מחלקות החבילה primitives

1. מבנה המחלקה Coordinate:

```
private double _coordinate;

// ***************** Constructors ********************** //
public Coordinate();
public Coordinate(double coordinate);
public Coordinate(Coordinate coordinate)

// ***************** Getters/Setters ********************** //
public double getCoordinate();
public void setCoordinate(double coordinate)

// ***************** Administration  ******************** //
public int compareTo(Coordinate coordinate);

// ***************** Operations ******************** //
public void add (Coordinate coordinate);
public void subtract (Coordinate coordinate);
```

2. מבנה המחלקה Point2D:

```
protected Coordinate _x;
protected Coordinate _y;

// ***************** Constructors ********************** //
public Point2D();
public Point2D(Coordinate x, Coordinate y);
public Point2D(Point2D point2D);

// ***************** Getters/Setters ********************** //
public Coordinate getX();
public Coordinate getY();
public void setX(Coordinate _x);
public void setY(Coordinate _y);

// ***************** Administration  ******************** //
public int compareTo(Point2D point2D);
```

3. מבנה המחלקה Point3D:

```
private Coordinate _z;

// ***************** Constructors ********************** //
public Point3D();
public Point3D(Coordinate x, Coordinate y, Coordinate z);
public Point3D(double x, double y, double z);
public Point3D(Point3D point3D);

// ***************** Getters/Setters ********************** //
public Coordinate getZ();
public void setZ(Coordinate _z);

// ***************** Administration  ******************** //
public int compareTo(Point3D point3D);
public String toString();

// ***************** Operations ******************** //
public void add(Vector vector);
public void subtract(Vector vector);
public double distance(Point3D point);
```

4. מבנה המחלקה Vector:

```
private Point3D _head;

// ***************** Constructors ********************** //
public Vector();
public Vector(Point3D head);
public Vector(Vector vector);
public Vector(double xHead,  double yHead, double zHead);
public Vector(Point3D p1, Point3D p2);

// ***************** Getters/Setters ********************** //
public Point3D getHead()
public void    setHead(Point3D head);

// ***************** Administration  ******************** //
public int compareTo(Vector vector);
public String toString();

// ***************** Operations ******************** //
public void add (Vector vector );
public void subtract (Vector vector);
public void scale(double scalingFactor);
public Vector crossProduct(Vector vector);
public double length();
public void normalize(); // Throws exception if length = 0
public double dotProduct(Vector vector);
```

5. מבנה המחלקה Ray:

```
// Point of origin
private Point3D _POO;

// Ray direction
private Vector _direction;

// ***************** Constructors ********************** //
public Ray();
public Ray(Ray ray);
public Ray(Point3D poo, Vector direction);

// ***************** Getters/Setters ********************** //
public void setPOO(Point3D _POO);
public void setDirection(Vector _direction);
public Vector  getDirection();
public Point3D getPOO();
```

6. מבנה המחלקה Material:

```
private double _Kd; // Diffusion attenuation coefficient
private double _Ks; // Specular attenuation coefficient
private double _Kr; // Reflection coefficient (1 for mirror)
private double _Kt; // Refraction coefficient (1 for transparent)
private double _n;  // Refraction index

// ***************** Constructors ********************** //
public Material()
 {
          _Kd = 1;
          _Ks = 1;
          _Kr = 0;
          _Kt = 0;
          _n  = 1;
}
public Material(Material material);

// ***************** Getters/Setters ********************** //
public double getKd();
public double getKs();
public double getKr();
public double getKt();
public double getN();

public void setKd(double _Kd);
public void setKs(double _Ks);
public void setKr(double _Kr);
public void setKt(double _Kt);
public void setN (double _n);
```