1.4 Extra exercises 1

Design and implement a new class for two dimensional `Triangle` objects similar to the manner in which the `Rectangle` class was discussed in the lecture.

Step 1. Specify the class name, the attributes and two constructors for the `Triangle` class using an UML class diagram. The two constructors are

- the default constructor, which can be used to create default triangles with the three points A (0.0, 0.0), B (0.0, 1.0) and C (1.0, 0.0).
- the so-called parameter or general constructor with six parameters to initialize the three points of a triangle object.

Implement the class for triangle objects in Java based on your specified UML class diagram.

Step 2. Extend the UML class diagram and the Java class for triangles with a `print` method to display relevant attributes and values of a triangle (i.e. coordinates of vertices, length of sides, etc.). You can expand your `print` method depending on what other methods you have programmed.

Step 3. Extend the UML class diagram and the Java class for triangles with a `draw` method which will draw a triangle object in a given color using a viewer object. You can use the `Line` class in the `view3D` package (`inf.v3d.obj.Line`). Alternatively, use the `Polyline` class.

Step 4. Extend the UML class diagram and the Java class for triangles by a method to calculate the area of a given triangle object. The value of the area is to be returned by the method.

Step 5. Extend the UML class diagram and the Java class for triangles by a method to test if a given point (x, y) is inside a triangle object.

Step 6. Extend the UML class diagram and the Java class for triangles by a method to draw the circumcircle of a certain triangle object based using a viewer object. Use the class `Circle` of the `view3D` package to draw the circle.

Step 7. Extend the UML class diagram and the Java class for triangles by three methods to calculate each of the three angles (α, β, γ) in a given triangle. Each method should return one corresponding angle α, β or γ.

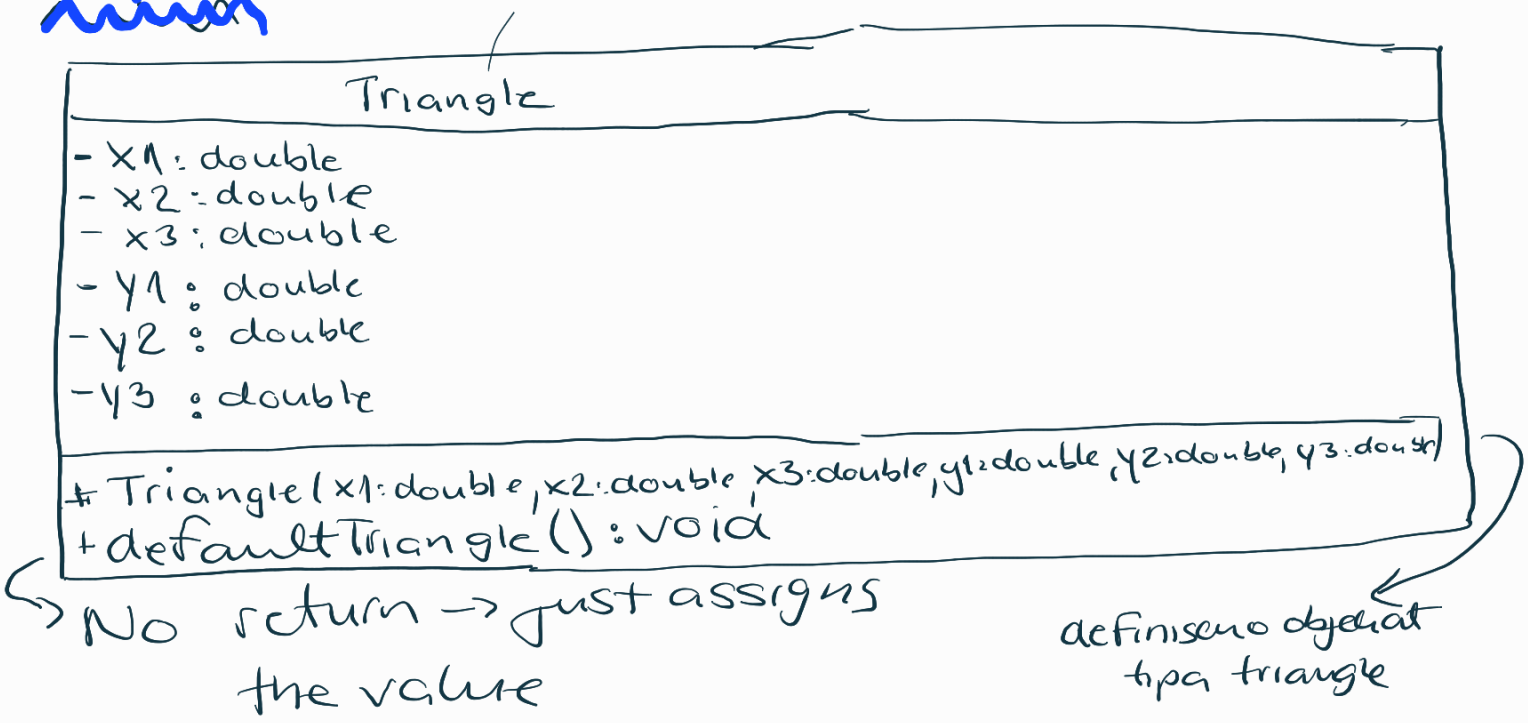
Step 8. For the ambitious students: Solve the problems given above for triangle in *three dimensional space*, i.e. for triangles where the vertices A, B and C are defined by three cooordinates (x, y, z) and z is not always 0.

