Čas 5.

Paketi

geometry\Point.java

```java
package geometry;

public class Point {
// Coordinates of the point
  private double x;
  private double y;

// Create a point from its coordinates
  public Point(double xVal, double yVal) { x = xVal;  y = yVal; }

// Create a Point from an existing Point object
  public Point(final Point aPoint) {  x = aPoint.x; y = aPoint.y; }

// Move a point
  public void move(double xDelta, double yDelta) { x += xDelta;   y += yDelta; }

// Calculate the distance to another point
  public double distance(final Point aPoint) {
    return Math.sqrt((x - aPoint.x)*(x - aPoint.x) + (y - aPoint.y)*(y - aPoint.y) );
  }

// Convert a point to a string
  public String toString() {return Double.toString(x) + ", " + y;} // As "x, y"

// Retrieve the x coordinate  - ACCESSOR METHOD
        public double getX() { return x;  }

// Retrieve the y coordinate  - ACCESSOR METHOD
  public double getY() { return y;  }

// Set the x coordinate    - MUTATOR METHOD
  public void setX(double inputX)  { x = inputX;  }

// Set the y coordinate    - MUTATOR METHOD
  public void setY(double inputY)  { y = inputY;  }
}
```

geometry\Line.java

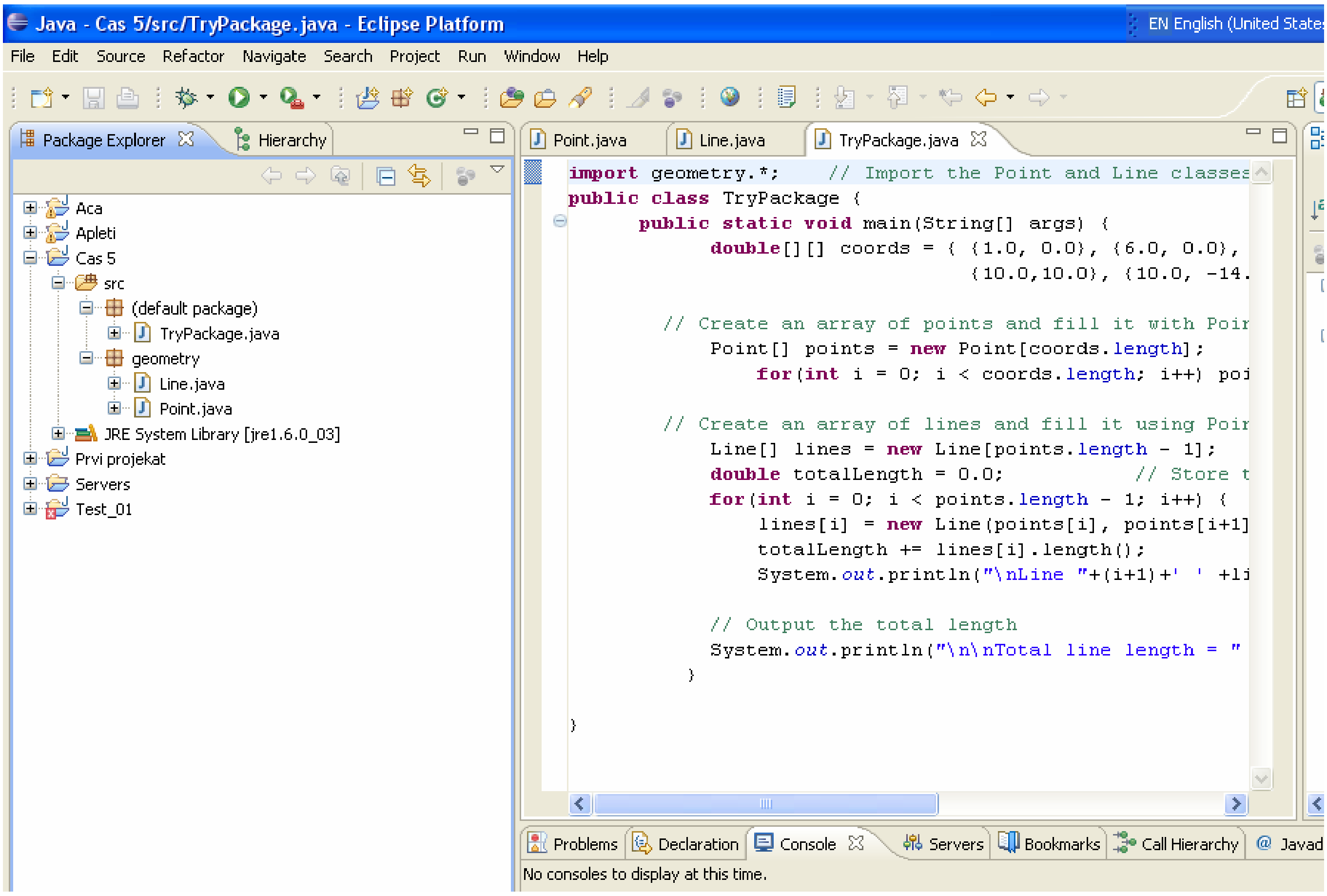
```java
package geometry;

public class Line {
// Data members
  Point start;     // Start point of line
  Point end;       // End point of line

// Create a line from two points
      public Line(final Point start, final Point end)   {
    this.start = new Point(start);      this.end = new Point(end);  }

// Create a line from two coordinate pairs
public Line(double xStart, double yStart, double xEnd, double yEnd)  {
   start = new Point(xStart, yStart);       // Create the start point
         end = new Point(xEnd, yEnd);      // Create the end point
}
// Calculate the length of a line
public double length()  { return start.distance(end); }

// Return a point as the intersection of two lines -- called from a Line object
