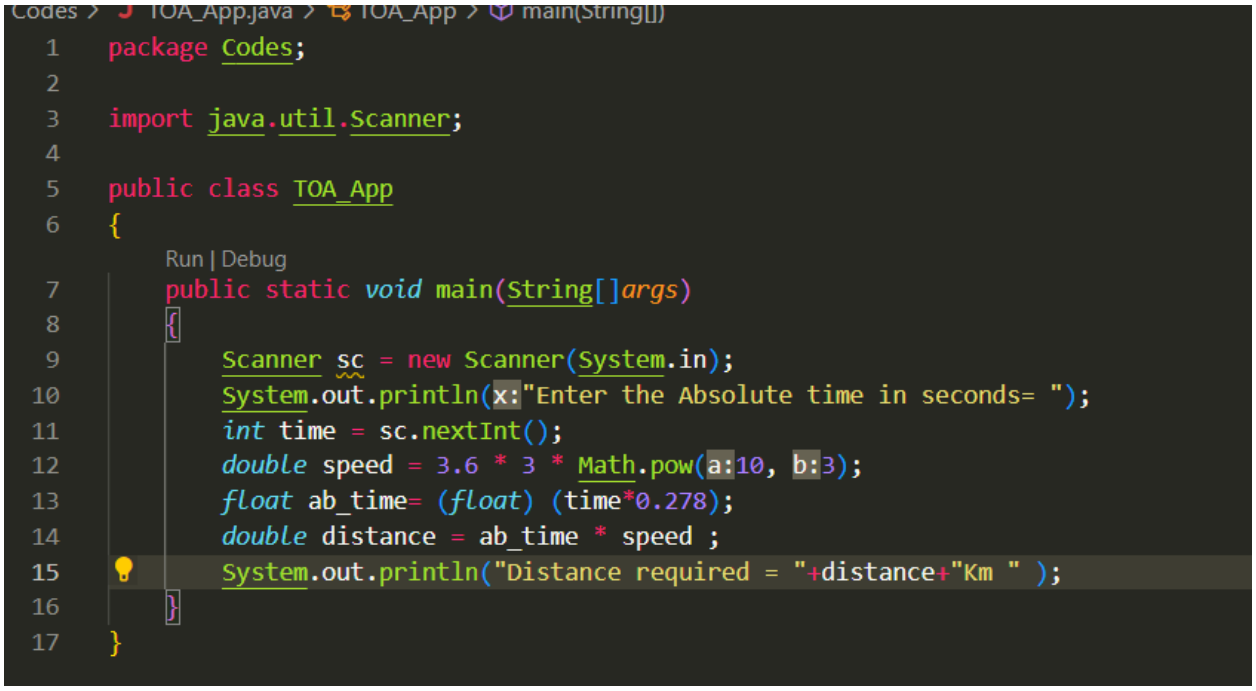
Implementations of Source localization techniques.

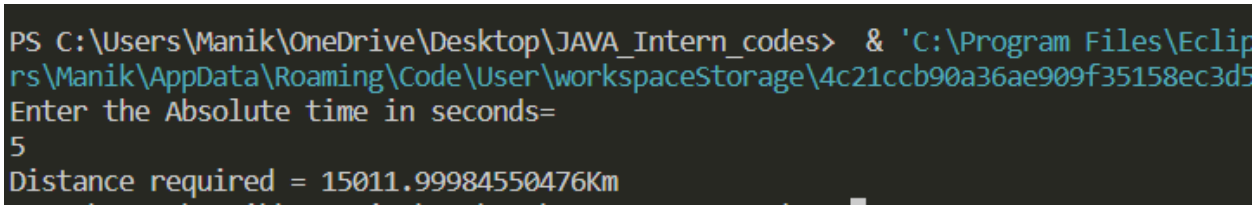
1. **Time of Arrival(TOA):**



```java
package Codes;

import java.util.Scanner;

public class TOA_App
{
    Run | Debug
    public static void main(String[]args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"Enter the Absolute time in seconds= ");
        int time = sc.nextInt();
        double speed = 3.6 * 3 * Math.pow(a:10, b:3);
        float ab_time= (float) (time*0.278);
        double distance = ab_time * speed ;
        System.out.println("Distance required = "+distance+"Km " );
    }
}
```



```
PS C:\Users\Manik\OneDrive\Desktop\JAVA_Intern_codes>  & 'C:\Program Files\Eclip
rs\Manik\AppData\Roaming\Code\User\workspaceStorage\4c21ccb90a36ae909f35158ec3d5
Enter the Absolute time in seconds=
5
Distance required = 15011.99984550476Km
```

Concept:

In TOA the object is localized with respect to the receiver.

Code Description;

- Here the Absolute time is taken as the input and multiplied by the propagation speed of the wave.
- After calculations, the Distance is returned as the Output in Km.