Hint: Use resources such as Wikipedia to get background information on how to compute the area, the angles and the circumcircle of a triangle. Make sure you understand the underlying mathematics before you start to program.

Test all implemented constructors and methods using simple test programs. Check to see if your results are correct by inspection (graphic output) or by calculation.

UML class diagram

Step 1



Java code implementation

```
public class Triangle {
    private double x1;
    private double x2;
    private double x3;
    private double y1;
    private double y2;
    private double y3;
```

Definisali smo atribute

// Sada kreiramo objekat, to jeste pravimo constructor za objekte tipa Triangle

parametri

```
public Triangle (double x1, double x2, double x3
                 double y1, double y2, double y3
{
```

atributi

```
this.x1 = x1;
this.x2 = x2;
this.x3 = x3;
this.y1 = y1;
this.y2 = y2;
this.y3 = y3; }
```

Metoda konstruktor, parametrima novog Triangle koji kreiramo dodjeli vrijednost i tako ga kreira

// Sada kreiramo metodu koja objektu tipa triangle dodjeli default vrijednost

void → ne daje izlaz, samo vrši dodjeljivanje vrijednosti

```
void defaultTriangle() {
    this.x1 = 0.0;
    this.x2 = 0.0;
    this.x3 = 1.0;
    this.y1 = 0.0;
    this.y2 = 1.0;
    this.y3 = 0.0; }
```

