Tutorial static method and args in main

> Based on the tutorial of "2020F-Java-A" designed by teaching group in SUSTech
>
> Modified (mainly change to markdown file) by ZHU Yueming in 2021. March. 22th
>
> Add before exercise by ZHU Yueming in 2021. Oct. 18th

Experimental Objective

1. Learn how to use static method.
2. Learn method overloading.
3. Learn invoking methods with array arguments and getting back the values.

Part 1. Static Method

Exercise 1:

Create a class named `MyTriangle` as below:

```
public class MyTriangle {
    public static double perimeter(double a, double b, double c) {
        return a + b + c;
    }
    public static boolean isValid(double side1, double side2, double side3) {
      //todo: complete method
        return false
    }
    public static double area(double a, double b, double c) {
       //todo:
    }
    public static double area(double bottom, double height) {
        //todo:
    }
    public static double area(double a, double b, int angleOfAandB) {
        //todo:
    }
}
```

Q1: Complete isValid method

Complete the method `isValid`, return true only if the three sides are positive and the sum of any two sides is larger then the third side.

Q2. Complete area method

In the `MyTriangle` class, complete three methods name `area`

```
public static double area(double a, double b, double c)
public static double area(double bottom, double height)
public static double area(double a, double b, int angleOfAandB)
```

There are three ways to compute the area:

1) compute area by three sides a,b,c. p= (a+b+c)/2, area =√p(p-a)(p-b)(p-c)

- In this case, it should check the isValid first then do the area calculation.

2) compute area by bottom and height: area = 1/2 * bottom * height

3) compute area by two sides a, b and the angle between the two sides(`angleOfAandB`)

area = 1/2 * a * b * sin(angleOfAandB)

Then test it and made it works.

Sample input and output:

```
public static void main(String[] args) {
        System.out.println(isValid(3.5, 4, -1));
        System.out.println(isValid(3, 4, 5));
        System.out.println(isValid(1, 2, 2));
        System.out.println(area(0, 4.5, 5.5));
        System.out.println(area(3.5, 4.5, 5.5));
        System.out.println(area(5, 6, 90));
        System.out.println(area(3,6));
    }
```

Result:

```
false
true
true
-1.0
7.854885024620029
15.0
9.0
```

Exercise 2: ScoreAnalyzer

Assume there is a set of students' exam scores, and the system needs to perform a series of operations on these scores:

Q1:

1. **Check Valid Score**

   ```
   public static boolean isValid(double score)
   ```

   The range of the valid score range is between 1 and 100. Any score outside this range is considered invalid.

2. **Filter invalid Score**

   ```
   public static double[] filterInvalidScores(double[] scores)
   ```

   The method should accept an array as input and return another array containing only valid scores. Inside this method, the `isValid` method needs to be called.

3. **Find the max score**

   ```
   public static double findMaxScore(double[] scores)
   ```

   The input array may contain invalid scores. We need to find the highest score among all and return it.

4. **Find the min score**

   ```
   public static double findMinScore(double[] scores)
   ```

   The input array may contain invalid scores. We need to find the lowest score among all and return it.

5. **Calculate Average Score**

   ```
   public static double calculateFinalAverage(double[] scores)
   ```

   The input parameters may contain invalid scores. We need to remove the highest and lowest scores from the valid ones, then calculate the average of the remaining scores.

   This method must call the `filterInvalidScores` function.

Sample test code:

```
 public static void main(String[] args) {
        double[] scores = {-5, 89.5, 67.3, 22.4, 67.8, 90.2, 75.5, -1, 99};
        double[] validScores = filterInvalidScores(scores);
        System.out.println(Arrays.toString(validScores));
        System.out.println("max in valid scores = " +
findMaxScore(validScores));
        System.out.println("min in valid scores = " +
findMinScore(validScores));
        System.out.println("max in all scores = " + findMaxScore(scores));
        System.out.println("min in all scores = " + findMinScore(scores));
        System.out.println("Average = "+ calculateFinalAverage(scores));
    }
```

Result:

```
[89.5, 67.3, 22.4, 67.8, 90.2, 75.5, 99.0]
max in valid scores = 99.0
min in valid scores = 22.4
max in all scores = 99.0
min in all scores = -5.0
Average = 78.06
```

Q2:

Assume that, the range of valid score can be difined by teacher. In this method,additional two parameters, `min` and `max`, are passed, representing the minimum and maximum range of valid scores.

