מחלקות החבילה Geometries

1. מבנה המחלקה Geometry:

```
private Material _material = new Material();
private double _nShininess = 1;
private Color _emmission = new Color(0, 0, 0); צבע הטבעי של אותו גאומטריה
```

```
public abstract List<Point3D> FindIntersections (Ray ray);
public abstract Vector getNormal(Point3D point);

public double   getShininess();
public Material getMaterial();
public Color getEmmission();

public void setShininess(double n);
public void setMaterial(Material _material);
public void setEmmission(Color emmission);

public void setKs(double ks);
public void setKd(double kd);
public void setKr(double Kr);
public void setKt(double Kt);
```

2. מבנה המחלקה FlatGeometry:

```
public interface FlatGeometry { } // Marker interface
```

3. מבנה המחלקה RadialGeometry:

```
protected double _radius;

public RadialGeometry();
public RadialGeometry(double radius);

public double getRadius();
public void   setRadius(double radius);
```

4. מבנה המחלקה Plane:

```
private Vector _normal;
private Point3D _Q;

// ***************** Constructors ********************** //
public Plane();
public Plane (Plane plane);
public Plane (Vector normal, Point3D q);

// ***************** Getters/Setters ********************** //
public Vector getNormal(Point3D point);
public Point3D getQ();

public void setNormal(Vector normal);
public void setQ(Point3D d);

// ***************** Operations ******************** //
public List<Point3D> FindIntersections(Ray ray);
```

5. מבנה המחלקה Triangle:

```
private Point3D _p1;
private Point3D _p2;
private Point3D _p3;

// ***************** Constructors ********************** //
public Triangle();
public Triangle(Triangle triangle);
public Triangle(Point3D p1, Point3D p2, Point3D p3);
public Triangle(Map<String, String> attributes);

// ***************** Getters/Setters ********************** //
public Point3D getP1();
public Point3D getP2();
public Point3D getP3();

public void setP1(Point3D p1);
public void setP2(Point3D p2);
public void setP3(Point3D p3);

// ***************** Operations ******************** //
public Vector getNormal(Point3D point);
public List<Point3D> FindIntersections(Ray ray);
```

6. מבנה המחלקה Cylinder:

```
private Point3D _axisPoint;
private Vector _axisDirection;

// ***************** Constructors ********************** //
public Cylinder();
public Cylinder(Cylinder cylinder);
public Cylinder(double radius, Point3D axisPoint, Vector axisDirection);

// ***************** Getters/Setters ********************** //
public Point3D getAxisPoint();
public Vector getAxisDirection();

public void setAxisPoint(Point3D axisPoint);
public void setAxisDirection(Vector axisDirection);

// ***************** Operations ******************** //
public List<Point3D> FindIntersections(Ray ray);
public Vector getNormal(Point3D point);
```

7. מבנה המחלקה Sphere:

```
private Point3D _center;

// ***************** Constructors ********************** //
public Sphere();
public Sphere (Sphere sphere);
public Sphere(double radius, Point3D center);
public Sphere(Map<String, String> attributes);

// ***************** Getters/Setters ********************** //
public Point3D getCenter();
public void setCenter(Point3D center);

// ***************** Operations ******************** //
public List<Point3D> FindIntersections(Ray ray);
public Vector getNormal(Point3D point);
```