```
1       public Triangle(int s1, int s2, int s3){
          side1 = s1;
          side2 = s2;
          side3 = s3;
        }

1       System.out.println("Type: " + triangle.classify());

                        public String classify(){
                2         if (isImpossible()) {

                                                public boolean isImposs
                                        2         if (side1 <= 0 || side2 <
                                                  {
                                        3          return true;
                                                  }

                                        4         if (!(side1>side3-side2
                                        5          return true;
                                                  }
                                        6         return false;

                                                }
                        7          return P_IMPOSSIBLE;
                                  }
                        8         if (side1 == side2) {
                        9          if (side2 == side3)
                                   {
                       10           return P_EQUILATERAL;
                                   }
                       11          else{
                       11           return P_ISOSCELES;
                                   }
                                  }
                       12         if (isRightAngled()){

                                                12       public boolean isRightA
                                                  int[] sides = new int[] { s
                                                  Arrays.sort(sides); //as
                                                  return sides[2]
                                                   == Math.sqrt((long) sid
                                                 }
```

```
                    13          return P_RIGHTANGLED;
                                }
                    14          return P_SCALENE;

14      System.out.println("Triangle sides: " + triangle.getSideLengths());

        public String getSideLengths(){
          return side1 + "," + side2 + "," + side3;
         }

        System.out.println("Area: " + triangle.getArea());

        public double getArea(){
 2        if (!isImpossible()){

                                public boolean isImpossible() {
                    2             if (side1 <= 0 || side2 <= 0 || side3 <= 0)
                                  {
                    3             return true;
                                  }

                    4             if (!(side1>side3-side2 && side1>side2-side3 &
                    5              return true;
                                  }
                    6             return false;
                                 }

15          return Math.sqrt(getPerimeter()
            / 2
            * (getPerimeter() / 2 - side1)
            * (getPerimeter() / 2 - side2)
            * (getPerimeter() / 2 - side3));
           }
16         return -1;
          }

17      System.out.println("Perimeter: " + triangle.getPerimeter());

        public int getPerimeter(){
17        if (! isImpossible()){

                                public boolean isImpossible() {
                    2             if (side1 <= 0 || side2 <= 0 || side3 <= 0)
                                  {
```

```
3           return true;
           }

4          if (!(side1>side3-side2 && side1>side2-side3 &
5           return true;
           }
6          return false;
          }

18          return side1 + side2 + side3;
           }
19         return -1;
          }
```

```
ible() {
= 0 || side3 <= 0)
```

```
 && side1>side2-side3 && side2>side1-side3)){
```

```
ngled(){
side1, side2, side3 };
cending array

des[0] * sides[0] + (long) sides[1] * sides[1]);
```

& side2>side1-side3)){

& side2>side1-side3)){