www.osgeo.kr

OSGeo
한국어지부
Korean Chapter

FOSS4G
#Code the #Earth, KOREA 2025
#Share the #Future

QGIS 공간 처리 도구 확장

- 모델 설계자와 GPT 활용 -

QGIS

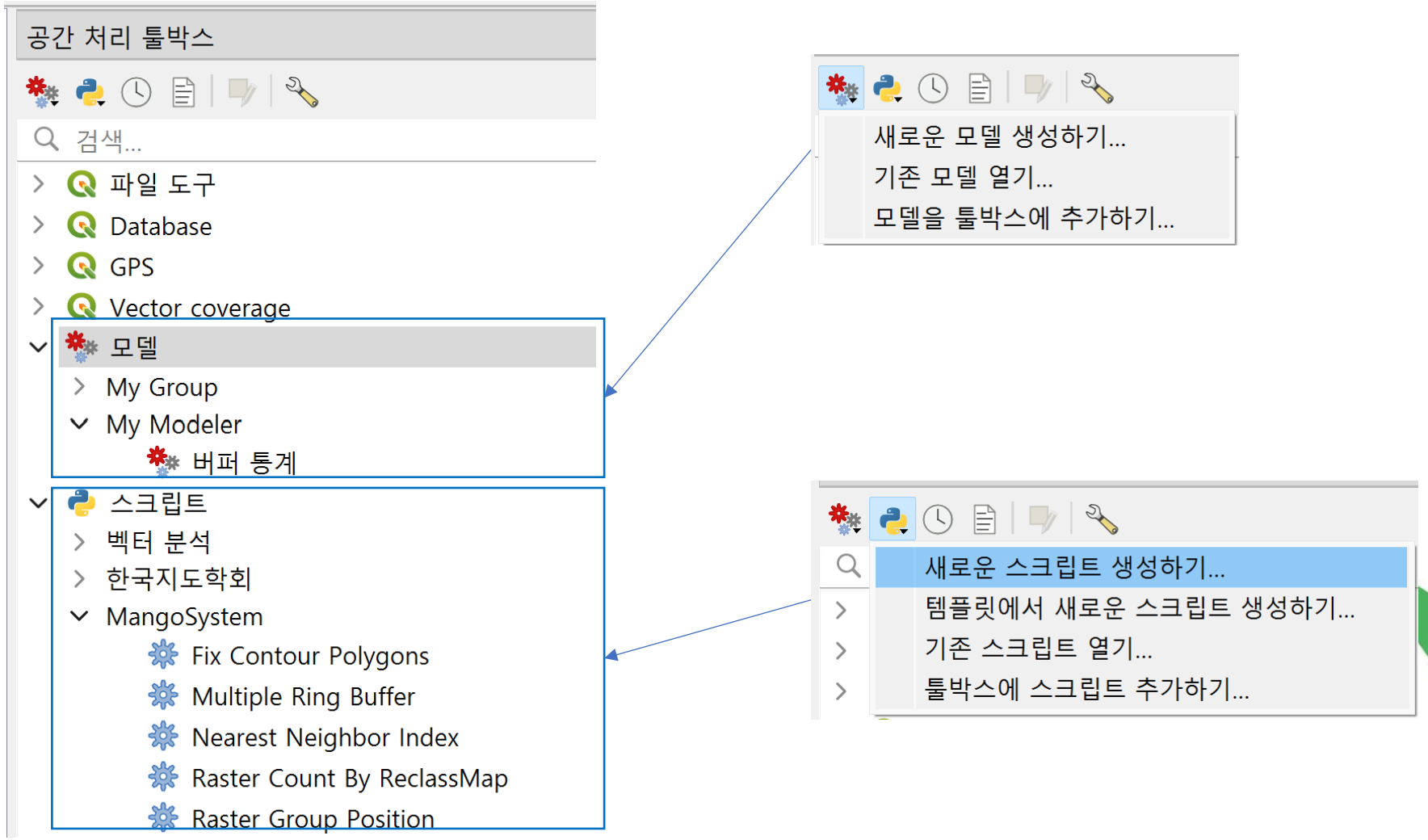
이민파 ㈜망고시스템
mapplus@gmail.com

https://www.osgeo.kr/

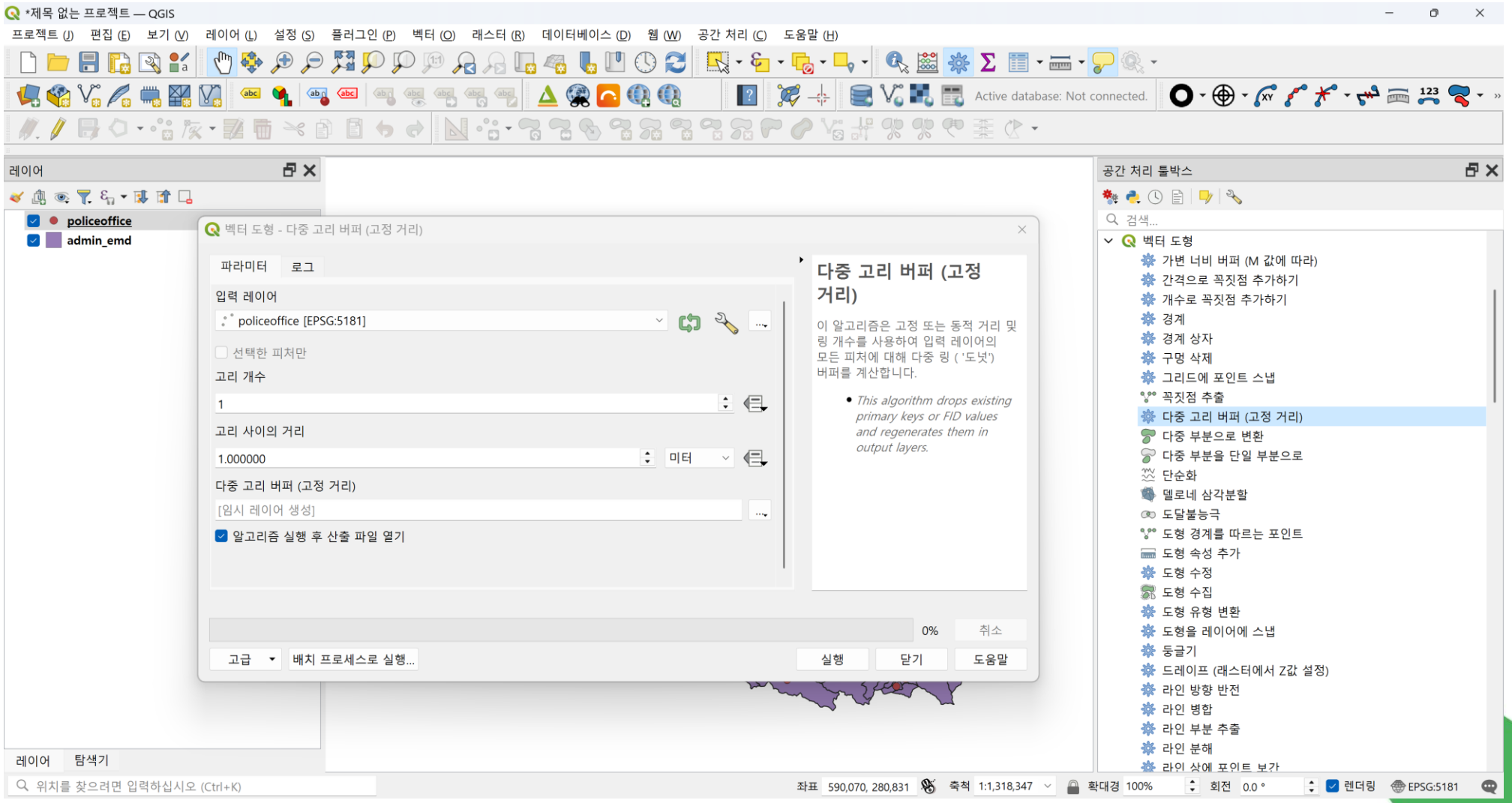
1. QGIS 공간 처리 툴박스의 이해

목표: QGIS의 공간 처리 툴박스 확장을 위한 핵심 개념을 이해한다.

