

모바일 시스템 프로그래밍

04 Location Manager 1

2017 1학기

강승우

위치 데이터의 활용

- Location-Based Application

- 사용자의 위치 정보를 바탕으로 필요한 정보나 서비스를 제공하는 애플리케이션

- 예

- 지도
- 네비게이션
- 위치 정보 로깅
- 장소 태깅(SNS)
- 위치 기반 쿠폰: 상점의 특정 반경 안에 들어가면 쿠폰 전송
- 현재 위치에서 가까운 주유소


위치 정보 얻는 방법

- GPS (Global Positioning System)
 - GPS 위성으로부터 수신한 신호를 기반으로 현재 위치를 계산
 - 실외에서만 사용 가능, 배터리 소모 큼
 - 오차: 수 미터 – 수십 미터
- 전화 기지국 (Cellular 망의 cell tower)
 - 여러 기지국에서 오는 전파의 시간 차이, 세기 등을 이용하여 위치 계산
 - 오차 범위가 GPS보다 큼
 - 오차: 수백 미터
- WiFi AP
 - 특정 AP로부터 수신한 신호의 세기, AP의 위치 등을 바탕으로 위치 계산
 - 실내 위치 추적용으로 많이 사용

안드로이드 Location API components

- 관련 주요 class 및 interface

- Class

- LocationManager
 - LocationProvider 
 - Location
 - Criteria

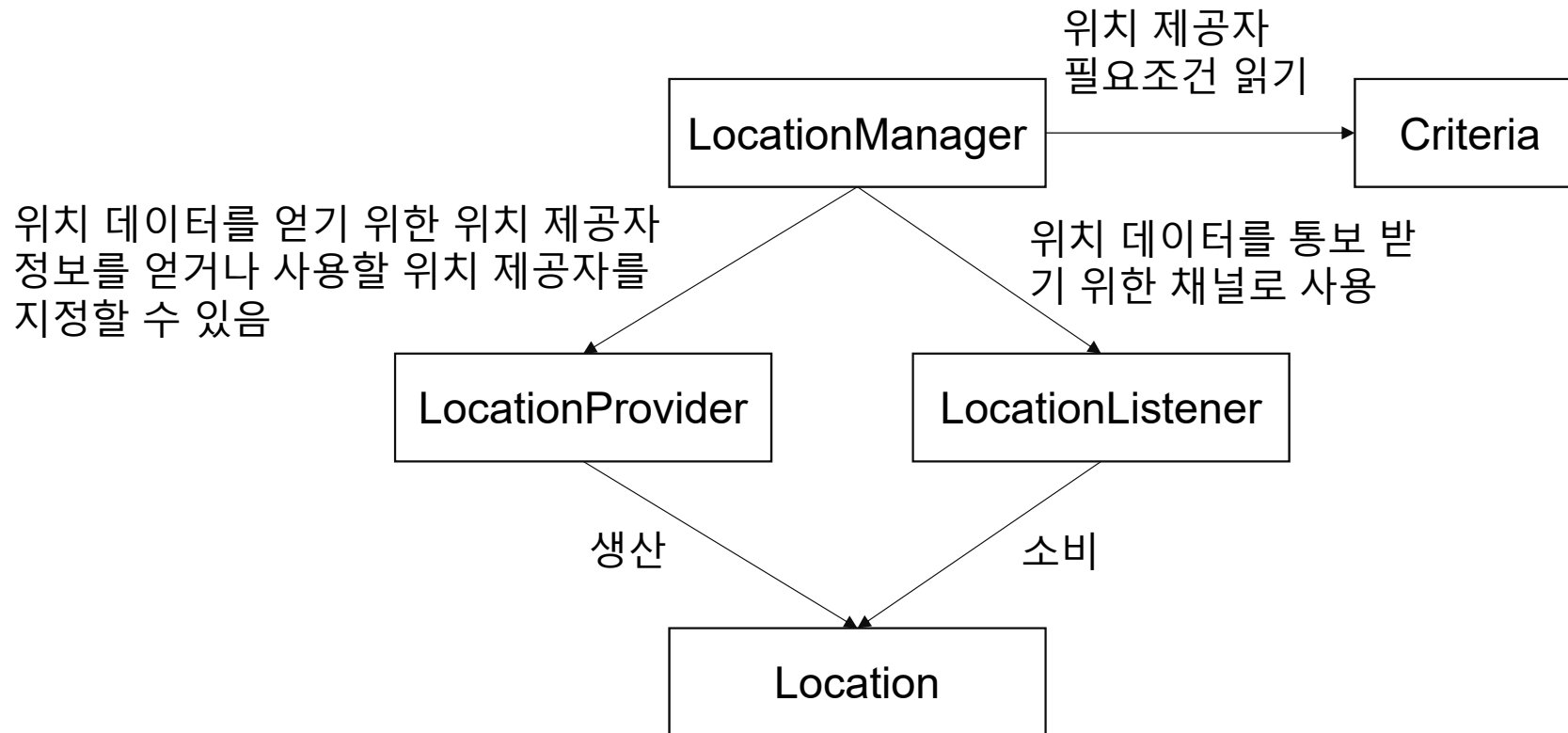
- Interface

- LocationListener

- ✓ Android framework location APIs

<https://developer.android.com/reference/android/location/package-summary.html?hl=ko>

Location API components 관계도



LocationManager 클래스

- 안드로이드 위치 서비스를 사용하기 위한 시작점
 - SensorManager와 같이 안드로이드가 제공하는 시스템 서비스
 - 앱에서 변경된 위치 정보를 제공받기를 원할 때 이를 LocationManager를 이용하여 안드로이드 시스템에게 알린다
 - 이용 가능한 위치 제공자(LocationProvider) 리스트, GPS 상태 정보와 같은 위치 시스템의 현재 상태 정보를 제공하거나 기기에 최근에 기록된(캐시된) 위치 정보를 제공할 수 있다

LocationManager manager = (LocationManager) getSystemService(Context.*LOCATION_SERVICE*);

<https://developer.android.com/reference/android/location/LocationManager.html?hl=ko>

LocationProvider 클래스

- 안드로이드에서 위치 정보를 제공하는 다양한 공급원에 대해 추상화(abstraction)한 것
 - 종류
 - GPS 이용
 - Cell ID 이용
 - WiFi AP 이용
 - 다른 앱에서 위치 정보를 요청하여 사용하고 있을 때 그것을 공유하여 사용
