1. 매개변수

- prdInfo: list → 4개의 값 (공장코드, 라인코드, 품목코드, 근무자번호)

- prdTime: datetime → 첫 제품 생산 시간 (예) 2021-09-17 13:51:57

- plan\_qty → 생산 계획수량 (plan\_qty, 생산지시 테이블 참고)

- specs: list → 8개 측정값에 대한 품목규격 값 (제품마다 달라짐, 생산정보 테이블 참고)

- tols: list → 8개 측정값에 대한 품목규격 허용치 (생산정보 테이블 참고)

2. 아래 함수에서 생성해야 할 데이터

- plan\_qty 개의 데이터 생성

- 생산시간 → 다음 제품 생산 시간은 앞 제품 생산 시간에 1분 더하기

- 제품일련번호 → MES 시나리오 - 생산 및 품질검사.docx 참고

- specs와 tols를 바탕으로 한 제품 측정값 8개 (np.random.normal 이용)

- 8개 측정값을 바탕으로 한 가로직진도, 세로직진도, 홀직경, 홀비율

3. 아래 함수에서 최종 실행해야 할 내용

- 생산정보 테이블의 각 항목에 해당하는 내용을 **productionData.csv** 파일에 한 줄씩 쓰기

(예) 공장코드,라인코드,품목코드,근무자번호,제품일련번호,가로길이,세로길이,가로면세로편차,세로면가로편차,홀가로길이,홀세로길이,홀가로중심,홀세로중심,가로직진도,세로직진도,홀직경,홀비율,생산시간 → csv 파일에 header로 들어감

**1,2,1,7,20210909\_080510\_1\_2,40.05,29.96,0.5,0.2,5.03,4.98,0.01,-0.02,xx,yy,pp,qq,2021-09-09 08:05:10**

1,2,1,7, ...........................

여기서 xx, yy, pp, qq는 앞에 있는 데이터에서 계산된 값 → 생산정보 테이블 참고

4. 함수 정의

**- def simProductionData(prdInfo, prdTime, plan\_qty, specs, tols):**

5. 추가 작업 내용

**- 위에서 생성된 데이터를 판별하여 품질검사정보를 생성하는 파이썬 함수 작성 (csv 쓰기 포함)**

**- 생성된 생산정보 및 품질검사정보 데이터를 DB(생산정보 테이블)에 업로드하는 코드 작성**

1. 파일명: generateData.py

2. 함수명: simulateData

def: simulateData(prdInfo, prdTime, plan\_qty, specs, tols):

# (1) Production 데이터 생성

prdInfoData = np.tile(np.array(prdInfo), (plan\_qty, 1)) → prdInfo column

prd\_dtData = ... → prd\_dt column

serial\_noData = ... → serial\_no column

dim\_x = ...

dim\_y = ...

...

hole\_xc = ...

hole\_yc = ...

measuredData = ... → 8개의 측정값 column을 하나로 합친 것

str\_x = ...

str\_y = ...

hole\_d = ...

hole\_ratio = ...

computedData = ... → 4개의 계산값 column을 하나로 합친 것

productionData = ... → prdInfoData, serial\_noData, measuredData, computedData, prd\_dtData column 전체를 합친 것

np.savetxt('productionData.csv', productionData, delimiter=",",

header="plant\_cd, line\_cd, item\_cd, worker\_no, serial\_no,\

dim\_x, dim\_y, dim\_h, dim\_w, hole\_x, hole\_y, hole\_xc, hole\_yc,\

str\_x, str\_y, hole\_d, hole\_ratio, prd\_dt")

# (2) Quality 데이터 생성

# prdInfo[3]의 값(근무자번호)에 1을 더해 prdInfo를 업데이트 하는 과정 필요

prdInfoData = ... → prdInfo column

checkedData = ... → 9개의 검사 결과 column을 합친 것, 각 필드의 값은 boolean (0: 불량 or 1:양품)

qualityData = ... -> prdInfoDat, serial\_noData, checkedData column 전체를 합친 것

np.savetxt('qualityData.csv', qualityData, delimiter=",", ... ))

# (3) productionData와 qualityData를 MariaDB 아래 mes 데이터베이스의 production과 quality 테이블에 각각 업로드