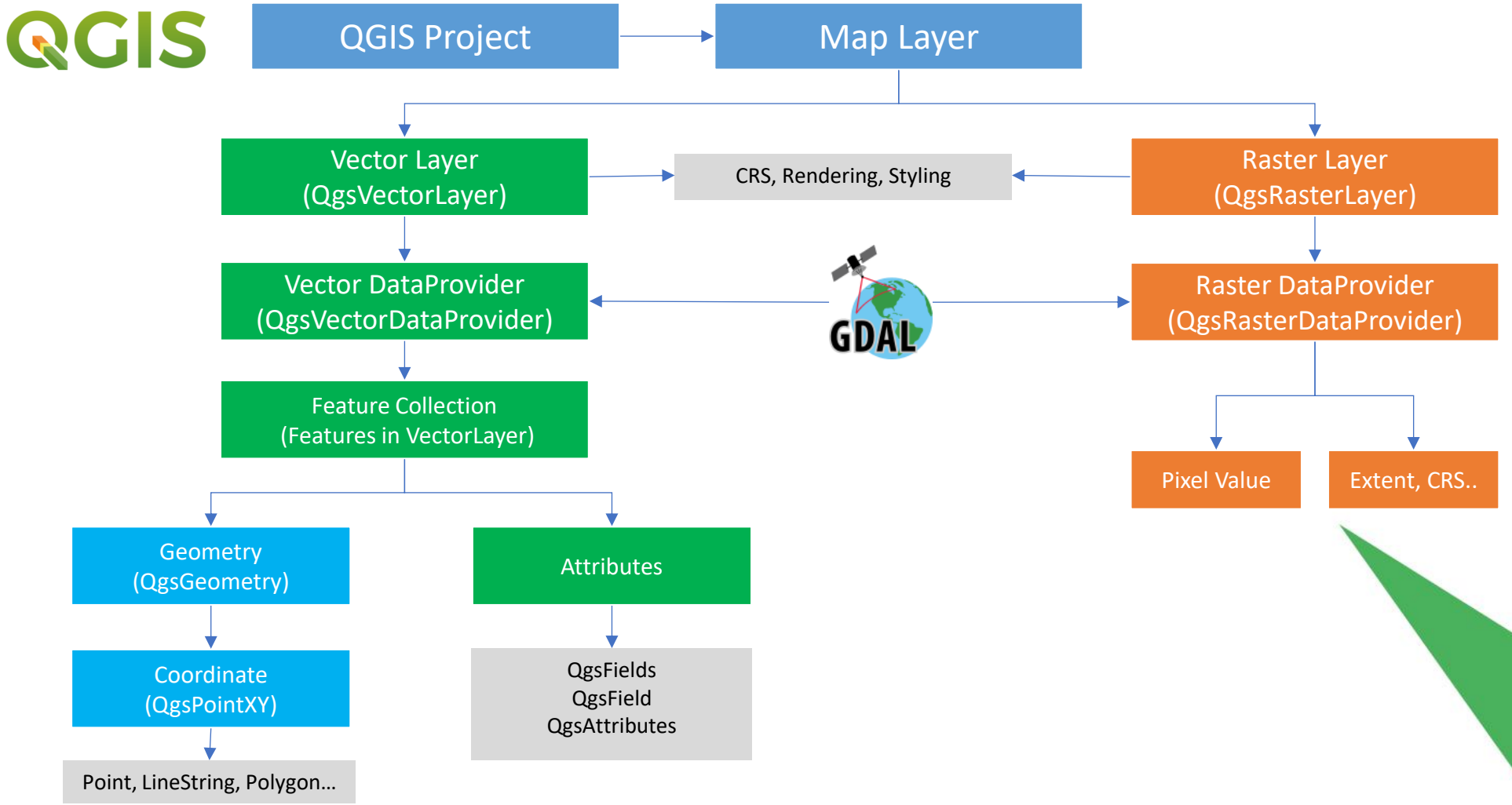
QGIS 공간 처리 툴박스



QGIS 공간 처리 툴박스



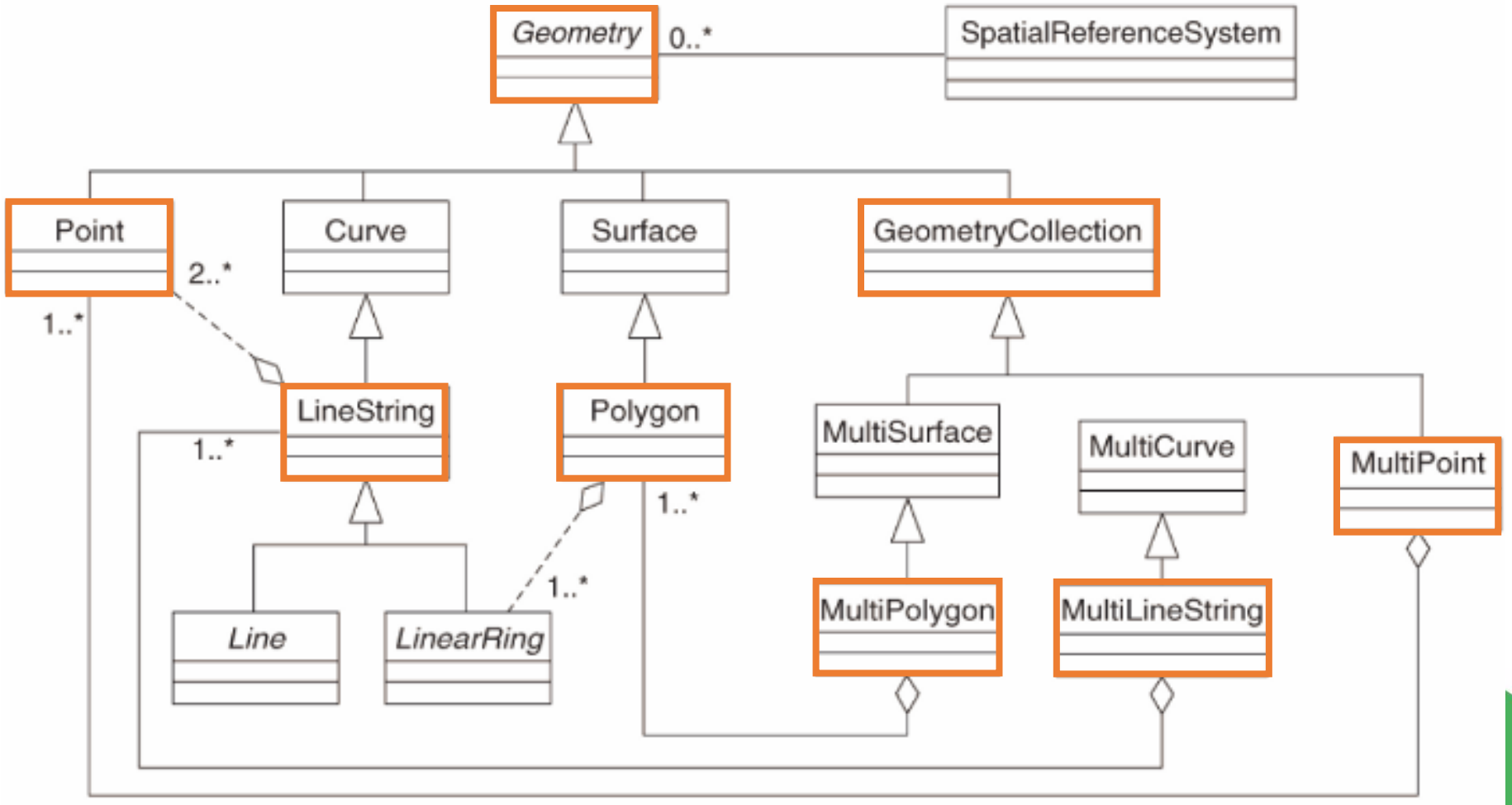
PyQGIS 벡터&래스터 레이어 핵심 구조



PyQGIS 벡터, 래스터 레이어 핵심 구조

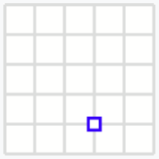
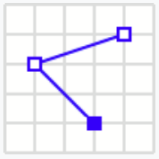
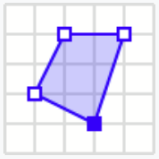
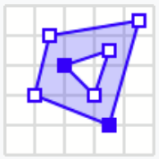
용어	정의	PyQGIS 클래스
Map Layer	지도 위에 올려 보여지는 레이어 객체	QgsMapLayer
VectorLayer	Feature 집합	QgsVectorLayer
RasterLayer	픽셀 그리드	QgsRasterLayer
Feature	속성 + 지오메트리 1세트	QgsFeature
Fields	속성 구조 정의	QgsFields, QgsField
Attributes	속성값 리스트	QgsAttributes
Geometry	공간 형태	QgsGeometry

Simple Feature for SQL (SFS) Model



Simple Feature for SQL (SFS) Model

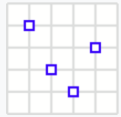
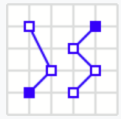
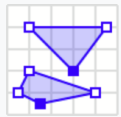
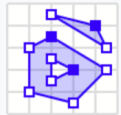
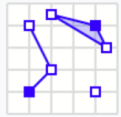
❖ Geometry primitives (2D)