1. **Check Valid Score**

   ```
   public static boolean isValid(double score, int min, int max)
   ```

Sample test code:

```
public static void main(String[] args) {
        System.out.println(isValid(90));
        System.out.println(isValid(110));
        System.out.println(isValid(90,10,80));
        System.out.println(isValid(105,0,120));
    }
```

result:

```
true
false
false
true
```

Part 2. Passing arguments from main method

Example Exercise:

According to the `MyTriangle` class we introduce before, try the following main method:

```
public class CircleTestArgs {
    public static void main(String[] args) {
        double side1 = Double.parseDouble(args[0]);
        double side2 = Double.parseDouble(args[1]);
        double side3 = Double.parseDouble(args[2]);
        System.out.println("Area = " + MyTriangle.area(side1, side2, side3));
        System.out.println("Perimeter = " + MyTriangle.perimeter(side1, side2,
side3));
    }
}
```

By Intellij IDEA:

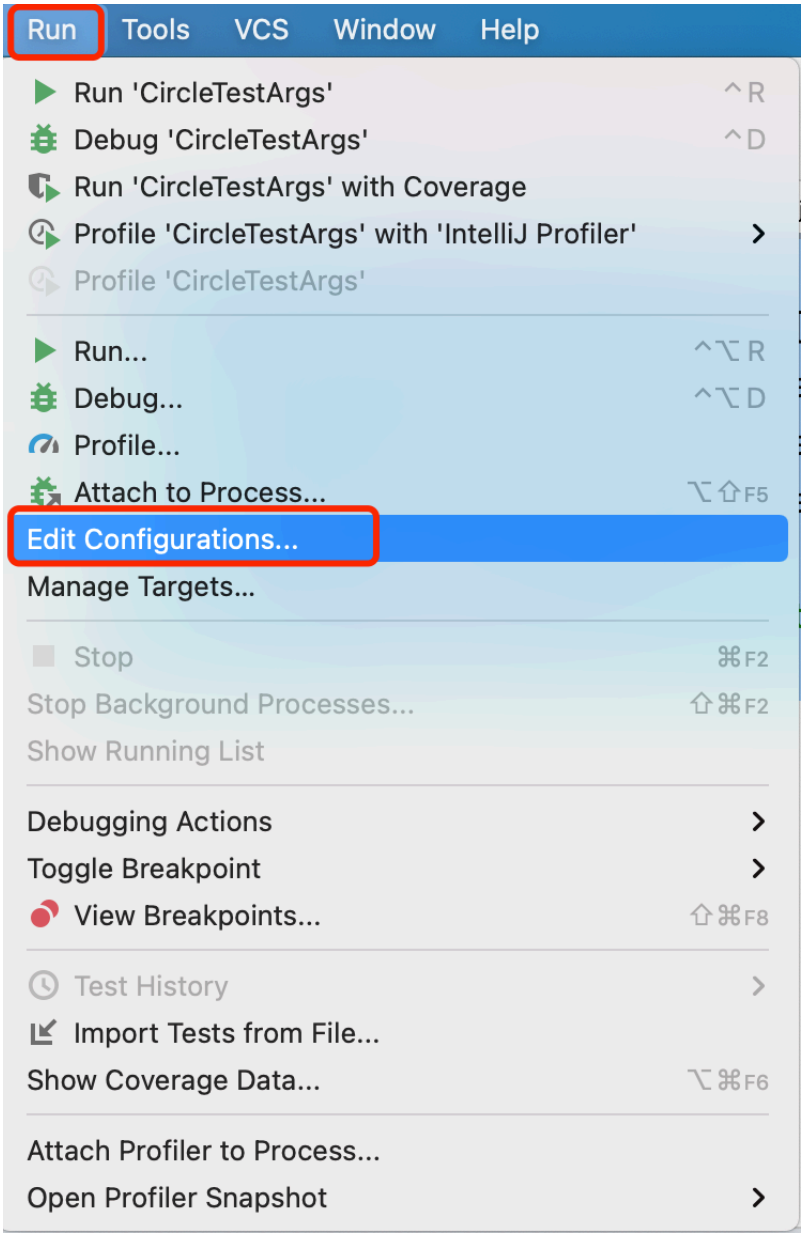
generate .class file : If we run the program by Intellij IDEA, a .class file about this main method will be created, but it will return :