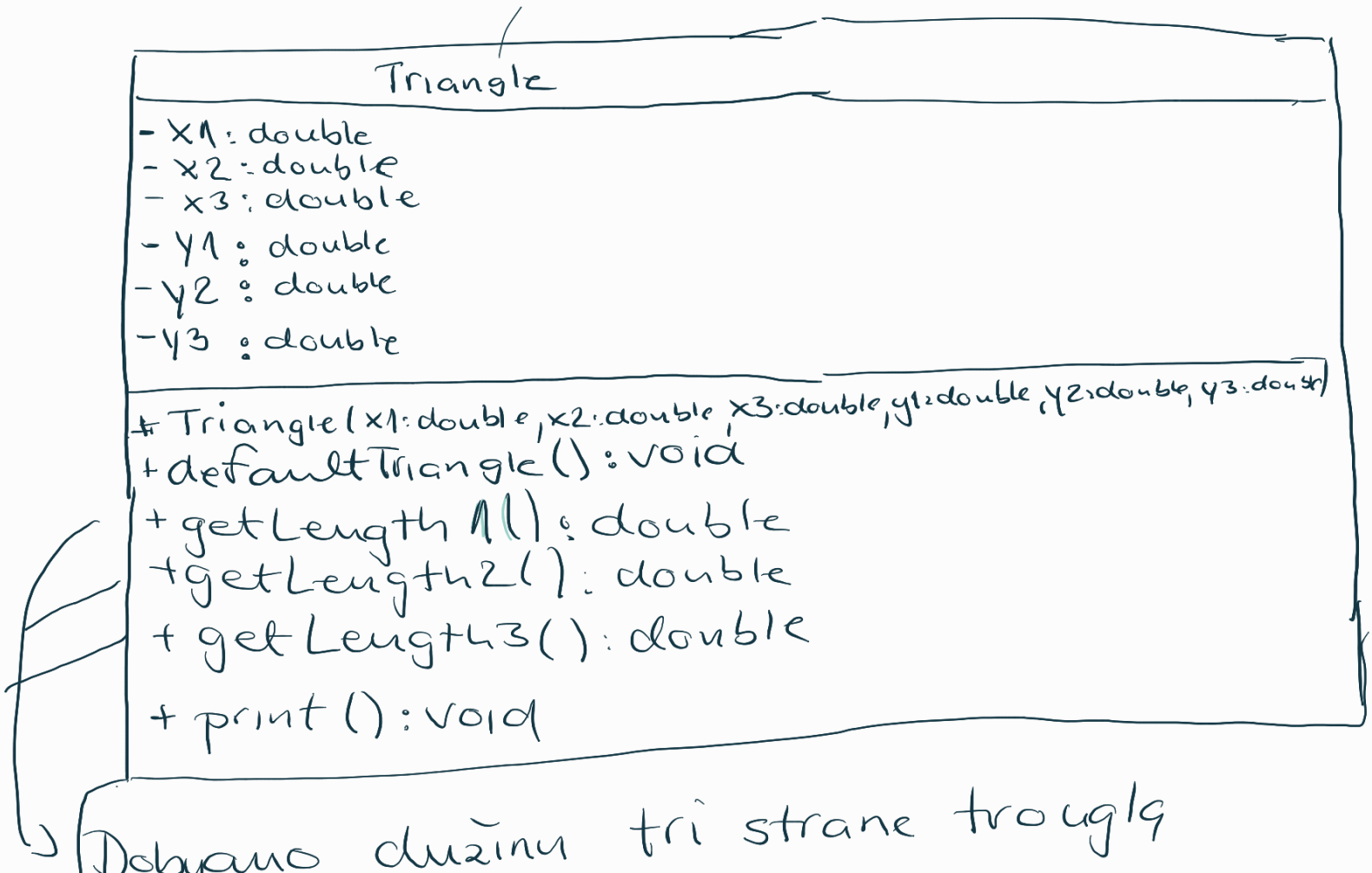
kada se metoda poziva npr.

```
Z1.defaultTriangle
```

↳ x1, x2, x3... ovog objekta su oni na koje se "this." odnosi i njima se dodjele vrijednosti

Step 2 - To introduce the print method that would display relevant attributes for a triangle

We will create methods to calculate the length of sides, the area
=> We expand the UML diagram



Dobijamo dužinu tri strane trougla

↳ printa pravougaonika, dužina svtranica između pojedinih tačaka !!!!

→ tip izlaza → ime metode

```
public double getLength1()
double e1 = Math.sqrt(Math.pow(this.x1 - this.x2,2)
return e1;                    - Math.pow(this.y1 - this.y2,2));
}
```

→ tip izlaza → ime metode

```
public double getLength2()
double e2 = Math.sqrt(Math.pow(this.x1 - this.x3,2)
return e2;                    - Math.pow(this.y1 - this.y3,2));
}
```

→ tip izlaza → ime metode

```
public double getLength2()
double e3 = Math.sqrt(Math.pow(this.x2 - this.x3,2)
return    ;                   - Math.pow(this.y2 - this.y3,2));
}
```

// Ako želimo da nam se svaki put pri pozivanju metode print ispiše i veličina stranica, moramo unutar print metode svih pozivanja getLength metodu

// Sada kreiramo metodu za ispis trougla
arg
ispis
→ nema rezultata, samo se izvrši ispis

odnosi se na objekt trougao npr. z1.print().

```
void print() {
double e1 = this.getLength1();
double e2 = this.getLength2();
double e3 = this.getLength3();

System.out.println("Triangle has the following points:");
System.out.println("A(x1,y1):(" + this.x1 + "," + this.y1 + ")");
System.out.println("A(x2,y2):(" + this.x2 + "," + this.y2 + ")");
System.out.println("A(x3,y3):(" + this.x3 + "," + this.y3 + ")");
System.out.println("The length of the triangle between points A and B is:" + e1);
System.out.println("The length of the triangle between points A and C is:" + e2);
System.out.println("The length of the triangle between points B and C is:" + e3);
}
```