Type	Examples	
Point		`POINT (30 10)`
LineString		`LINESTRING (30 10, 10 30, 40 40)`
Polygon		`POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))`
		`POLYGON ((35 10, 45 45, 15 40, 10 20, 35 10), (20 30, 35 35, 30 20, 20 30))`

출처: https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry

Simple Feature for SQL (SFS) Model

❖ Multipart geometries (2D)

Type	Examples	
MultiPoint		`MULTIPOINT ((10 40), (40 30), (20 20), (30 10))`
		`MULTIPOINT (10 40, 40 30, 20 20, 30 10)`
MultiLineString		`MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))`
MultiPolygon		`MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))`
		`MULTIPOLYGON (((40 40, 20 45, 45 30, 40 40)), ((20 35, 10 30, 10 10, 30 5, 45 20, 20 35), (30 20, 20 15, 20 25, 30 20)))`
GeometryCollection		`GEOMETRYCOLLECTION (POINT (40 10), LINESTRING (10 10, 20 20, 10 40), POLYGON ((40 40, 20 45, 45 30, 40 40)))`

출처: https://en.wikipedia.org/wiki/Well-known_text_representation_of_geometry

PyQGIS - QGIS Python API

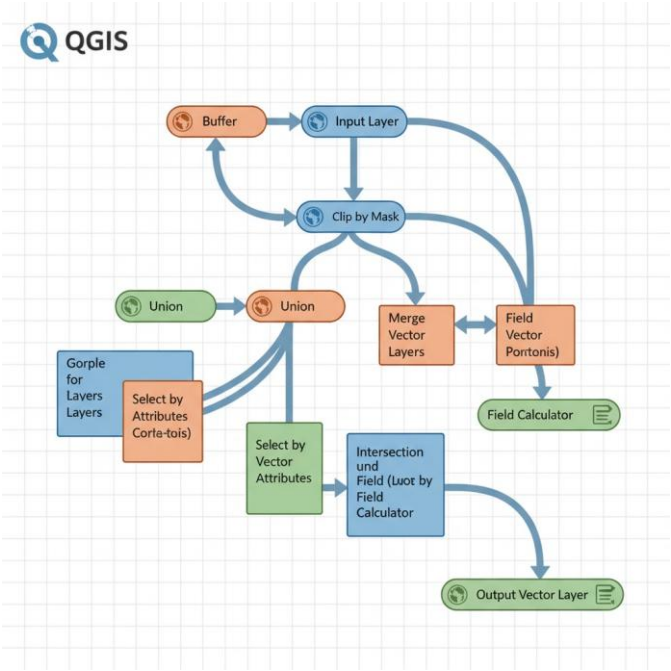
ProcessingAlgFactory 상수	대응하는 PyQGIS 클래스 (역할)	주요 역할
alg.STRING (str)	QgsProcessingParameterString	문자열 텍스트 입력
alg.INT (int)	QgsProcessingParameterNumber	정수 숫자 입력
alg.NUMBER (float 또는 alg.NUMBER)	QgsProcessingParameterNumber	실수 또는 일반 숫자 입력
alg.BOOL (bool)	QgsProcessingParameterBoolean	참/거짓(Boolean) 값 입력 (체크박스)
alg.VECTOR_LAYER	QgsProcessingParameterVectorLayer	**벡터 레이어** 입력 (Shapefile, GeoPackage 등)
alg.RASTER_LAYER	QgsProcessingParameterRasterLayer	**래스터 레이어** 입력 (GeoTIFF, DEM 등)
alg.MESH_LAYER	QgsProcessingParameterMeshLayer	**메시 레이어** 입력
alg.POINTCLOUD_LAYER	QgsProcessingParameterPointCloudLayer	**포인트 클라우드 레이어** 입력
alg.MAPLAYER	QgsProcessingParameterMapLayer	모든 유형의 **지도 레이어** 입력
alg.MULTILAYER	QgsProcessingParameterMultipleLayers	여러 개의 **레이어** 목록 입력
alg.SOURCE	QgsProcessingParameterFeatureSource	**입력 피처 소스** (일반적으로 벡터 레이어와 유사)
alg.SINK	QgsProcessingParameterFeatureSink	**출력 피처 싱크** (결과 저장 위치)
alg.FILE	QgsProcessingParameterFile	**파일 경로** 입력 (기존 파일)
alg.FILE_DEST	QgsProcessingParameterFileDestination	**출력 파일 경로** 입력 (저장할 파일)
alg.FOLDER	QgsProcessingParameterFolder	**폴더 경로** 입력 (기존 폴더)
alg.FOLDER_DEST	QgsProcessingParameterFolderDestination	**출력 폴더 경로** 입력 (저장할 폴더)
alg.EXTENT	QgsProcessingParameterExtent	**공간 범위(Extent)** 또는 경계 상자 입력
alg.CRS	QgsProcessingParameterCrs	**좌표계(CRS)** 입력
alg.DISTANCE	QgsProcessingParameterDistance	**거리** 값 입력 (단위 포함)
alg.FIELD	QgsProcessingParameterField	**입력 레이어의 필드(속성)** 선택

https://qgis.org/pyqgis/master/processing/ProcessingAlgFactory.html

PyQGIS - QGIS Python API

ProcessingAlgFactory 상수	대응하는 PyQGIS 클래스 (역할)	주요 역할
alg.BAND	QgsProcessingParameterBand	**래스터 레이어의 밴드** 선택
alg.ENUM	QgsProcessingParameterEnum	미리 정의된 목록에서 **선택(열거형)** 입력
alg.EXPRESSION	QgsProcessingParameterExpression	QGIS **표현식** 문자열 입력
alg.RANGE	QgsProcessingParameterRange	**숫자 범위** 입력 (최소값, 최대값)
alg.GEOMETRY	QgsProcessingParameterGeometry	**기하 객체 유형** 선택 (Point, Line, Polygon 등)
alg.POINT	QgsProcessingParameterPoint	**좌표 포인트** 입력
alg.COLOR	QgsProcessingParameterColor	**색상** 입력
alg.DATETIME	QgsProcessingParameterDateTime	**날짜 및 시간** 입력
alg.DATE	QgsProcessingParameterDateTime	**날짜** 입력
alg.TIME	QgsProcessingParameterDateTime	**시간** 입력
alg.LAYOUT	QgsProcessingParameterLayout	**인쇄 레이아웃** 선택
alg.LAYOUT_ITEM	QgsProcessingParameterLayoutItem	**레이아웃 항목** 선택
alg.MAP_THEME	QgsProcessingParameterMapTheme	**지도 테마** 선택
alg.SCALE	QgsProcessingParameterScale	**축척** 입력
alg.AUTH_CFG	QgsProcessingParameterAuthConfig	**인증 구성** 입력
alg.PROVIDER_CONNECTION	QgsProcessingParameterProviderConnection	**데이터 공급자 연결** 선택
alg.DATABASE_SCHEMA	QgsProcessingParameterDatabaseSchema	**데이터베이스 스키마** 선택
alg.DATABASE_TABLE	QgsProcessingParameterDatabaseTable	**데이터베이스 테이블** 선택
alg.COORDINATE_OPERATION	QgsProcessingParameterCoordinateOperation	**좌표 연산** 선택
alg.ANNOTATION_LAYER	QgsProcessingParameterAnnotationLayer	**주석 레이어** 입력