 - 위치 정보를 제공하는 방법은 다르지만 LocationProvider 클래스를 통해 모두 동일한 방식으로 앱과 인터랙션하고, 앱에 같은 형태로 데이터를 제공
- LocationManager 클래스에 정의된 LocationProvider를 지칭하는 이름
 - LocationManager.GPS_PROVIDER → GPS 이용
 - LocationManager.NETWORK_PROVIDER → Cell ID, AP 이용
 - LocationManager.PASSIVE_PROVIDER → 다른 앱이 이미 위치 정보를 요청한 것이 있을 때 그 정보를 공유

<https://developer.android.com/reference/android/location/LocationProvider.html?hl=ko>

Location 클래스

- 위치 제공자(LocationProvider)에서 앱에 제공하는 실제 위치 데이터를 캡슐화 한 것
 - 기본적인 데이터
 - 위도 (latitude)
 - 경도 (longitude)
 - 고도 (altitude), 시각 (timestamp), 속도 (velocity)
 - 앱이 Location 객체를 받은 뒤에는 앱에서 필요한 데이터 처리를 하면 됨
 - 다양한 위치 데이터 속성을 가지고 있으나 모든 LocationProvider가 모든 속성을 채워서 제공하지는 않음
 - 예) 고도 정보는 없는 경우
 - 앱에서 Location 객체가 어떤 정보를 포함하고 있는지 검사하는 메소드 제공
 - hasAltitude(), hasSpeed(), hasAccuracy()

<https://developer.android.com/reference/android/location/Location.html?hl=ko>

Criteria 클래스

- 원하는 특징을 가진 위치 제공자(LocationProvider) 목록을 얻기 위한 검색 용도로 사용
 - Criteria 클래스의 속성을 설정함으로써 관심 있는 위치 제공자의 특성을 지정할 수 있다
- 속성
 - accuracy: 위치 제공자의 전반적인 정확도 레벨
 - Criteria.ACCURACY_FINE / Criteria.ACCURACY_COARSE
 - altitudeRequired: 위치 제공자가 고도 정보를 제공할 필요가 있는지 여부
 - bearingRequired: 위치 제공자가 방향(이동 방향)을 제공할 필요가 있는지 여부
 - bearingAccuracy: 요구되는 방향 정보의 정확도
 - Criteria.ACCURACY_HIGH / Criteria.ACCURACY_LOW
 - costAllowed: 위치 제공자가 비용이 드는 것을 허용하는지 여부
 - horizontalAccuracy: 요구되는 위도 및 경도의 정확도
 - Criteria.ACCURACY_HIGH / Criteria.ACCURACY_LOW / Criteria.ACCURACY_MEDIUM
 - powerRequirement: 위치 제공자가 요구하는 배터리 소모량
 - Criteria.POWER_LOW / Criteria.POWER_MEDIUM / Criteria.POWER_HIGH
- 각 속성의 값을 설정하거나 확인할 수 있는 메소드 제공