Step 3 - Extend the UML class diagram with "draw" method which will draw a triangle object in a given color using a viewer object

- Koristit ću Polyline class (nosuće i sa Line classom iz NEW 3D paketa)

Triangle
- x1 : double - x2 : double - x3 : double - y1 : double - y2 : double - y3 : double
+ Triangle(x1:double, x2:double, x3:double, y1:double, y2:double, y3:double) + defaultTriangle() : void + getLength1() : double + getLength2() : double + getLength3() : double + print() : void + draw(color : String, V : Viewer)

pozivaju objekt

ako nema 12 i 29 mora si staviti void !!!

↳ String je boja

↳ Viewer (mora se importovati library)

Moramo importovati da bi uopšte ovaj program radio

```
import info.v3d.obj.UnstructuredMesh;
import inf.v3d.view.*;
```

Java code:

nema izlaza

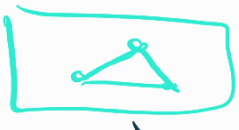
mesh će biti nas trougaoni

```
void draw(String color, Viewer v) {
    UnstructuredMesh um = new UnstructuredMesh();
    int t1 = um.addPoint(this.x1, this.y1, 0);
    int t2 = um.addPoint(this.x2, this.y2, 0);
    int t3 = um.addPoint(this.x3, this.y3, 0);
```

↳ dodajemo tačke trougla u mrežu

```
    um.addCell(t1, t2, t3);
```

↳ dodajemo ćeliju sastavljenu od tih tačaka (u suštini povezujemo tačke)



```
    um.setColor(color);
    um.setOutlineVisible(true);
```

↳ menjamo boju i animo granice vidljivim

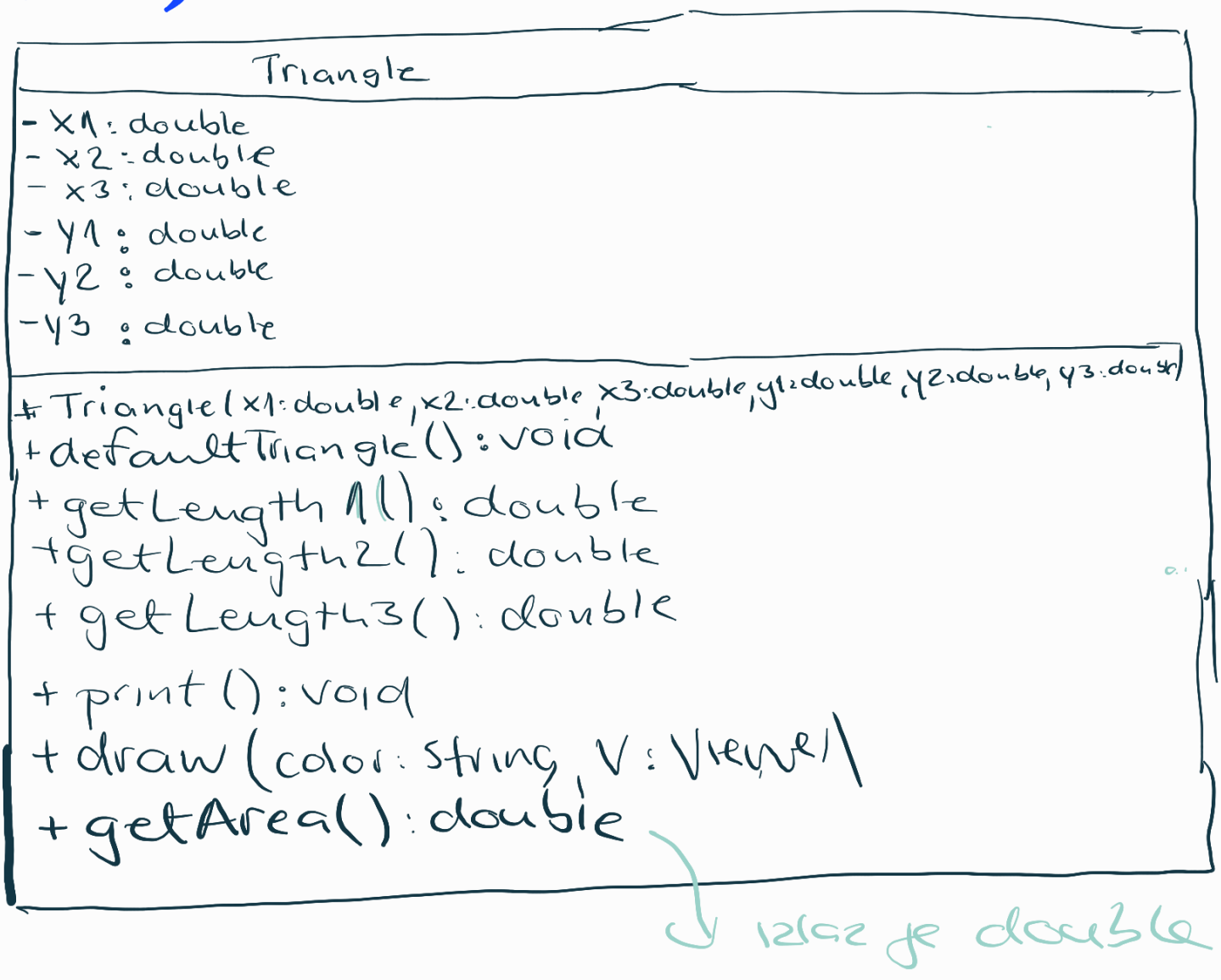
```
    V.addObject3D(um);
    V.setVisible(true);
}
```