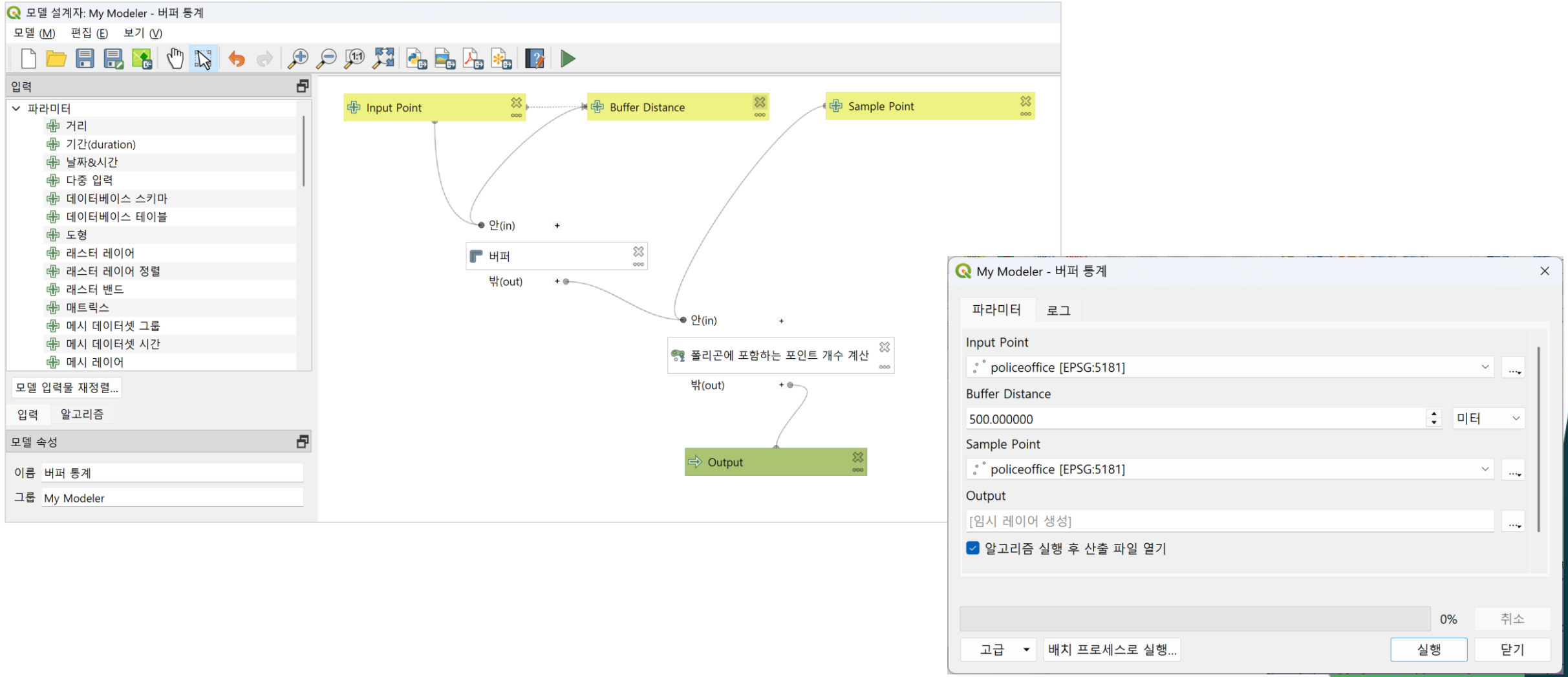
https://qgis.org/pyqgis/master/processing/ProcessingAlgFactory.html



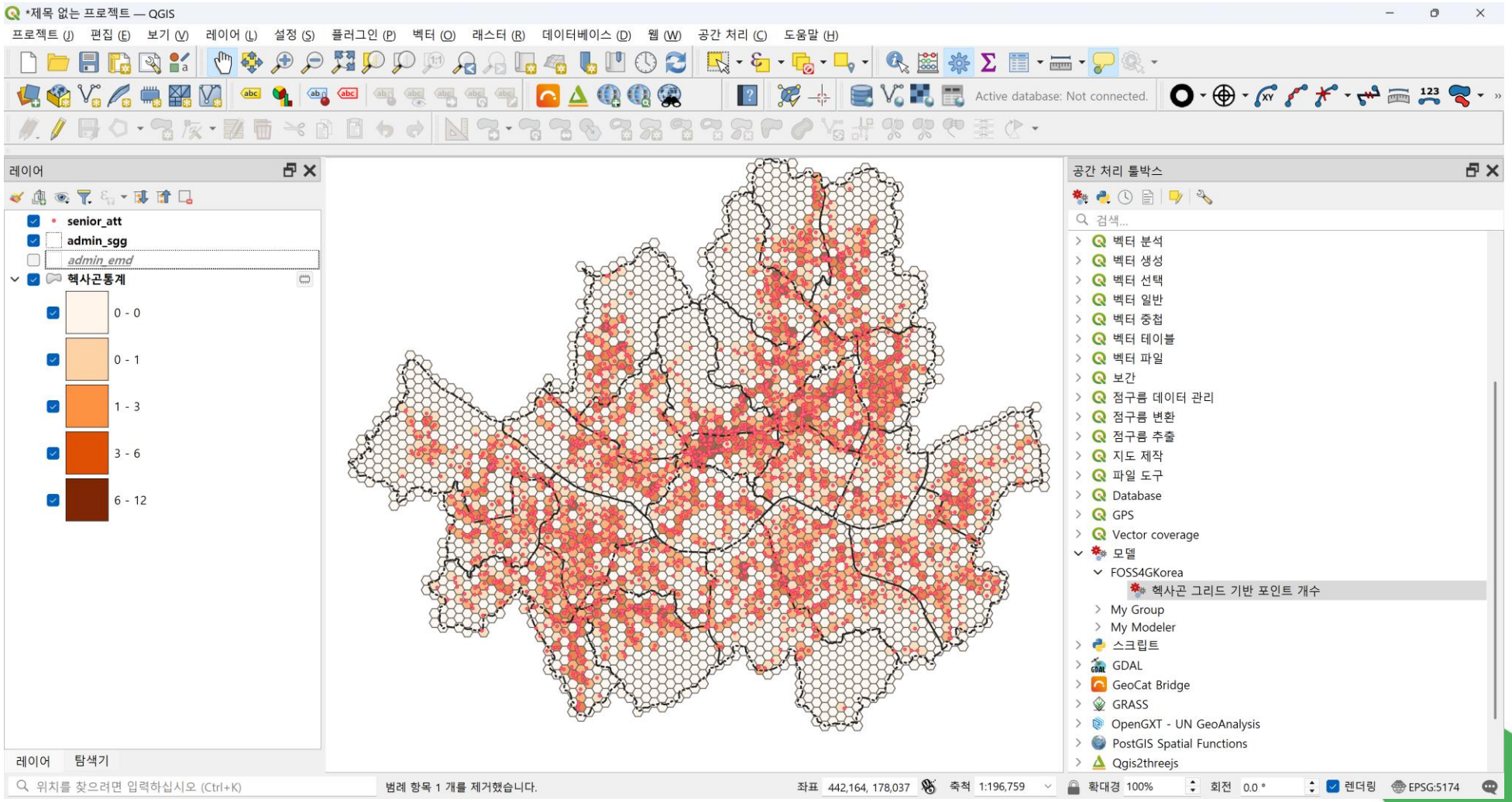
2. 모델 설계자를 이용하여 확장하기

목표: QGIS 모델 설계자를 활용하여 여러 공간처리 단계를 하나의 워크플로우(체인)로 구성하고 실행해 본다.

모델 설계자(Model Designer)