public Point intersects(final Line line1){
    Point localPoint = new Point(0, 0);
    double num =
     (this.end.getY() - this.start.getY())*(this.start.getX() - line1.start.getX()) -
     (this.end.getX() - this.start.getX())*(this.start.getY() - line1.start.getY());

    double denom =
      (this.end.getY() - this.start.getY())*(line1.end.getX() - line1.start.getX()) -
      (this.end.getX() - this.start.getX())*(line1.end.getY() - line1.start.getY());

localPoint.setX(line1.start.getX() + (line1.end.getX() - line1.start.getX())*num/denom);
localPoint.setY(line1.start.getY() + (line1.end.getY() - line1.start.getY())*num/denom);

return localPoint;
}

// Convert a line to a string
  public String toString()   {
             return "(" + start+ "):(" + end + ")";    // As "(start):(end)"
      }    // that is, "(x1, y1):(x2, y2)"
}
```

TryPackage.java

```
import geometry.*;    // Import the Point and Line classes

public class TryPackage {
  public static void main(String[] args) {
    double[][] coords = { {1.0, 0.0}, {6.0, 0.0}, {6.0, 10.0},
                          {10.0,10.0}, {10.0, -14.0}, {8.0, -14.0}};

// Create an array of points and fill it with Point objects
  Point[] points = new Point[coords.length];
  for(int i = 0; i < coords.length; i++) points[i] = new Point(coords[i][0],coords[i][1]);

// Create an array of lines and fill it using Point pairs
    Line[] lines = new Line[points.length - 1];
    double totalLength = 0.0;           // Store total line length here
    for(int i = 0; i < points.length - 1; i++) {
      lines[i] = new Line(points[i], points[i+1]); // Create a Line
      totalLength += lines[i].length();            // Add its length
      System.out.println("\nLine "+(i+1)+' '+lines[i]+"  Length is "+lines[i].length()); }

// Output the total length
    System.out.println("\n\nTotal line length = " + totalLength);
  }
}
```