→ napravljenu mrežu ubacujemo u viewer

↳ ako primetimo da se Viewer prosleđuje kao put kada pozivamo draw metodu

↳ Viewer može biti u potpunosti definisan unutar klase ?? → pitati asistenticu

Step 4 - method to calculate the area (method returns the area)



To calculate the area of a triangle when given only the coordinates we use the following formula:

$$A = \frac{1}{2}\left| x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_3) \right|$$

Java code:

```
public double getArea() {
 double a=0.5*Math.abs(this.x1(this.y2-
   this.y3)+this.x2(this.y3-this.y1)+
   this.x3(this.y1-this.y2));
 return a; }
```

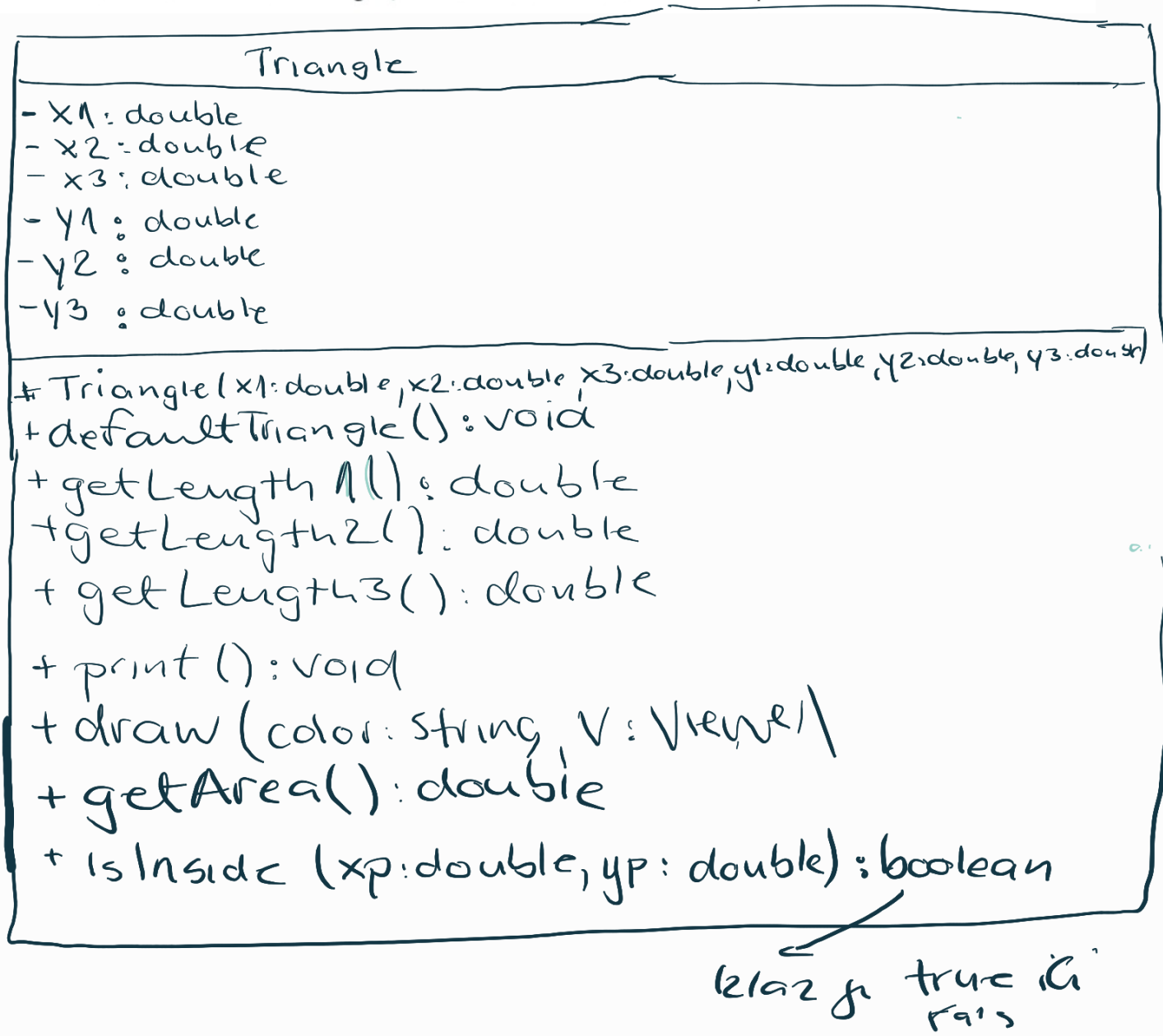
Step 5 – test if a point is inside the triangle