<https://developer.android.com/reference/android/location/Criteria.html?hl=ko>

Criteria 클래스

- 사용 예제

```
LocationManager manager =  
(LocationManager) getSystemService(Context.LOCATION_SERVICE);  
Criteria criteria = new Criteria();  
criteria.setAccuracy(Criteria.ACCURACY_COARSE);  
List<String> enabledProviders = manager.getProviders(criteria, true);
```

✓ LocationManager 클래스에 정의된 메소드

public List<String> getProviders (Criteria criteria, boolean enabledOnly) 

→ criteria를 만족하면서 현재 enabled 상태인 location provider의 리스트를 반환


❖ Criteria의 여러 속성 값을 변경해보면서 그에 따라 어떤 LocationProvider가 반환이 되는지 확인해 보자

LocationListener 인터페이스

- 기기의 현재 위치가 변할 때 또는 위치 서비스의 상태가 변할 때 호출되는 콜백 메소드를 포함
 - 앱이 위치 데이터를 받아서 처리를 하거나, 위치 서비스의 상태 변화에 대응하여 처리를 하고자 할 때 LocationListener를 구현해야 한다
 - 구현된 LocationListener를 LocationManager에 등록하거나 등록 해제 하여 사용한다
- 구현해야 하는 메소드
 - abstract void onLocationChanged(Location location)
 - abstract void onProviderDisabled(String provider)
 - abstract void onProviderEnabled(String provider)
 - 🗨 abstract void onStatusChanged(String provider, int status, Bundle extras)

<https://developer.android.com/reference/android/location/LocationListener.html?hl=ko>

LocationListener 인터페이스

- onLocationChanged(Location location) 
 - 위치 데이터를 받고자 할 때 구현해야 하는 메소드
 - 앱에서 사용할 새로운 위치 데이터가 준비되었을 때 호출된다
 - 이 메소드 안에 Location 객체에 담긴 데이터를 이용하여 필요한 작업을 수행하도록 코드를 작성한다
 - 앱에서 LocationListener를 구현하기 위해 작성해야 하는 유일한 메소드가 될 수도 있다
 - 다른 메소드가 필요하지 않은 경우 코드를 작성하지는 않더라도 메소드를 포함해야 한다 (그렇지 않은 경우 컴파일 에러 발생)

LocationListener 인터페이스

- onLocationChanged(Location location) 메소드 구현 예제

❖ 화면 상의 TextView 에 위도, 경도, Provider 정보를 표시하는 코드

```
@Override
public void onLocationChanged(Location location) {
    latitudeValue.setText(String.valueOf(location.getLatitude()));
    longitudeValue.setText(String.valueOf(location.getLongitude()));
    providerValue.setText(String.valueOf(location.getProvider()));
}
```

LocationListener 인터페이스

- onProviderDisabled(String provider)

/ onProviderEnabled(String provider)

- 사용자가 위치 제공자를 위치 설정 메뉴에서 사용 가능 혹은 사용 불가능으로 변경할 때 이를 통지 받을 수 있도록 해주는 메소드
 - 예: 설정에서 GPS를 사용하지 않도록 변경한 경우
- 사용 가능했던 위치 제공자가 사용 불가능해진 경우 다른 위치 제공자를 사용하여 앱에서 위치 정보를 얻을 수 있도록 변경한다든지 하는 작업을 할 수 있다

LocationListener 인터페이스

- `onStatusChanged(String provider, int status, Bundle extras)`
 - 위치 제공자의 이용 가능 상태의 변화가 발생했을 때 호출되는 메소드
 - 사용자의 설정 변경에 의한 것이 아님
 - 사용자가 GPS를 이용하도록 설정했음에도 GPS 신호가 수신이 안 된다거나 등의 이유로 위치 데이터를 얻을 수 없는 경우 호출

LocationListener 인터페이스

- `onStatusChanged(String provider, int status, Bundle extras)`
 - status 값
 - `LocationProvider.OUT_OF_SERVICE`
 - 위치 제공자는 현재 이용 가능하지 않고 가까운 미래에도 가능하지 않을 것으로 예상
 - `LocationProvider.TEMPORARILY_UNAVAILABLE`
 - 위치 제공자는 현재 이용 가능하지 않지만, 조만간 이용 가능해질 것으로 예상
 - `LocationProvider.AVAILABLE`
 - 위치 제공자는 현재 이용 가능
 - Bundle extras
 - 추가적인 위치 제공자의 정보 (key/value pairs 형태)
 - 예: GPS 제공자의 경우 위치 업데이트를 얻는데 사용된 위성의 개수

