

Aufgabe 3

a)

```
class Account {
    // Attributes:
    private String owner;
    // Balance is given in cent as an integer value to prevent
    // floating-point math rounding errors.
    private long balance;

    // Constructor, taking initial deposit into account:
    public Account( String owner, long balance ) {
        this.owner = owner;
        this.balance = balance;
    }

    public long getBalance() {
        return( balance );
    }

    // Add/Remove money from this account.
    public void deposit( long amount ) {
        this.balance = this.balance + amount;
    }
    public void withdraw( long amount ) {
        this.balance = this.balance - amount;
    }
}
```

b)

```
class Transfer {
    // Attributes:
    private Account source;
    private Account destination;
    private long amount;

    // Constructor:
    public Transfer( Account source, Account destination, long amount )
    {
        this.source = source;
        this.destination = destination;
        this.amount = amount;
    }

    // Withdraw / deposit the proper amount from the Accounts involved.
    public void execute() {
        source.withdraw( amount );
        destination.deposit( amount );
    }
}
```

Aufgabe 4

a,b,c,d,e,f)

```

class Triangle {
    // Corner vertices
    Point p1;
    Point p2;
    Point p3;

    // Constructors
    Triangle( Point p1, Point p2, Point p3 ) {
        this.p1 = p1;
        this.p2 = p2;
        this.p3 = p3;
    }

    // Shift the triangle by dx, dy.
    public void shift( float dx, float dy ) {
        this.p1.shift( dx, dy );
        this.p2.shift( dx, dy );
        this.p2.shift( dx, dy );
    }

    // Move triangle in space.
    public void shift( Point dv ) {
        this.p1.shift( dv );
        this.p2.shift( dv );
        this.p3.shift( dv );
    }

    // Rotate triangle around the origin.
    public void rotate( double angle ) {
        this.p1.rotate( angle );
        this.p2.rotate( angle );
        this.p3.rotate( angle );
    }

    // Rotate triangle around a Point.
    public void rotate( Point center, double angle ) {
        this.p1.rotate( center, angle );
        this.p2.rotate( center, angle );
        this.p3.rotate( center, angle );
    }

    // Helper methods: Length of sides.
    private double side_a() {
        return( Math.sqrt(
            Math.pow( this.p1.x - this.p2.x, 2.0 ) +
            Math.pow( this.p1.y - this.p2.y, 2.0 )
        ) );
    }

    private double side_b() {
        return( Math.sqrt(
            Math.pow( this.p2.x - this.p3.x, 2.0 ) +
            Math.pow( this.p2.y - this.p3.y, 2.0 )
        ) );
    }
}

```

```

        ) );
    }

    private double side_c() {
        return( Math.sqrt(
            Math.pow( this.p3.x - this.p1.x, 2.0 ) +
            Math.pow( this.p3.y - this.p1.y, 2.0 )
        ) );
    }

    // Calculate this Triangles circumference.
    public double circumference() {
        return( this.side_a() + this.side_b() + this.side_c() );
    }

    // Calculate the area using Herons formula.
    public double area() {
        double s = this.circumference() / 2.0;
        return( Math.sqrt(
            s *
            (s - this.side_a()) *
            (s - this.side_b()) *
            (s - this.side_c())
        ) );
    }

    // Calculate the centroid of this triangle.
    public Point centroid() {
        return( new Point(
            (this.p1.x + this.p2.x + this.p3.x) / 3.0,
            (this.p1.y + this.p2.y + this.p3.y) / 3.0
        ) );
    }
}

```

g)

Main-Methode:

```

class TestApp {
    public static void main( String args[] ) {
        double readPx;
        double readPy;
        Point p1, p2, p3;
        double centerX;
        double centerY;
        double angle;
        Point centroid;

        // Read in coordinates for three points.
        readPx = Terminal.askDouble( "Point 1 X: " );
        readPy = Terminal.askDouble( "Point 1 Y: " );
        p1 = new Point( readPx, readPy );

        readPx = Terminal.askDouble( "Point 2 X: " );
        readPy = Terminal.askDouble( "Point 2 Y: " );
        p2 = new Point( readPx, readPy );

        readPx = Terminal.askDouble( "Point 3 X: " );

```

```
readPy = Terminal.askDouble( "Point 3 Y: " );
p3 = new Point( readPx, readPy );

// Make a triangle.
Triangle tri = new Triangle( p1, p2, p3 );
centerX = Terminal.askDouble( "Rotation center X: " );
centerY = Terminal.askDouble( "Rotation center Y: " );
angle = Terminal.askDouble( "Rotation angle: " );

// Rotate around a Point - this uses shifting
// and rotating around the center.
tri.rotate( new Point( centerX, centerY ), angle );

// Calculate centroid for printing out.
centroid = tri.centroid();

// Print out the new triangle values.
Terminal.print( "Circumference: " );
Terminal.println( tri.circumference() );
Terminal.print( "Area: " );
Terminal.println( tri.area() );
Terminal.print( "Centroid X, Y: " );
Terminal.print( centroid.x );
Terminal.print( ", " );
Terminal.println( centroid.y );
}
```

Ausgabe:

```
Point 1 X: 0
Point 1 Y: 0
Point 2 X: 3
Point 2 Y: 3
Point 3 X: 3
Point 3 Y: 0
Rotation center X: 2
Rotation center Y: 2
Rotation angle: 70
Circumference: 10.242640687119284
Area: 4.499999999999999
Centroid X, Y: 2.9396926207859084, 1.6579798566743313
```