1. 헥사곤 그리드 포인트 집계



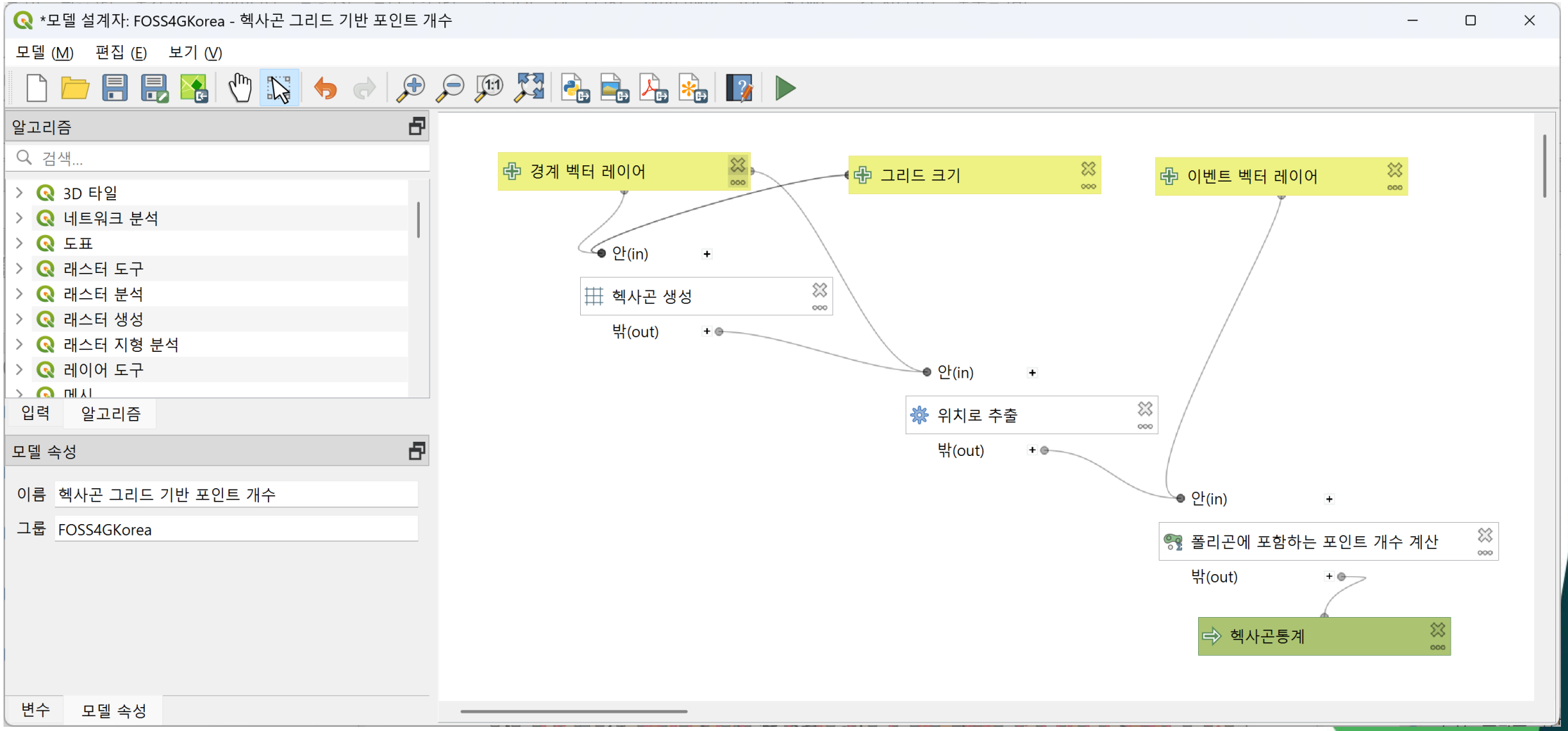
1. 핵사곤 격자 기반 포인트 개수 분석

- **목표**: 특정 경계 영역 내에서 이벤트 포인트의 공간적 밀도 또는 분포를 시각적으로 분석하고 정량화 → **파라미터**와 **알고리즘** 선별

파라미터	유형	변수 설명
1. 경계 영역 폴리곤	벡터 레이어(폴리곤)	행정구역 경계 등 격자가 생성될 영역 폴리곤 레이어
2. 격자 크기	숫자(정수형)	격자의 크기 (단위는 프로젝트 좌표계 단위 기본)
3. 이벤트 포인트	벡터 레이어(포인트)	개수 분석 대상 포인트 레이어

단계	QGIS 알고리즘	역할
1. 그리드 생성	native:creategrid	지정된 **경계 영역**을 기준으로 사용자가 정의한 크기(500m 등)의 정육각형 격자(Hexagon Grid)를 생성
2. 경계 추출	native:extractbylocation	생성된 전체 헥사곤 그리드 중, 분석 대상인 **경계 벡터 레이어**와 실제로 교차(Intersect)하는 격자만을 추출하여 불필요한 영역을 제거
3. 통계 계산	native:countpointsinpolygon	추출된 각 헥사곤 폴리곤 내부에 포함(Contain)되는 **이벤트 포인트**의 **총 개수**를 계산하고, 이 개수를 새로운 속성 필드(**NUMPOINTS**)로 격자에 추가

1. 헥사곤 그리드 기반 포인트 개수 분석



2. 실습 예제

- 백화점 및 마트로부터 반경 500m 이내에 있는 지하철 역 개수 구하기

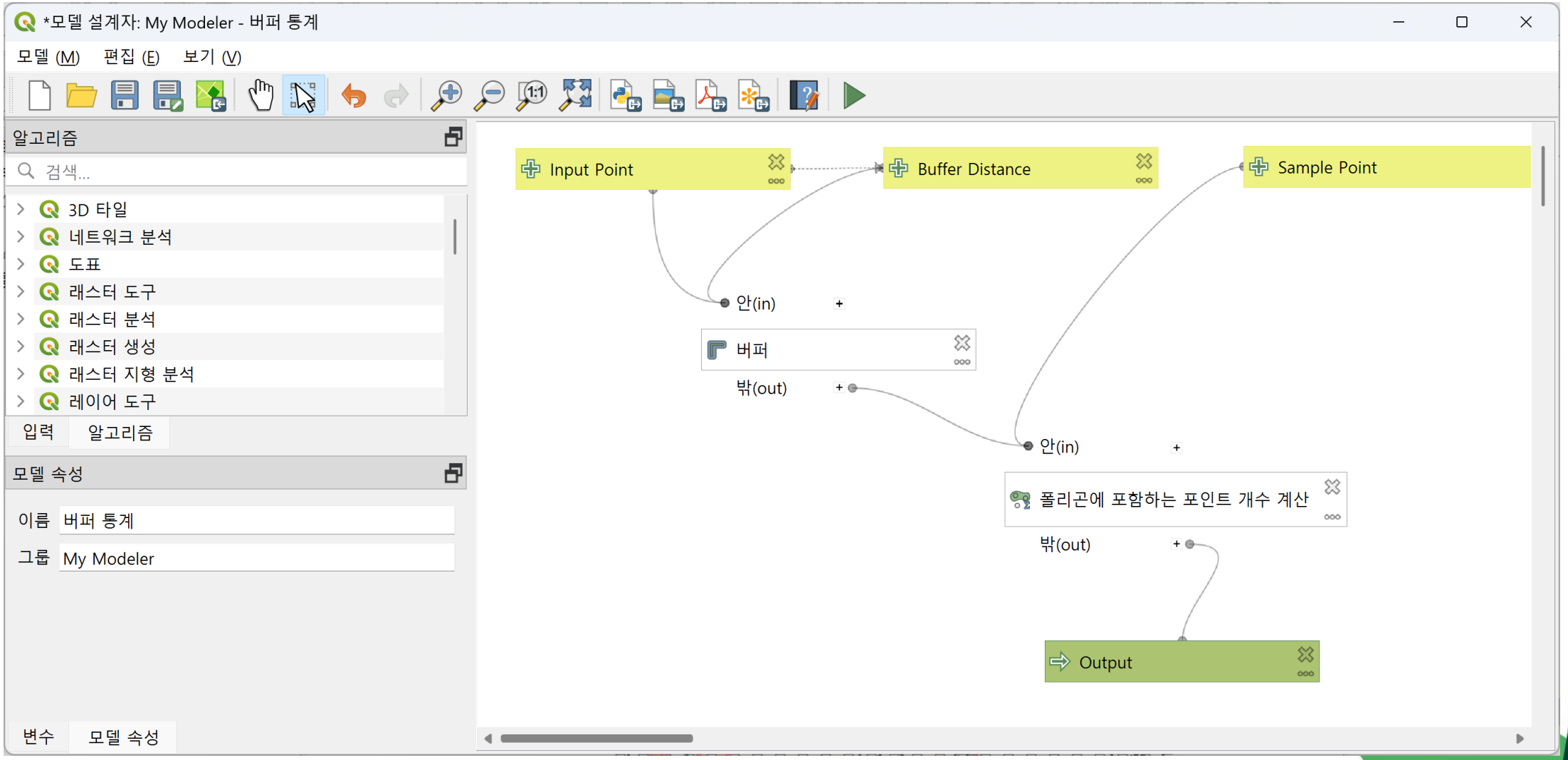
  → A포인트 레이어에서 500m 버퍼를 설정한 범위 이내에 속해 있는 B포인트 레이어의 포인트 개수를 계산

  → 입력된 input points에서 사용자 버퍼 거리만큼 버퍼한 폴리곤 내에 포함된 sample points의 개수를 계산하여 폴리곤 레이어 반환