To create this method, we use the following logic

Solution:

Let the coordinates of three corners be (x1, y1), (x2, y2) and (x3, y3). And coordinates of the given point P be (x, y)

1. Calculate area of the given triangle, i.e., area of the triangle ABC in the above diagram.
   Area A = [x1 (y2 – y3) + x2 (y3 – y1) + x3 (y1 - y2)] / 2
2. Calculate area of the triangle PAB. We can use the same formula for this. Let this area be A1.
3. Calculate area of the triangle PBC. Let this area be A2.
4. Calculate area of the triangle PAC. Let this area be A3.
5. If P lies inside the triangle, then A1 + A2 + A3 must be equal to A.



Java code:

tip izlaza — ulazni parametri

```
public boolean getArea(double xp, double yp){
```

računamo površinu glavnog trougla — odnosi se na → t1.getArea()

```
double a = this.getArea();  {

Triangle t1=new (this.x1,this.x2,xp,this.y1,this.y2,yp);
Triangle t2=new (this.x1,this.x3,xp,this.y1,this.y3,yp);
Triangle t3=new (this.x3,this.x2,xp,this.y3,this.y2,yp);

double a1 = t1.getArea();
double a2 = t2.getArea();
double a3 = t3.getArea();

boolean res;

if (a==a1+a2+a3) {
    res = true;
    } else {
        res = false; }
        return res; }
```

Step 6 – Draw the circumcircle of a certain triangle

We will use the Circle class from View3d library

Polupreciza $R = \frac{abc}{4S}$ ← parsing — Bfraue

The Cartesian coordinates of the circumcenter $U = (U_x, U_y)$ are

$$U_x = \frac{1}{D}\left[(A_x^2 + A_y^2)(B_y - C_y) + (B_x^2 + B_y^2)(C_y - A_y) + (C_x^2 + C_y^2)(A_y - B_y)\right]$$

$$U_y = \frac{1}{D}\left[(A_x^2 + A_y^2)(C_x - B_x) + (B_x^2 + B_y^2)(A_x - C_x) + (C_x^2 + C_y^2)(B_x - A_x)\right]$$

with

$$D = 2\left[A_x(B_y - C_y) + B_x(C_y - A_y) + C_x(A_y - B_y)\right].$$

Triangle

- x1 : double
- x2 : double
- x3 : double
- y1 : double
- y2 : double
- y3 : double