POKRETANJE

Eclipse



```
import geometry.*;     // Import the Point and Line classes
public class TryPackage {
      public static void main(String[] args) {
            double[][] coords = { {1.0, 0.0}, {6.0, 0.0},
                                  {10.0,10.0}, {10.0, -14.

        // Create an array of points and fill it with Poir
            Point[] points = new Point[coords.length];
                for(int i = 0; i < coords.length; i++) poi

        // Create an array of lines and fill it using Poir
            Line[] lines = new Line[points.length - 1];
            double totalLength = 0.0;          // Store t
            for(int i = 0; i < points.length - 1; i++) {
                lines[i] = new Line(points[i], points[i+1]
                totalLength += lines[i].length();
                System.out.println("\nLine "+(i+1)+' ' +li

            // Output the total length
            System.out.println("\n\nTotal line length = "
          }

}
```

No consoles to display at this time.

Komandna linija

`Point.java` i `Line.java` moraju biti smešteni u direktorijumu `geometry`. Neka je direktorijum `geometry` smešten u direktorijumu `d:\mojipaketi`. i neka se `TryPackage.java` nalazi u `D:\moji programi`, tada je kompajliranje moguće izvesti ovako:

```
      D:\moji programi\> javac -classpath “D:\mojipaketi” TryPackage.java
```

a pokretanje ovako:

```
      D:\moji programi> java -classpath “.;D:\mojipaketi” TryPackage
```

Neke od grešaka:

```
D:\moji programi >javac TryPackage.java
TryPackage.java:1: package geometry does not exist
import geometry.*;     // Import the Point and Line classes
^
TryPackage.java:8: cannot find symbol
symbol  : class Point
location: class TryPackage
                    Point[] points = new Point[coords.length];
                    ^
TryPackage.java:8: cannot find symbol
symbol  : class Point
location: class TryPackage
```

```
                    Point[] points = new Point[coords.length];
                                        ^
...
8 errors

D:\moji programi\>java TryPackage
Exception in thread "main" java.lang.NoClassDefFoundError: geometry/Point
        at TryPackage.main(TryPackage.java:8)
Caused by: java.lang.ClassNotFoundException: geometry.Point
        at java.net.URLClassLoader$1.run(URLClassLoader.java:200)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:188)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:307)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:252)
        at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:320)
        ... 1 more

D:\moji programi>java -classpath "D:\mojipaketi" TryPackage
Exception in thread "main" java.lang.NoClassDefFoundError: TryPackage
Caused by: java.lang.ClassNotFoundException: TryPackage
...
```