  → 파라미터: 기준 포인트, 버퍼 거리, 대상 포인트

  → 알고리즘: 버퍼, 폴리곤에 포함하는 포인트 개수

2. 실습 예제

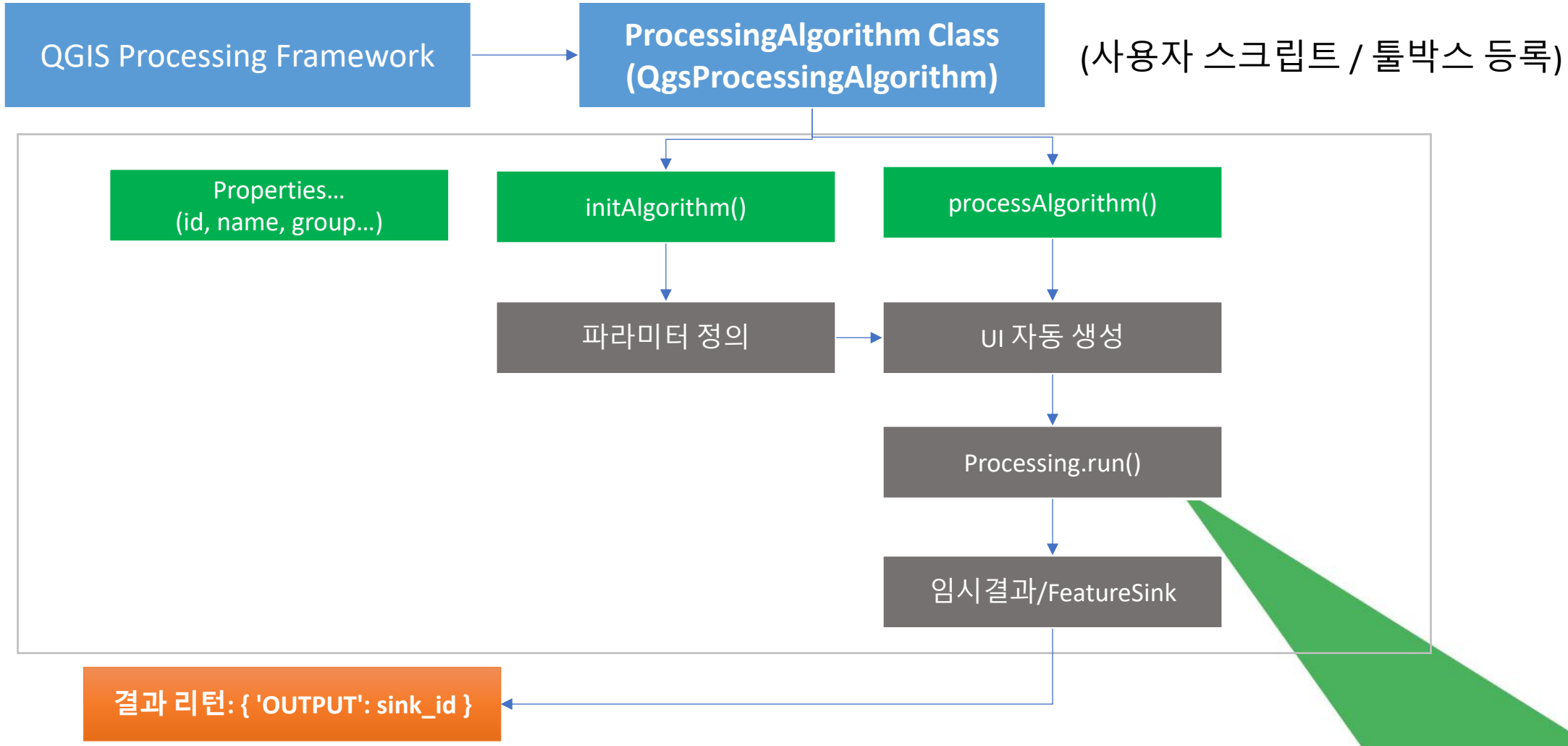

*모델 설계자: My Modeler - 버퍼 통계
모델 (M)
편집 (E)
보기 (V)
알고리즘
검색...
3D 타일
네트워크 분석
도표
래스터 도구
래스터 분석
래스터 생성
래스터 지형 분석
레이어 도구
입력
알고리즘
모델 속성
이름
버퍼 통계
그룹
My Modeler
변수
모델 속성
Input Point
Buffer Distance
Sample Point
안(in)
버퍼
밖(out)
안(in)
폴리곤에 포함하는 포인트 개수 계산
밖(out)
Output

ChatGPT Claude Gemini Grok

3. GPT 활용하여 툴박스 스크립트 작성

목표: 자연어 기반 AI 모델을 활용해 QGIS 툴박스용 Python 스크립트를 자동으로 생성하고, 이를 수정·실행해 본다.

Processing Framework 구조



PyQGIS - QGIS Python API

ProcessingAlgFactory 상수	대응하는 PyQGIS 클래스 (역할)	주요 역할
alg.STRING (str)	QgsProcessingParameterString	문자열 텍스트 입력
alg.INT (int)	QgsProcessingParameterNumber	정수 숫자 입력
alg.NUMBER (float 또는 alg.NUMBER)	QgsProcessingParameterNumber	실수 또는 일반 숫자 입력
alg.BOOL (bool)	QgsProcessingParameterBoolean	참/거짓(Boolean) 값 입력 (체크박스)
alg.VECTOR_LAYER	QgsProcessingParameterVectorLayer	**벡터 레이어** 입력 (Shapefile, GeoPackage 등)
alg.RASTER_LAYER	QgsProcessingParameterRasterLayer	**래스터 레이어** 입력 (GeoTIFF, DEM 등)
alg.MESH_LAYER	QgsProcessingParameterMeshLayer	**메시 레이어** 입력
alg.POINTCLOUD_LAYER	QgsProcessingParameterPointCloudLayer	**포인트 클라우드 레이어** 입력
alg.MAPLAYER	QgsProcessingParameterMapLayer	모든 유형의 **지도 레이어** 입력
alg.MULTILAYER	QgsProcessingParameterMultipleLayers	여러 개의 **레이어** 목록 입력
alg.SOURCE	QgsProcessingParameterFeatureSource	**입력 피처 소스** (일반적으로 벡터 레이어와 유사)
alg.SINK	QgsProcessingParameterFeatureSink	**출력 피처 싱크** (결과 저장 위치)
alg.FILE	QgsProcessingParameterFile	**파일 경로** 입력 (기존 파일)
alg.FILE_DEST	QgsProcessingParameterFileDestination	**출력 파일 경로** 입력 (저장할 파일)
alg.FOLDER	QgsProcessingParameterFolder	**폴더 경로** 입력 (기존 폴더)
alg.FOLDER_DEST	QgsProcessingParameterFolderDestination	**출력 폴더 경로** 입력 (저장할 폴더)
alg.EXTENT	QgsProcessingParameterExtent	**공간 범위(Extent)** 또는 경계 상자 입력
alg.CRS	QgsProcessingParameterCrs	**좌표계(CRS)** 입력
alg.DISTANCE	QgsProcessingParameterDistance	**거리** 값 입력 (단위 포함)
alg.FIELD	QgsProcessingParameterField	**입력 레이어의 필드(속성)** 선택

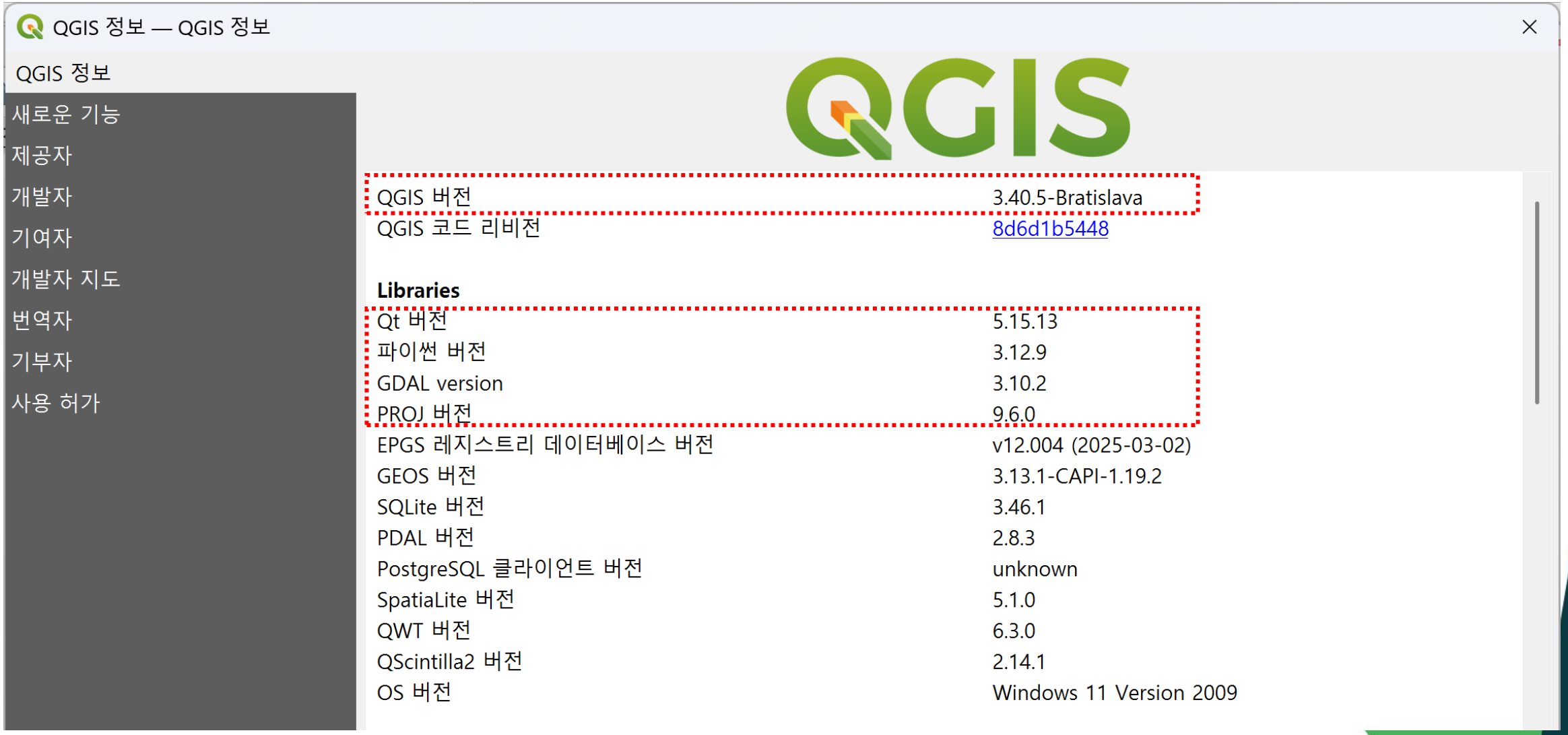
https://qgis.org/pyqgis/master/processing/ProcessingAlgFactory.html

