**package** renderer;  
  
**import** elements.Camera;  
**import** primitives.Color;  
**import** primitives.Point3D;  
**import** primitives.Ray;  
**import** scene.Scene;  
  
**import** java.util.List;  
**import** java.util.MissingResourceException;  
  
*/\*\*  
 \* this class create the image's color matrix from the scene  
 \*/***public class** Render {  
 **private** ImageWriter **\_imageWriter**;  
 **private** Camera **\_camera**;  
 **private** RayTracerBase **\_rayTracer**;  
  
 *//chaining methods* **public** Render setImageWriter(ImageWriter imageWriter) {  
 **\_imageWriter** = imageWriter;  
 **return this**;  
 }  
  
  
 **public** Render setCamera(Camera camera) {  
 **\_camera** = camera;  
 **return this**;  
 }  
  
 **public** Render setRayTracer(RayTracerBase rayTracer) {  
 **\_rayTracer** = rayTracer;  
 **return this**;  
 }  
  
 */\*\*  
 \* this function calculate the color of each pixel, and color it  
 \*/* **public void** renderImage() {  
 *//check that all the field is not null* **if** (**\_imageWriter** == **null** || **\_camera** == **null** || **\_rayTracer** == **null**) {  
 **throw new** MissingResourceException(**"one or more field in render is null"**, **"render"**, **""**);  
 }  
 *//calculate and paint all the pixels* **int** nX = **\_imageWriter**.getNx();  
 **int** nY = **\_imageWriter**.getNy();  
 **for** (**int** i = 0; i < nY; i++) {  
 **for** (**int** j = 0; j < nX; j++) {  
 **if**(i==100&&j==100) {  
 System.***out***.println(i + **"-"** + j);  
 }  
 System.***out***.println(i + **"-"** + j);  
 Ray ray = **\_camera**.constructRayThroughPixel(nX, nY, j, i);  
 Color color = **\_rayTracer**.traceRay(ray);  
 **\_imageWriter**.writePixel(j, i, color);  
 }  
 }  
 }  
  
 */\*\*  
 \* print grid on the picture  
 \*  
 \** ***@param interval*** *distance between the lines grid  
 \** ***@param intervalColor*** *the color of the grid  
 \*/* **public void** printGrid(**int** interval, Color intervalColor) {  
 **if** (**\_imageWriter** == **null**) {  
 **throw new** MissingResourceException(**"\_imageWriter is null"**, **"render"**, **""**);  
 }  
 **int** nX = **\_imageWriter**.getNx();  
 **int** nY = **\_imageWriter**.getNy();  
 **for** (**int** i = 0; i < nY; i++) {  
 **for** (**int** j = 0; j < nX; j++) {  
 **if** (i % interval == 0 || j % interval == 0) {  
 **\_imageWriter**.writePixel(j, i, intervalColor);  
 }  
 }  
 }  
  
 }  
  
 */\*\*  
 \* this methode implement the low of Demeter  
 \*/* **public void** writeToImage() {  
 **if** (**\_imageWriter** == **null**) {  
 **throw new** MissingResourceException(**"\_imageWriter is null"**, **"render"**, **""**);  
 }  
 **\_imageWriter**.writeToImage();  
 }  
}