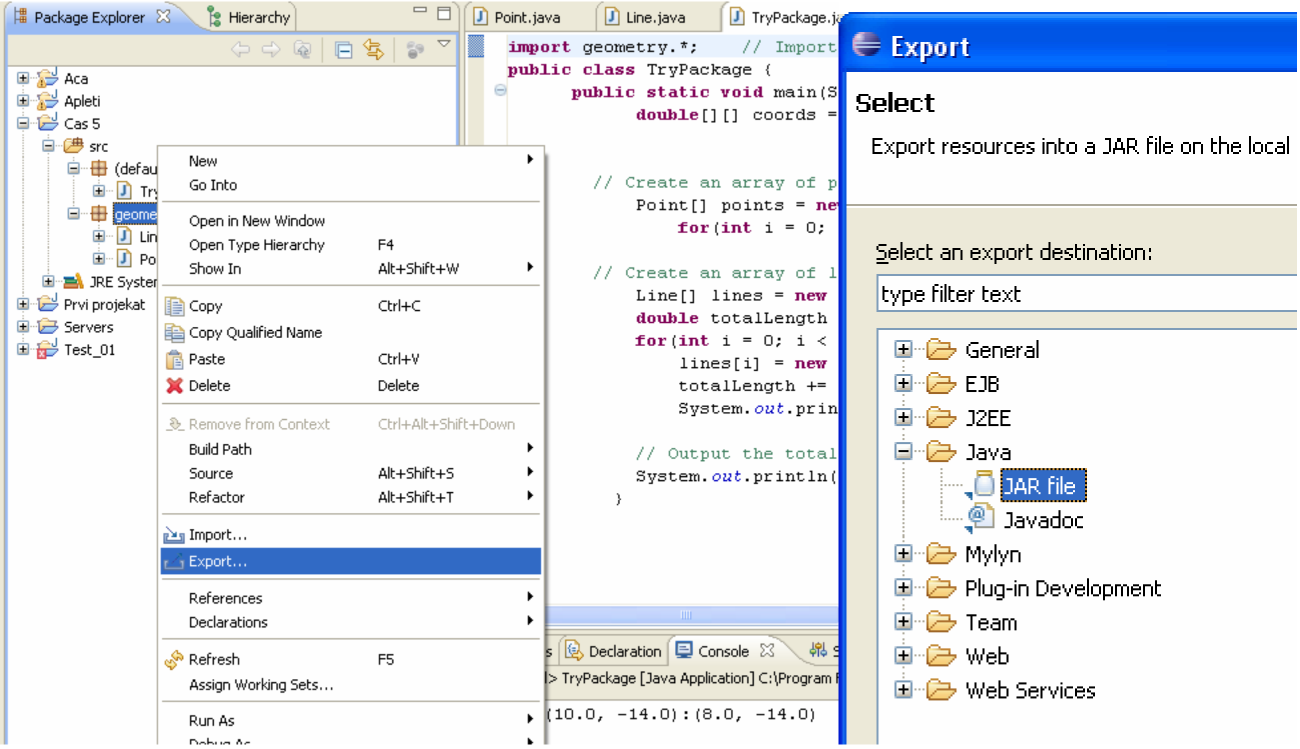
KREIRANJE jar arhiva

Komandna linija

```
D:\moji programi>javac -classpath "D:\moji jar fajlovi\geometry.jar" TryPackage.java
```

Eclipse

napomena: Prikazani su samo screenshot-ovi koraka u kreiranju .jar fajla u kojima je bilo nekih podešavanja.



```
import geometry.*;    // Import
public class TryPackage {
    public static void main(S
        double[][] coords =

    // Create an array of p
        Point[] points = ne
        for(int i = 0;

    // Create an array of l
        Line[] lines = new
        double totalLength
        for(int i = 0; i <
            lines[i] = new
            totalLength +=
            System.out.prin

        // Output the total
        System.out.println(
    }
```

```
> TryPackage [Java Application] C:\Program F
(10.0, -14.0):(8.0, -14.0)
```

Export

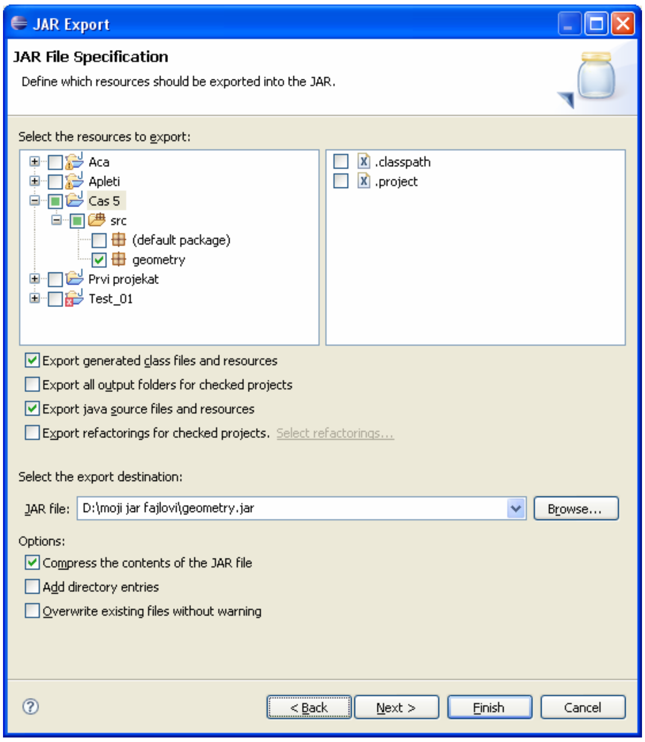
Select

Export resources into a JAR file on the local

Select an export destination:

type filter text

- General
- EJB
- J2EE
- Java
  - JAR file
  - Javadoc
- Mylyn
- Plug-in Development
- Team
- Web
- Web Services



JAR Export

JAR File Specification

Define which resources should be exported into the JAR.

Select the resources to export:

- □ Aca
- □ Apleti
- ▨ Cas 5
  - ▨ src
    - □ (default package)
    - ☑ geometry
- □ Prvi projekat
- □ Test_01

□ .classpath
□ .project

☑ Export generated class files and resources
□ Export all output folders for checked projects
☑ Export java source files and resources
□ Export refactorings for checked projects. Select refactorings...

Select the export destination:

JAR file: D:\moji jar fajlovi\geometry.jar

Options:

☑ Compress the contents of the JAR file
□ Add directory entries
□ Overwrite existing files without warning

UPOTREBA jar arhiva

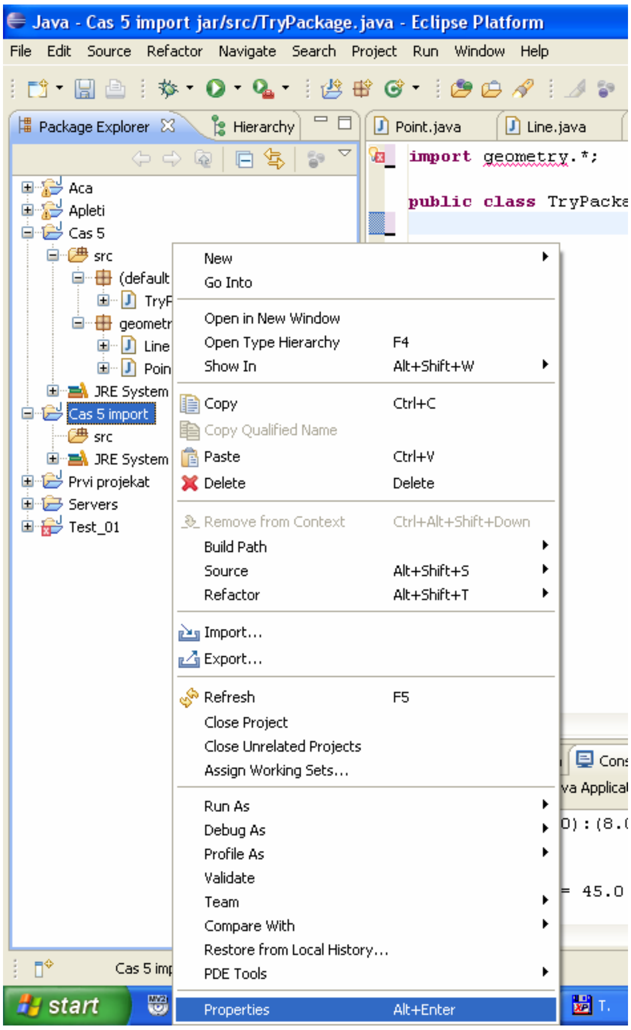
Komandna linija

```
D:\moji programi>java -classpath ".;D:\moji jar fajlovi\geometry.jar" TryPackage
```