PyQGIS - QGIS Python API

ProcessingAlgFactory 상수	대응하는 PyQGIS 클래스 (역할)	주요 역할
alg.BAND	QgsProcessingParameterBand	**래스터 레이어의 밴드** 선택
alg.ENUM	QgsProcessingParameterEnum	미리 정의된 목록에서 **선택(열거형)** 입력
alg.EXPRESSION	QgsProcessingParameterExpression	QGIS **표현식** 문자열 입력
alg.RANGE	QgsProcessingParameterRange	**숫자 범위** 입력 (최소값, 최대값)
alg.GEOMETRY	QgsProcessingParameterGeometry	**기하 객체 유형** 선택 (Point, Line, Polygon 등)
alg.POINT	QgsProcessingParameterPoint	**좌표 포인트** 입력
alg.COLOR	QgsProcessingParameterColor	**색상** 입력
alg.DATETIME	QgsProcessingParameterDateTime	**날짜 및 시간** 입력
alg.DATE	QgsProcessingParameterDateTime	**날짜** 입력
alg.TIME	QgsProcessingParameterDateTime	**시간** 입력
alg.LAYOUT	QgsProcessingParameterLayout	**인쇄 레이아웃** 선택
alg.LAYOUT_ITEM	QgsProcessingParameterLayoutItem	**레이아웃 항목** 선택
alg.MAP_THEME	QgsProcessingParameterMapTheme	**지도 테마** 선택
alg.SCALE	QgsProcessingParameterScale	**축척** 입력
alg.AUTH_CFG	QgsProcessingParameterAuthConfig	**인증 구성** 입력
alg.PROVIDER_CONNECTION	QgsProcessingParameterProviderConnection	**데이터 공급자 연결** 선택
alg.DATABASE_SCHEMA	QgsProcessingParameterDatabaseSchema	**데이터베이스 스키마** 선택
alg.DATABASE_TABLE	QgsProcessingParameterDatabaseTable	**데이터베이스 테이블** 선택
alg.COORDINATE_OPERATION	QgsProcessingParameterCoordinateOperation	**좌표 연산** 선택
alg.ANNOTATION_LAYER	QgsProcessingParameterAnnotationLayer	**주석 레이어** 입력

https://qgis.org/pyqgis/master/processing/ProcessingAlgFactory.html

PyQGIS - QGIS Python API



ProcessingAlgorithm 핵심 체크

■ 환경
- QGIS & Python 버전 명시 (예: QGIS 3.34 / Py 3.10)
- Processing Toolbox용 코드인지 구분

■ 핵심 API
- Geometry Type: Point/Line/Polygon 검증 필수
- CRS/단위: 미터 기반 투영좌표계로 처리
- Feature = Geometry + Attributes

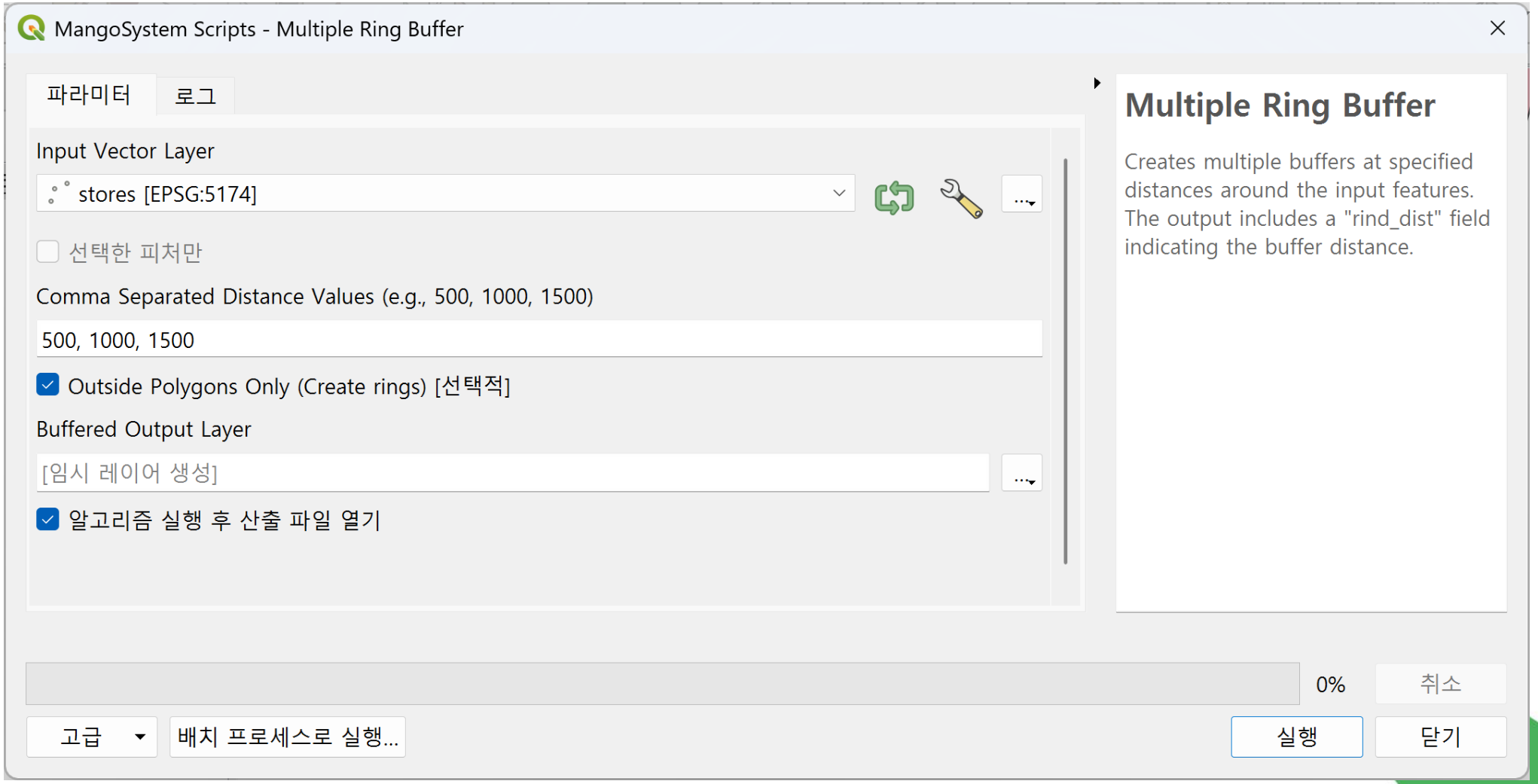
■ Processing 구조
- 프로세스 메타데이터(name, group)
- initAlgorithm(): 파라미터 정의
- processAlgorithm(): 처리 + FeatureSink
- createInstance(): 필수

■ 입출력 설계
- 레이어 타입/필드 타입 정확히 지정
- 출력: FeatureSink 또는 직접 스타일 변경

■ 프롬프트 필수 요소
- 목표 한 줄 요약
- 파라미터 목록 상세
- 지원 Geometry Type/CRS 명시
- 로그(feedback) & 예외 처리 요구
- 코드만 출력 요청