+ Triangle(x1:double, x2:double, x3:double, y1:double, y2:double, y3:double)
+ defaultTriangle() : void
+ getLength1() : double
+ getLength2() : double
+ getLength3() : double
+ print() : void
+ draw(color:String, V:Viewer) : void
+ getArea() : double
+ isInside(xp:double, yp:double) : boolean
+ circumcircle(Viewer v) : void

Java code:

```
//Odredjivanje circumcirle-a
void circumcircle (Viewer v) {
double D= 2*(this.x1*(this.y2-this.y3)+this.x2*(this.y3-this.y1)+this.x3*(this.y1-this.y2));
double mx= ((Math.pow(x1,2)+Math.pow(y1,2))*(this.y2-this.y3)
            + ((Math.pow(x2,2)+Math.pow(y2,2))* (this.y3-this.y1))
            +  ((Math.pow(x3,2)+Math.pow(y3,2))* (this.y1-this.y2)))/D;
double my= ((Math.pow(x1,2)+Math.pow(y1,2))*(this.x3-this.x2)
        + ((Math.pow(x2,2)+Math.pow(y2,2))* (this.x1-this.x3))
        +  ((Math.pow(x3,2)+Math.pow(y3,2))* (this.x2-this.x1)))/D;

double a=this.getLength1();
double b=this.getLength2();
double c=this.getLength3();

double p=this.getArea();
double r= (a*b*c)/(4*p);
Circle c1 = new Circle (r);
c1.setCenter(mx, my, 0);
c1.setRadius(r);
v.addObject3D(c1);
v.setVisible(true); }
```

Step 7 - three methods to calculate angles of a triangle

To calculate the angles of a triangle, we use the following formulas

$$A = \arccos\left(\frac{b^2+c^2-a^2}{2bc}\right)$$

$$B = \arccos\left(\frac{a^2+c^2-b^2}{2ac}\right)$$

$$C = \arccos\left(\frac{a^2+b^2-c^2}{2ab}\right)$$

Triangle

- x1 : double
- x2 : double
- x3 : double
- y1 : double
- y2 : double
- y3 : double

+ Triangle(x1:double, x2:double, x3:double, y1:double, y2:double, y3:double)
+ defaultTriangle() : void
+ getLength1() : double
+ getLength2() : double
+ getLength3() : double
+ print() : void
+ draw(color:String, V:Viewer1) : void
+ getArea() : double
+ isInside(xp:double, yp:double) : boolean
+ circumcircle(Viewer v) : void
+ Alpha() : double
+ Beta() : double
+ Gamma() : double

Java code :

```
//Odredjivanje jednog ugla - alpha
public double Alpha () {
double alpha;
double a= this.getLength1();
double b=this.getLength2();
double c= this.getLength3();

alpha=Math.acos((Math.pow(b, 2)+ Math.pow(c, 2)-Math.pow(a, 2))/(2*b*c));
return alpha;
}


//Odredjivanje jednog ugla - beta
public double Beta () {
double beta;
double a= this.getLength1();
double b=this.getLength2();
double c= this.getLength3();

beta=Math.acos((Math.pow(a, 2)+ Math.pow(c, 2)-Math.pow(b, 2))/(2*a*c));
return beta;
}
```

```
//Odredjivanje jednog ugla - gama
public double Ghama () {
double ghama;
double a= this.getLength1();
double b=this.getLength2();
double c= this.getLength3();

ghama=Math.acos((Math.pow(a, 2)+ Math.pow(b, 2)-Math.pow(c, 2))/(2*b*a));
return ghama;
}
```

Test case - In the end we write a test case in which we invoke all these methods (We can write this however we want)

```
    //Test case
    public static void main (String[] args) {

        Triangle z1 = new Triangle(7, 12, 4.0, 6,8.0,11.0);
        z1.print();
        Viewer v = new Viewer ();
        z1.circumcircle(v);
        z1.draw("blue", v);
        double xp=1;
        double yp=4;
        boolean b = z1.isInside(xp, yp);
        double a=z1.Alpha();
        double be=z1.Beta();
        double g=z1.Ghama();
        System.out.println("\n"+"The angle alpha is " + a + "\n");
        System.out.println("The angle beta is " + be + "\n");
        System.out.println("The angle ghama is " + g + "\n");
        System.out.println("Is the point in the triangle ?  " + b + "\n");

    }
}
```