



گزارش پروژه

رژینا قیاسی

۴۰۱۷۲۵۱۳۷

۲۴ تیر ۱۴۰۲

فصل ۱

دسترسی دادن به برنامه:

با استفاده از کتابخانه های زیر دسترسی را برای برنامه ایجاد میکنیم که به گوشی متصل شود. کد دسترسی به صورت زیر است.

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
```

در اینجا با استفاده از `checkSelfPermission` دسترسی برنامه را چک میکنیم. اگر دسترسی به صورت `PERMISSION_GRANTED` باشد، تابع اصلی برنامه یعنی `startCalculations()` اجرا می شود.

```
// Check and request location permission if not granted
// if (ContextCompat.checkSelfPermission(this,
//     Manifest.permission.ACCESS_FINE_LOCATION)

// != PackageManager.PERMISSION_GRANTED) {
//     ActivityCompat.requestPermissions(this,
//         new
//         String[]{Manifest.permission.ACCESS_FINE_LOCATION},
//         PERMISSIONS_REQUEST_CODE);
//     } else {
//         startCalculations();
//     }
// }
```

اگر اجازه برای برنامه صادر شود؛ برنامه به سراغ اجرا کردن تابع اصلی می رود. در غیر اینصورت برنامه پیام "اجازه منع شد" را در قسمت log نشان می دهد.

```
public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)
{
    if (requestCode == PERMISSIONS_REQUEST_CODE) {
        if (grantResults.length > 0 && grantResults[0] ==
            PackageManager.PERMISSION_GRANTED) {
            startCalculations();
        } else {
            Log.e("MainActivity", "Location permission denied");
        }
    }
}
```

فصل ۲

تابع اصلی : startCalculations

در این تابع باید پارامترهای p_0 , d , β را بگیرد و مقدار نویز گاوسی را حساب کند. ابتدا پارامترها را تعریف میکنیم:

```
private void startCalculations() {
    results = new ArrayList<>();

    for (int i = 0; i < NUM_CALCULATIONS; i++) {
        double xT = getRandomPosition();
        double yT = getRandomPosition();
        double xI = getRandomPosition();
        double yI = getRandomPosition();

        double d = Math.sqrt(Math.pow(xT - xI, 2) + Math.pow(yT - yI, 2));

        double beta = getBetaFromBTS();
        double p0 = getP0FromBTS();
        double pr = p0 - 10 * beta * Math.log10(d);

        double xSigma = pr - p0 + 10 * beta * Math.log10(d);

        results.add(xSigma);
    }
}
```

در کد بالا هر پارامتر با یکی از توابع تعریف شده در پروژه ایجاد می شود . که در ادامه توضیح داده خواهد شد. در نهایت مقدار d با توجه به تابع زیر محاسبه می شود.

$$d = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2}$$

در نهایت مقدار نویز گاوسی را با فرمول زیر محاسبه می کند.

$$pr = p_0 - 10 \log d + \chi_\sigma$$

فصل ۳

نمایش نتیجه

در آخر برای محاسبه میانگین نویز گاوسی که در این پروژه من ۱۰۰ در نظر گرفتیم و نشان دادن مقدار آن از کد زیر استفاده می کنم:

```
double averageXSigma = calculateAverage(results);
Log.d("MainActivity", "Average X: " + averageXSigma);
TextView average_x_sigma_textview=findViewById(R.id.average_x_sigma_textview);
average_x_sigma_textview.setText("fit X: " + averageXSigma);
```

در کد بالا TextView را فراخوانی میکنیم تا نتیجه نشان داده شود.

فصل ۴

توابع مورد استفاده در پروژه

- getRandomPosition()

در این تابع پوزیشن مقادیر xt,xi,yt,yi را بدست می آوریم. (که به دلیل مشکلات گوشی من از تابع رندم استفاده کردم).

```
private double getRandomPosition() {  
    // Generate a random position between 0 and 100  
    Random random = new Random();  
    return random.nextDouble() * 100;  
}
```

- getBetaFromBTS() double private

این تابع برای گرفتن پارامتر beta برای برنامه است. که برای این کار ابتدا باید به اطلاعات سیمکارت دسترسی داشته باشد و برای اجازه دسترسی از کتابخانه های زیر استفاده می کند:

```
import android.telephony.CellInfo;  
import android.telephony.CellInfoLte;  
import android.telephony.CellSignalStrength;  
import android.telephony.TelephonyManager;
```

حال باید از توابع درون کتابخانه ها برای بررسی دسترسی و لیست اطلاعات درون سیمکارت استفاده کنیم. برای این کار یک لیست به عنوان List cellInfoList می سازیم و مقادیری که می خواهیم را از لیست میگیریم. برای این کار یک تابع با عنوان readBtsInfo\ تعریف می کنیم و تابع این صورت است :

```
private double readBtsInfo1() {  
  
    TelephonyManager telephonyManager = (TelephonyManager) getSystemService(  
        TELEPHONY_SERVICE);  
    List cellInfoList = telephonyManager.getAllCellInfo();  
    double rsrp=0; // rsrp  
  
    if (cellInfoList != null) {  
        for (CellInfo cellInfo : cellInfoList) {  
            if (cellInfo instanceof CellInfoLte) {  
                CellSignalStrength signalStrength = ((CellInfoLte) cellInfo).  
                    getCellSignalStrength();  
                rsrp= signalStrength.getLevel(); // p0  
            }  
        }  
    }  
  
    return rsrp;  
}
```

در این تابع یک مقدار پیشفرض (rsrp) را تعریف میکنیم و در صورت بودن اطلاعات مقدار را میگیریم و پارامتر را بر میگردانیم. در آخر در تابع اصلی این تابع را فراخوانی کرده و مقدار را میگیریم:

```
private double getBetaFromBTS() {  
    // TODO: Implement method to retrieve value from BTS  
    double rsrpp= readBtsInfo1();  
    return rsrpp;  
}
```

- `private double getPFromBTS()` : برای این تابع نیز به روش بالا عمل میکنیم و تابع بع صورت زیر می باشد :

```
private double getPFromBTS() {
// TODO: Implement method to retrieve P0 value from BTS
double pp=readBtsInfo2();
return pp;
}
```

- `calculateAverage(List<Double> values)` : در اخر نیز مقدار میانگین نویز گاوسی را حساب می کنیم:

```
private double calculateAverage(List<Double> values) {
double sum = 0.0;
for (Double value : values) {
sum += value;
}
return sum / values.size();
}
```