■ 검증 포인트
- 파라미터 UI 정상
- Geometry/CRS 오류 처리됨
- 결과가 지도에 바로 반영

실습 1. Multiple Ring Buffer



실습 1. Multiple Ring Buffer

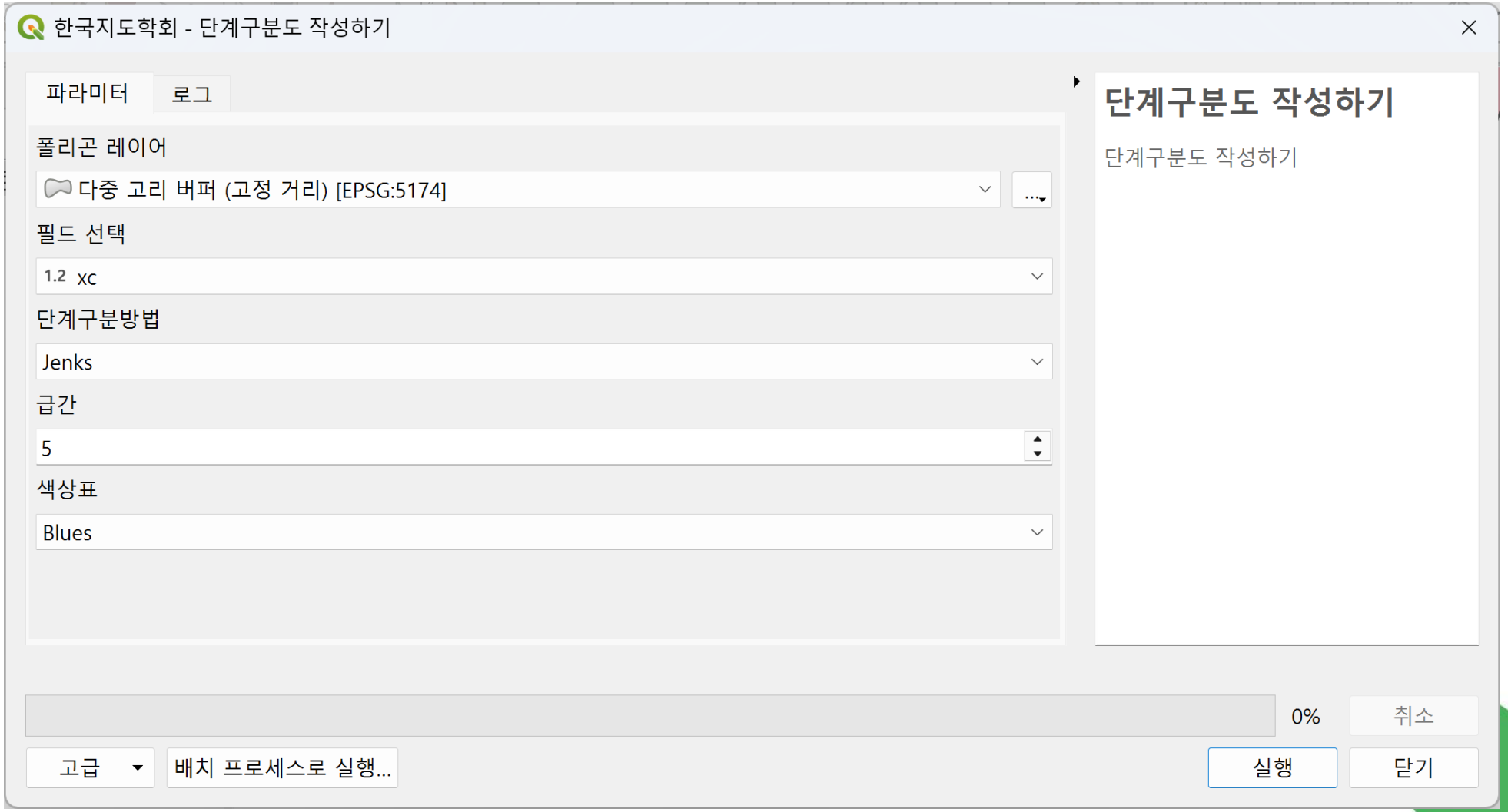
저는 QGIS 3.x Processing 툴박스에서 사용할 PyQGIS 3 기반의 사용자 정의 알고리즘을 작성하려고 합니다.
알고리즘의 목표는 입력 벡터 피처를 중심으로 쉼표로 구분된 여러 거리에 대한 다중 링 버퍼를 생성하는 것입니다.
클래스 이름은 MultipleRingBufferAlgorithm으로 하고, 모든 코드는 4개의 공백으로 들여쓰고 상세한 주석을 포함해 주세요.

1. 알고리즘 기본 정보 및 그룹 설정
 - 고유 이름 (name): multiple_ring_buffer
 - 표시 이름 (displayName): Multiple Ring Buffer
 - 그룹 이름 (group): MangoSystem Scripts
 - 그룹 ID (groupId): mangoscripts
 - 도움말 (shortHelpString): 생성된 버퍼 거리가 포함된 'rind_dist' 필드가 출력에 포함됨을 명시합니다.

2. 입력 및 출력 파라미터 정의 (initAlgorithm)
알고리즘은 다음 4가지 파라미터를 정의해야 합니다.
 - INPUT (입력 벡터 레이어): QgsProcessingParameterFeatureSource (벡터 타입만 허용)
 - DISTANCES (거리 값): QgsProcessingParameterString.
  - 역할: 쉼표로 구분된 문자열 (예: '500, 1000, 1500').
  - 기본값: '500, 1000, 1500'
 - OUTSIDE (링 옵션): QgsProcessingParameterBoolean.
  - 역할: True로 설정되면 인접 버퍼 간의 차이(difference())를 계산하여 링 모양의 폴리곤만 생성합니다.
  - 기본값: True (선택 사항)
 - OUTPUT (출력 싱크): QgsProcessingParameterFeatureSink (폴리곤 벡터 타입 지정)

3. 핵심 처리 로직 (processAlgorithm)
처리 로직은 다음과 같은 순서와 세부 규칙을 따라야 합니다.
 - 입력 유효성 검사: 입력 레이어 (INPUT)가 유효한지 확인하고, DISTANCES 문자열을 파싱하여 유효한 float 값 리스트를 얻습니다. 값이 없거나 숫자가 아닌 경우 QgsProcessingException을 발생시켜야 합니다.
 - 출력 스키마 정의: 입력 레이어의 모든 속성을 복사하고, 버퍼 거리를 저장할 rind_dist 필드 (QVariant.Double, 길이 10, 정밀도 4)를 출력 필드에 추가합니다.
 - 피처 반복: 입력 레이어의 각 피처를 반복하면서 버퍼를 생성합니다.
 - 버퍼 및 링 생성:
  - 각 거리 값 (rind_distance)에 대해 geometry.buffer(rind_distance, 24)를 호출하여 버퍼를 생성합니다.
  - OUTSIDE 옵션이 **True**이고 첫 번째 버퍼가 아닌 경우, 현재 버퍼 지오메트리에서 직전 버퍼 지오메트리를 뺀 **차이(difference())**를 계산하여 링(Ring) 모양의 폴리곤을 생성합니다.
 - 피처 출력: 각 단계에서 생성된 지오메트리를 가진 새로운 피처에 입력 피처의 속성과 rind_distance 값을 속성으로 설정하여 QgsFeatureSink에 추가합니다.
 - 진행률 보고: 반복문 내에서 feedback.setProgress()를 사용하여 진행률을 업데이트합니다.

실습 2. 단계구분도


한국지도학회 - 단계구분도 작성하기
파라미터
로그
폴리곤 레이어
다중 고리 버퍼 (고정 거리) [EPSG:5174]
필드 선택
1.2 xc
단계구분방법
Jenks
급간
5
색상표
Blues
단계구분도 작성하기
단계구분도 작성하기
단계구분도 작성하기
0%
취소
고급
배치 프로세스로 실행...
실행
닫기

실습 2. 단계구분도

너는 QGIS 3.34+ 환경에서 동작하는 Processing Toolbox 플러그인 스크립트를 작성하는 GIS 개발자야.
QGIS 3.34 이상에서 사용할 수 있는 Processing Toolbox용 Python 스크립트를 만들어줘.

목표:
선택한 폴리곤 레이어와 숫자 속성 필드를 사용해 단계구분도(Graduated Color Renderer)를 자동으로 적용하는 도구를 만들고 싶어.

요구 기능:
1) 입력 파라미터
  - 폴리곤 레이어 선택
  - 숫자 필드 선택
  - 분류 방법 선택 (예: Equal Interval, Quantile, Jenks)
  - 급간 수 (예: 기본 5)
  - 색상표 선택 (QGIS 기본 색상표 사용)

2) 처리 내용
  - 사용자가 지정한 설정에 맞게 분류하고 색상을 적용
  - 레이어 스타일 변경 후 지도에 반영

3) 클래스 기반 QGIS ProcessingAlgorithm 형식으로 만들어줘.
  - name(), displayName(), group(), initAlgorithm(), processAlgorithm() 필요
  - processing.run() 대신 QgsGraduatedSymbolRenderer를 사용해 직접 스타일 설정

4) 출력
  - 입력 레이어에 적용된 스타일이 바로 보이도록 해줘
  - 실행 후 메시지 출력

5) 주석은 초보자도 이해할 수 있게 친절히 넣어줘.

완성된 코드만 출력해줘.

감사합니다!

Welcome to OSGeo Korean Chapter

OSGeo
한국어지부
Korean Chapter

함께 성장하는 새로운 방법,
오픈 소스 소프트웨어!!