Eclipse

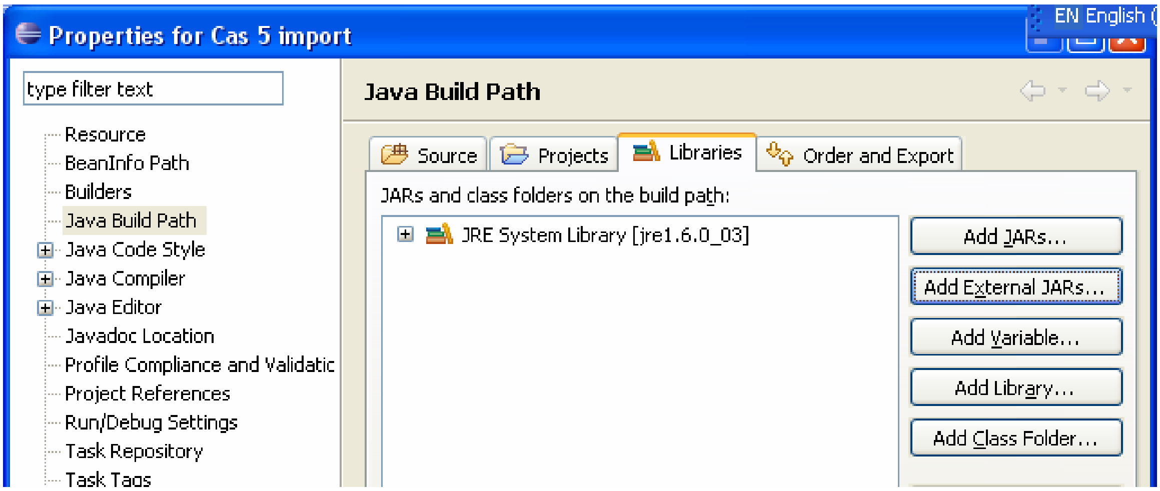
napomena: Prikazani su samo screenshot-ovi koraka u podešavanju za korišćenje .jar fajla u kojima je bilo nekih podešavanja.

Za ovaj primer je napravljen novi projekat sa TryPackage klasom



Java - Cas 5 import jar/src/TryPackage.java - Eclipse Platform

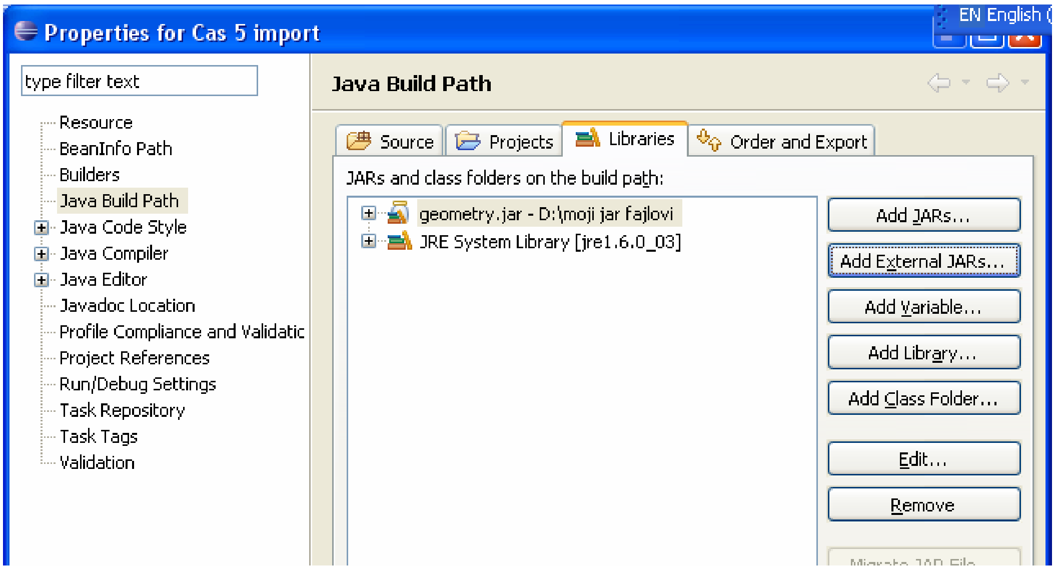
```
import geometry.*;

public class TryPacka
```