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 0
out of bounds for length 0
  at CircleTestArgs.main(CircleTestArgs.java:3)
```

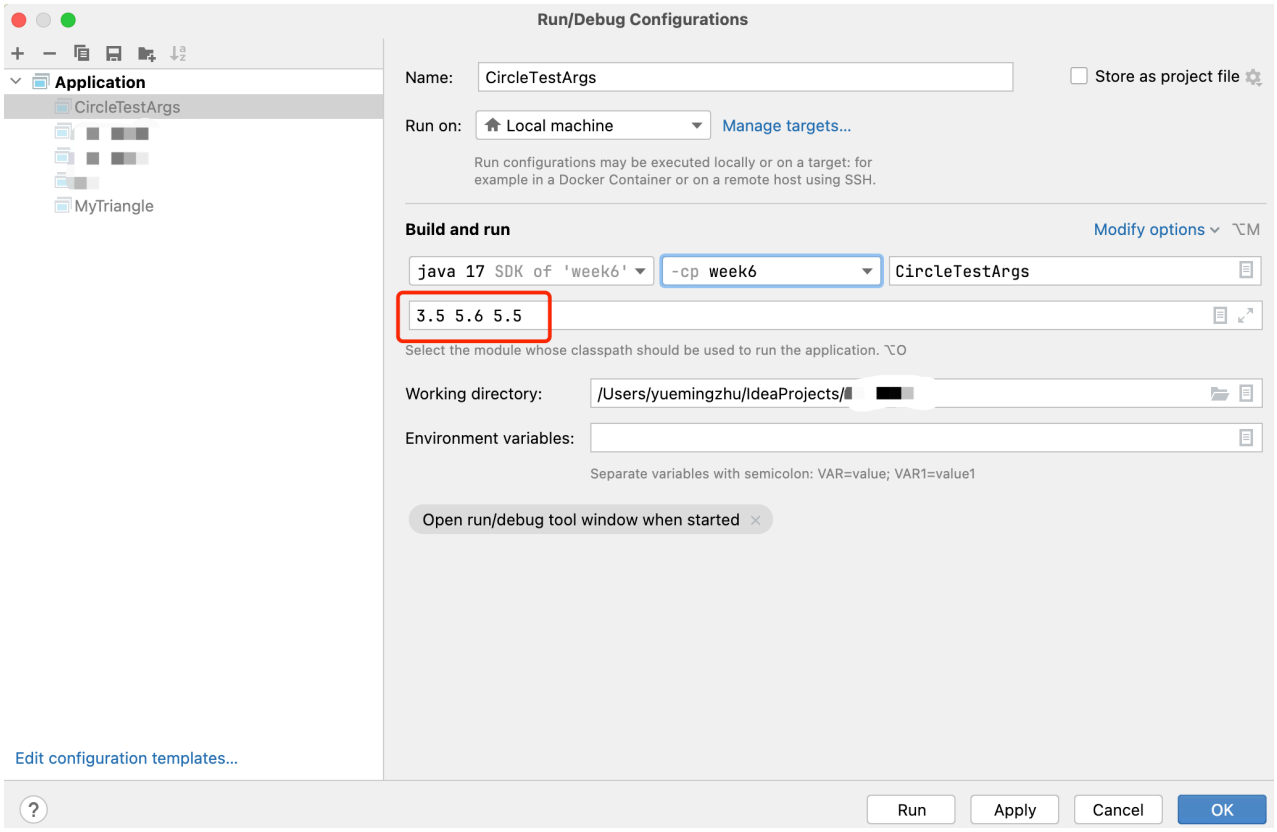
It is before we haven't set the arguments.

Set arguments:

Click run -> Edit Configurations



Input arguments here:



Then run it and it will return :

```
Area = 9.213273034052557
Perimeter = 14.6
```

By Command Line:

Open your terminal and visit the corrent path, then try:

```
java CircleTestArgs 3.3 4.4 5.1
```

The whole process is:

```
(base) yuemingzhu@YUEMINGs-Air src % ls
CircleTestArgs.java     MyTriangle.java
(base) yuemingzhu@YUEMINGs-Air src % javac CircleTestArgs.java
(base) yuemingzhu@YUEMINGs-Air src % java CircleTestArgs 3.3 4.4 5.1
Area = 7.182200219988304
Perimeter = 12.8
```