String[] args) {

Vehicle truck = new Truck(); //truck 인스턴스에 Truck 객체를 할당

Vehicle car = new Car(); //car 인스턴스에 Car 객체를 할당

Vehicle bus = new Bus(); //bus 인스턴스에 Bus 객체릃 할당.

//3개 모두 Vehicle 을 상속받았으므로 Vehicle 타입의 인스턴스로 하면 된다.

Vehicle[] list = { truck,car,bus }; //Vehicle 인스턴스를 담는 list 를 생성, 인자는 truck car bus 인스턴스 들어감.

int y = 50; //프레임의 위치를 정해줄 y 좌표의 초기값 설정

for (Vehicle V : list) { //Vehicle 인스턴스인 V 변수에 list 각각 인덱스 할당 > for each 문의 특징이다.

V.setLocation(50, y); //출발점은 모두 같으므로 50 ,y 로 해준다.

y += 150; //y 좌표를 150 씩 증가하여 보기 좋게 만들어준다.

}

try { //thread 의 sleep 메소드가 작동하지 않을 경우를 위해 try catch 문 사용

boolean Start = true; //while 문 실행 유무를 위해 정해준다.

do {

Thread.sleep(100); //100 밀리세컨드 쉰다.

for (Vehicle V : list) {//Vehicle 인스턴스인 V 변수에 list 각각 인덱스 할당 > for each 문의 특징이다.

int Random = (int)(Math.random() \* 50) + 1; //Math 클래스의 random 메소드를 사용하여 1~51 까지 랜덤 숫자 생성

V.setLocation(V.getLocation().x + Random, V.getLocation().y);

//for 문을 돌리며 할당받은 Vehicle 인스턴스인 V 에 setLocation 메소드를 사용하여 위치를 증가해준다.

//x 좌표가 누적해서 덧셈해야 되므로 V.getLocation메소드의 x 좌표를 하여 더하기전 X 좌푤 구해온다.

//그 x 좌표에 Random 을 더해주고, y 좌표는 V.getLocation 메소드의 y 값을 가져와서 할당해준다.

if (V.getLocation().x > 1500) { //만약 더한 값이 1500보다 클 경우

Start = false; //while 문을 종료시키기 위해 start 는 false 로 바꾼다.

break; //for 문을 종료시키면, while 문으로 넘어가서 종료식 검사를 수행하게 된다

}

}

} while (Start); //do while 문을 사용하여 1회 이상 수행시킨다.

}

catch (Exception e) {

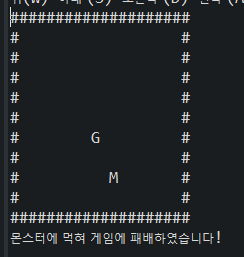
e.printStackTrace(); //오류가 날 경우, 그 오류가 무엇인지 콘솔에 찍어준다.

}

}

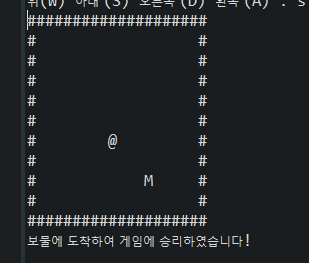
}

실행결과 ))



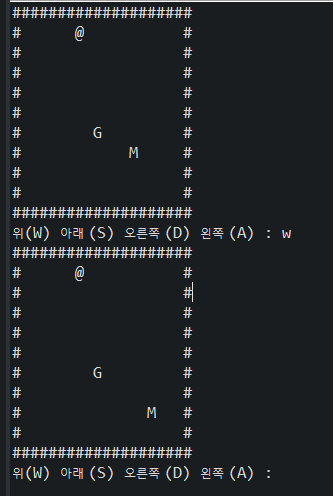
몬스터에게 먹혀 게임에 패배한 경우, 유저 모양인 @는 안보이고 G와 M 만 남는다.

게임에 패배하였다고 출력되고 게임이 종료된다.

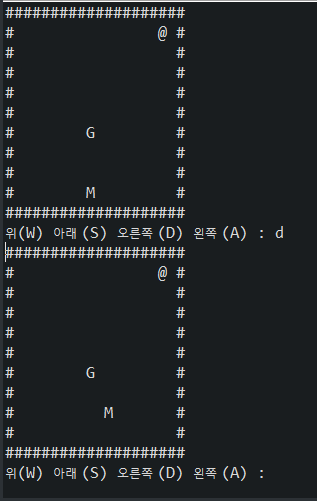


보물에 도착한 경우, 보물 모양인 G 는 안보이고 @ 와 M 만 남는다.

게임에 승리하였다 출력되고 프로그램이 종료된다.



배열의 최상단에서 w 를 눌러도 오류 보정을 했기 떄문에, 더 이상 위로 올라가진 않는다.



오른쪽으로 이동하는것도 마찬가지로 오류 보정을 했으므로, 더 이상 오른쪽으로 이동하지 않는다.

왼쪽도 마찬가지다.