JARs and class folders on the build path:

JRE System Library [jre1.6.0_03]

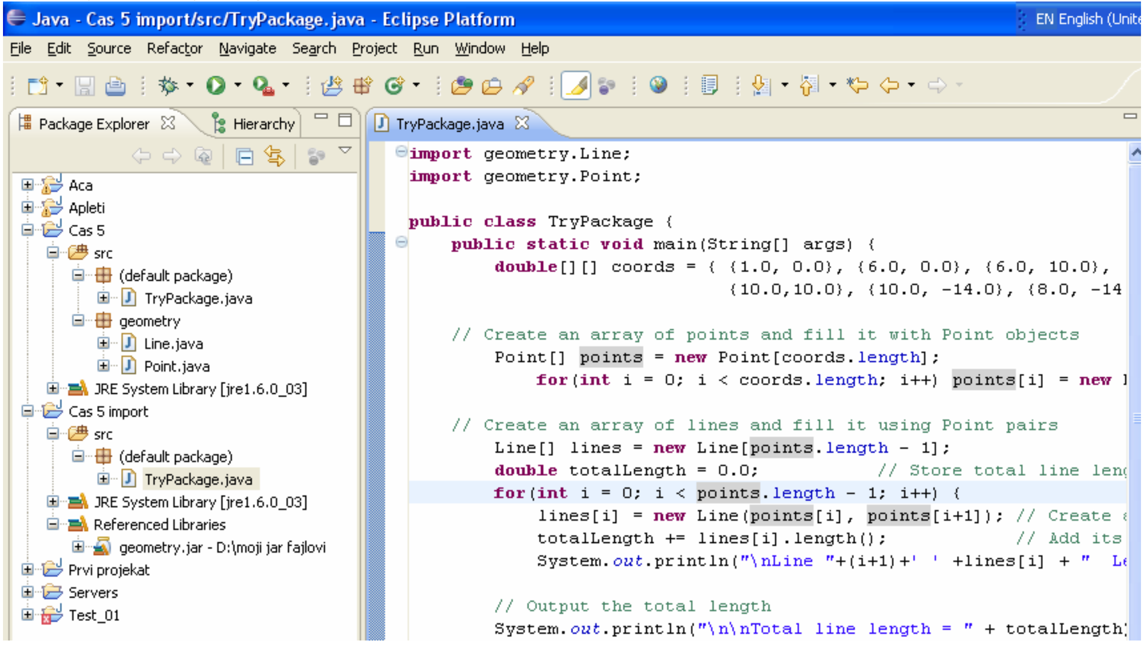
Add External JARs



JARs and class folders on the build path:

geometry.jar - D:\moji jar fajlovi
JRE System Library [jre1.6.0_03]

Na kraju sitacija izgleda ovako



```
import geometry.Line;
import geometry.Point;

public class TryPackage {
    public static void main(String[] args) {
        double[][] coords = { {1.0, 0.0}, {6.0, 0.0}, {6.0, 10.0},
                              {10.0,10.0}, {10.0, -14.0}, {8.0, -14

    // Create an array of points and fill it with Point objects
        Point[] points = new Point[coords.length];
            for(int i = 0; i < coords.length; i++) points[i] = new I

    // Create an array of lines and fill it using Point pairs
        Line[] lines = new Line[points.length - 1];
        double totalLength = 0.0;              // Store total line leng
        for(int i = 0; i < points.length - 1; i++) {
            lines[i] = new Line(points[i], points[i+1]); // Create a
            totalLength += lines[i].length();            // Add its
            System.out.println("\nLine "+(i+1)+' ' +lines[i] + "  Le

        // Output the total length
        System.out.println("\n\nTotal line length = " + totalLength)
```

i run :)