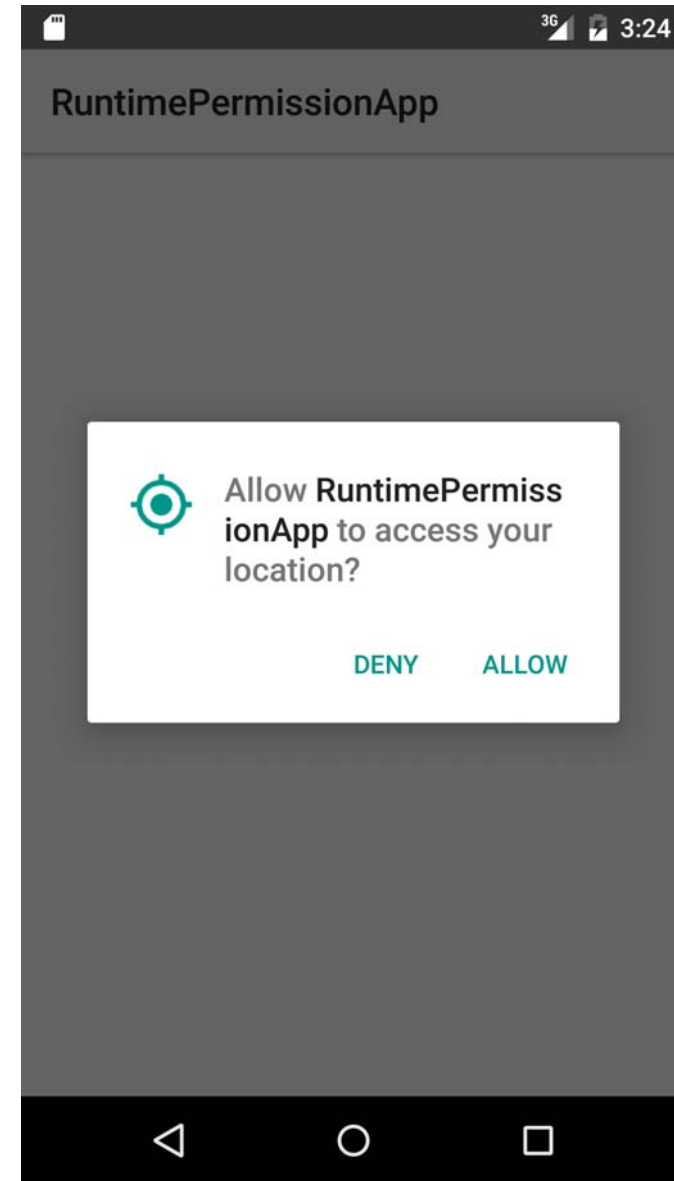
Permission 설정

- 매니페스트 파일에 위치 서비스를 사용하겠다는 선언을 해야 함
 - 두 가지 권한
 - android.permission.ACCESS_FINE_LOCATION
 - android.permission.ACCESS_COARSE_LOCATION
 - 위치 서비스가 앱에 제공하는 정확도를 의미하여, 결국 어떤 위치 제공자를 사용할 지와 관련됨
- 예
`<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />`

Permission 설정

- Android 6.0 이후부터는 사용자로부터 Runtime permission을 얻어야 함
 - 매니페스트 파일에 permission 선언하는 것과 별도

앱을 처음 실행할 경우 오른쪽 그림과 같이 위치 정보에 접근을 허용할 것인지 묻는 대화상자가 표시



적절한 위치 제공자 결정

- 사용할 위치 제공자 선언 방법
 - 1) 원하는 위치 제공자(GPS, Network, Passive)를 명시적으로 LocationManager에 등록
 - 2) Criteria 객체에 원하는 속성을 설정하고 이를 LocationManager에 전달
- 어떤 기준으로 위치 제공자를 결정하면 좋을까?
 - 고려 사항
 - 위치 제공자마다 다른 정확도와 배터리 소모를 보임
 - 위치 데이터를 얻는데 걸리는 시간이 다름
 - 앱에서 필요한 정확도가 어느 정도인가? 배터리 소모를 최대한 줄여야 하는가?
이런 질문에 답을 먼저 해봐야 함

