1.开始太阳系模拟（必须要了解是简便，开头做一点复杂的东西，后来就会简便了）

1.1》开始做一个父类，将来直接继承，画画就行了

package 飞机;

/\*\*

\* 下面这个程序，可以使我们将来可以省去很多的东西，

\*/

public class ChangYongFuLei extends Frame{

public static final int Game\_Width = 500;

public static final int Game\_Height = 500;

public void chuang(){ 这个将来会再次在里面调用的

setSize(Game\_Width,Game\_Height);

setLocation(100,100);

setVisible(true);

new huahua().start();

addWindowListener(new WindowAdapter(){

public void windowClosing(WindowEvent e){

System.exit(0);}

}); }

//这个将来用来画画的内容

public void paint(Graphics g){}

//是画面动起来

//设置 sleep的时间

long sleep\_time = 40;

class huahua extends Thread{

public void run(){

while(true){

repaint();

try{

Thread.sleep(sleep\_time);

}catch(InterruptedException e){

e.printStackTrace();

}

}

}

}

}

2.这里需要 我么试验一下我们建好的父类

**public** **class** Haha **extends** ChangYongFuLei{

**public** **void** paint(Graphics g){

g.drawString("我叫张宇晋",100,100);

g.drawLine(100, 100, 200, 200);

}

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

**new** Haha().chuang();

}

}

3.现在我们建立一个图片的父类

**public** **class** HuaTu {

Image img;

**double** x,y;

//将来调用的时候 直接 draw（g）

**public** **void** draw(Graphics g){

g.drawImage(img,(**int**)x,(**int**)y,**null**);

}

下面这个事初始化

**public** HuaTu(String path,**double** x,**double** y){

**this**.img = JiazaiTupian.*getmage*(path);

**this**.x = x;

**this**.y = y;

}

}

4.我们看看如何调用上面那个图片

HuaTu s = **new** HuaTu("images/123.jpg",400,400);

上面的是初始化 图片的路径和 位置

**public** **void** paint(Graphics g){

s.draw(g); 使用里面的一个函数开始画图

}

5.》》如果我们要画出两个图片 ，那么上面的父类也就不好使了，因为，如果调用上面的父类的时候，会初始化数据，即使我们提供的初始化数据是不一样的，但是，在调用画画的过程中都是 new HuaTu(数据).draw(g)虽然是可以画出的，但是到了后来不利于我们进行管理，所以我们现在要将图片也制作成一个父类

**public** **class** XingXin **extends** HuaTu{

主要是下面的那个super 它会向父类 请求初始化东西，并且在调用画画过程中的变量也会不一样，使得我们的程序可以维持

**public** XingXin(String path,**double** x,**double** y){

**super**(path,x,y); //将来画图那个程序会 在父类中找的

}

}

6.》要知道子类拥有构造器，那么父类也必须有构造器

**public** **class** XingXin **extends** HuaTu{

Image xinimg;**double** xinx,xiny;**double** speed;

HuaTu center; //这里是代表的中心的星球（HuaTu中代表的是一个星球）

**double** degree;

这里写下下面的构造器时候，父类中也应该有一个构造器（即使是空的）

**public** XingXin(HuaTu center,String xinpath,**double** xinx,**double** xiny,**double** speed){

//以下的属性都是开始时的属性

**this**.center = center; //这个球体是在 绕着 传进来的中心走的

**this**.x = center.x + xiny; // 中心 球的x 加上长轴的距离

**this**.y = center.y ;

//下面这个是传入图片

**this**.xinimg = JiazaiTupian.*getmage*(xinpath);

**this**.xinx = xinx; //长轴和短轴 开始时候传入的是啥就是啥

**this**.xiny = xiny;

**this**.speed = speed;

}

//因为每次行星的坐标都要发生变化，所以我们需要重写draw（）；方法

**public** **void** draw(Graphics g){

//沿着椭圆轨迹飞行

g.drawImage(xinimg,(**int**)x,(**int**)y,**null**);

//现在开始制作飞行了

x = center.x + 5 + xinx \* Math.*cos*(degree);

y = center.y + 5 + xiny \* Math.*sin*(degree);

degree += speed;

}

7.这个是完整的

package 太阳系;

import java.awt.Graphics;

import java.awt.Image;

import 飞机.JiazaiTupian;

public class XingXin extends HuaTu{

Image xinimg;double xinx,xiny;double speed;HuaTu center; //这里是代表的中心的星球（HuaTu中代表的是一个星球）

double degree;

public XingXin(HuaTu center,String xinpath,double xinx,double xiny,double speed){

//以下的属性都是开始时的属性

this.center = center; //这个球体是在 绕着 传进来的中心走的

this.x = center.x + xiny; // 中心 球的x 加上长轴的距离

this.y = center.y ;

//下面这个是传入图片

this.xinimg = JiazaiTupian.getmage(xinpath);

this.xinx = xinx; //长轴和短轴 开始时候传入的是啥就是啥

this.xiny = xiny;

this.speed = speed;

}

//因为每次行星的坐标都要发生变化，所以我们需要重写draw（）；方法

public void draw(Graphics g){

//沿着椭圆轨迹飞行

g.drawImage(xinimg,(int)x,(int)y,null);

//现在开始制作飞行了  **下面这个5代表的是图片的一半高和宽，为了让他绕中心转**

x = center.x + 5 + xinx \* Math.cos(degree);

y = center.y + 5 + xiny \* Math.sin(degree);

degree += speed;

}

public XingXin(String path,double x,double y){  **其实这个没有多大用**

super(path,x,y); //将来画图那个程序会 在父类中找的

}

}

8.有时候我们需要图片的宽度和高度，多以学习他的怎么用也是非常重要的

new Star.img.getWidth(**null)**