Future aspects of the code:

➢ A sensor can be used to measure the absolute time.

2. Time Difference of Arrival(TDOA):

Codes > TDOA_Apli.java > TDOA_Apli > main(String[])

```java
package Codes;
import java.util.Scanner;
public class TDOA_Apli {
    public static double calculateTDOA(double receptionTime1, double receptionTime2) {
        // Assuming the speed of signal propagation is known and constant
        double speedOfPropagation = 300000;
        return (receptionTime2 - receptionTime1) * speedOfPropagation;
    }
    Run | Debug
    public static void main(String[]args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"Enter the Relative position for Reciver 1");
        System.out.print(s:"X: ");
        int x1 = sc.nextInt();
        System.out.print(s:"\nY: ");
        int y1 = sc.nextInt();
        System.out.println(x:"Enter the Relative position for Reciver 2");
        System.out.print(s:"X: ");
        int x2 = sc.nextInt();
        System.out.print(s:"\nY: ");
        int y2 = sc.nextInt();

        System.out.println(x:"Enter the Relative Time");
        System.out.print(s:"At reciver 1 : ");
        double t1 = sc.nextDouble();
        System.out.print(s:"\nAt reciver 2 : ");
        double t2 = sc.nextDouble();

        double tdoa = calculateTDOA(t1, t2);
        double RelativeX= x1 + (tdoa *(x2 - x1));
        double RelativeY= y2 + (tdoa *(y2 - y1));
        System.out.println(x:"Relative position of object is");
        System.out.print("X: "+RelativeX);
        System.out.print("\nY: "+RelativeY);

    }
```

```
PS C:\Users\Manik\OneDrive\Desktop\JAVA_Intern_codes>  & 'C:\Program Files\
rs\Manik\AppData\Roaming\Code\User\workspaceStorage\4c21ccb90a36ae909f35158
Enter the Relative position for Reciver 1
X: 0

Y: 0
Enter the Relative position for Reciver 2
X: 3

Y: 0
Enter the Relative Time
At reciver 1 : 10.5

At reciver 2 : 12.3
Relative position of object is
X: 1620000.0000000007
Y: 0.0
```

Concept:

To implement TDOA , N must be greater than equal to 2, where N is the number of receivers. Here, I have considered 2 receivers whose position can be predetermined or as in here I have taken it as an input.

Code Description:

- Here first the position of receivers is taken as input.
- After determining the position the time for perception is noted.
- Then the object's relative position is calculated with respect to receiver 1.

Future aspects of the code:

➢ The absolute time can be noted by the sensor.

➢ Using GPS the position of receivers can be predefined.

3. Angle of Arrival(AOA):

```java
package Codes;
import java.util.Scanner;
public class DOA_appli {
    public static double calculateDOA(double angleMeasurement1, double angleMeasurement2, double antennaSpacing) {
        // Calculate the direction of arrival
        return Math.atan2(angleMeasurement2 - angleMeasurement1, antennaSpacing);
    }

    Run | Debug
    public static void main(String[]args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println(x:"Enter the Relative angular measurement in degrees:");
        System.out.print(s:"For Reciver1 : ");
        double x1 =Math.toRadians( sc.nextDouble());
        System.out.print(s:"\nFor Reciver2: ");
        double y1 =Math.toRadians( sc.nextDouble());
        System.out.println(x:"Enter the Distance between the antennas :");
        double dis = sc.nextDouble();
        double DOA = calculateDOA(x1, y1, dis);
        System.out.println(x:"Estimated direction of object w.r.t Reciver1");
        System.out.print("Pos: "+Math.toDegrees(DOA)+" deg");

    }
}
```

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL   GITLENS

PS C:\Users\Manik\OneDrive\Desktop\JAVA_Intern_codes>  c:; cd 'c:\Users\M
exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Manik\AppD
19f92a\bin' 'Codes.DOA_appli'
Enter the Relative angular measurement in degrees:
For Reciver1 : 40

For Reciver2: 60
Enter the Distance between the antennas :
10
Estimated direction of object w.r.t Reciver1
Pos: 1.9991882802315375 deg
```

Concept:

To implement AOA/DOA , 2 receivers are used. Each receiver measures the relative angle. Then the difference is considered as a rectangular component,'x'. The rectangular component 'y' is the distance between the receivers. Now to localize the object math function 'atan2' is used to convert the rectangular coordinates to polar coordinates.

Code Description:

- Here first the angular measurements by receivers are taken as inputs.
- Then the distance between receivers is noted.
- Then the direction of the object is calculated in degrees with respect to receiver 1.

Future aspects of the code:

- ➢ Through methods like beamforming, phase differences, or signal processing algorithms, the receivers can obtain the angles.
- ➢ Using GPS the distance of receivers can be predefined.