적절한 위치 제공자 결정

- GPS 위치 제공자
 - 다른 위치 제공자와 비교해서 상대적으로 정확하다
 - 위성 신호를 받기 위해서 Line of Sight를 요구한다
 - 실내에서는 신호를 받기 어렵다
 - 실외라 하더라도 도시의 빌딩 숲에서는 오차가 클 수 있다
 - 배터리 소모가 많다
 - 위성 신호를 수신하여 위치를 판독해야 하므로 데이터를 얻는데 시간이 오래 걸릴 수 있다
 - TTFF (Time To First Fix): 초기 위치 결정 시간
 - 요구 권한
 - `android.permission.ACCESS_FINE_LOCATION`

적절한 위치 제공자 결정

- Network 위치 제공자
 - WiFi 네트워크, Cell ID 위치를 사용
 - GPS 위치 제공자와 비교하여 상대적으로 부정확하지만, 빠른 TTFF를 보일 수 있다
 - GPS 위치 제공자와 비교하여 배터리 소모가 적다
 - 요구 권한
 - `android.permission.ACCESS_COARSE_LOCATION`

적절한 위치 제공자 결정

- Passive 위치 제공자

- 다른 앱이 위치 서비스를 이용하고 있을 때 이에 편승해 위치 데이터를 얻을 수 있다
 - GPS 위치 제공자 혹은 Network 위치 제공자
- 추가적인 배터리 소모 없이 데이터를 얻는다
- 데이터를 받는 시간, 빈도를 제어할 수 없고, 위치 데이터를 받는 것이 보장되지 않는다
 - 앱이 전면(foreground)에서 사용자와 인터랙션 하는 경우에 사용하는 것은 적합하지 않음
 - 명시적으로 위치 데이터를 요청하지 않고 백그라운드에서 실행되는 동안 애플리케이션의 데이터를 최신으로 유지하고자 할 때 사용
- 요구 권한
 - `android.permission.ACCESS_FINE_LOCATION`

이용 가능한 LocationProvider 출력 예제

- 주요 코드

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    LocationManager manager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);  
    List<String> providers = manager.getAllProviders();  
    String s = "";  
  
    for(int i = 0; i < providers.size(); i++) {  
        s += "위치 제공자 " + providers.get(i) + "의 상태: "  
            + manager.isProviderEnabled(providers.get(i)) + "\n";  
    }  
    // output.setText(s);  
}
```

output은 UI layout에 정의된 TextView를 참조하는 객체로
위에서 선언이 되어 있어야 함

사용자의 위치 얻기

- LocationManager에게 위치 업데이트를 받겠다고 요청을 먼저 해야 함
- 위치 데이터 요청 메소드
 - LocationManager 클래스에 정의된 메소드
 - requestLocationUpdates()
 - requestSingleUpdate()
 - <https://developer.android.com/reference/android/location/LocationManager.html?hl=ko>
 - 다른 매개 변수를 갖는 형태로 다양하게 정의되어 있음

위치 데이터 요청 메소드

`public void requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener)`

- provider: 사용 등록할 위치 제공자 이름
- minTime: 위치 업데이트 간의 최소 시간 간격 (ms)
- minDistance: 위치 업데이트 간의 최소 거리 (m)
- listener: 위치 업데이트가 발생했을 때 호출될 `onLocationChanged()` 메소드가 구현된 `LocationListener` 객체
- minTime, minDistance
 - 모두 0으로 지정하면 가급적 자주 위치 정보가 갱신되므로 정확도는 향상되지만 그만큼 배터리를 많이 소모하므로 적당한 수준에서 결정해야 한다

`public void requestSingleUpdate(String provider, LocationListener listener, Looper looper)`

- looper: 콜백을 구현하기 위해 사용될 메시지 큐를 갖는 `Looper` 객체, UI 스레드에서 콜백을 받는 경우 필요 없음 (null로 설정)

위치 데이터 요청 중단

`void removeUpdates(LocationListener listener)`

- 해당 listener에 대한 위치 업데이트를 제거

실습: 위치 데이터 표시

- 예제 프로젝트 이름
 - MSP03LocationRequest
- LocationListener 구현
 - onLocationChanged(Location location) 메소드 외
- onCreate()
 - LocationManager 객체 얻음
- onResume()
 - requestLocationUpdates 호출 (runtime permission check 관련 코드 참고)
- onPause()
 - removeUpdates 호출

실습: 위치 데이터 표시

- GPS provider, Network provider를 변경하면서 테스트 해볼 것
- onProviderEnabled / onProviderDisabled 메소드를 사용
 - 사용 가능 여부에 따라 동적으로 Location Provider를 변경하는 방식으로 코드를 수정해 보자
 - GPS provider를 이용하다가 사용자가 설정에서 GPS를 사용하지 않도록 하면 이를 감지하여 Network provider를 이용하도록 변경하고,
 - 다시 GPS를 이용하도록 설정한 경우 GPS provider를 이용하도록 변경한다