

## Komponent 9 - ToCSVConverter

### 1. Charakterystyka problemu

Klient potrzebuje komponentu umożliwiającego zapisanie danych telemetrycznych na dysku w formacie csv.

### 2. Wizja produktu

Komponent powinien umożliwiać wyeksportowanie zbioru danych wejściowych do formatu csv. Format danych wejściowych powinien być uniwersalny.

### 3. Zakres funkcjonalności

#### 3.1. Wymagania funkcjonalne

- Odbieranie zbioru danych typu (IdentyfikatorZbioru, ListaDanych, NazwaParametru)
- Odbieranie zbioru danych typu (IdentyfikatorZbioru, ListaListDanych, ListaNazwParametrów)
- Konwertowanie danych do formatu CSV
- Tworzenie wymaganej do zapisu struktury katalogów
- Dopisywanie nowych danych do istniejącego już zbioru

#### 3.2. Wymagania niefunkcjonalne

- Obsługa sytuacji, w której wprowadzono niespójne dane
- Obsługa sytuacji, w której nie jest możliwe zapisanie przetworzonych danych na dysku

### 4. Przykład wygenerowanego pliku

- SENSOR\_TEMP\_PRES\_HUMD\_withTimeStamp.csv

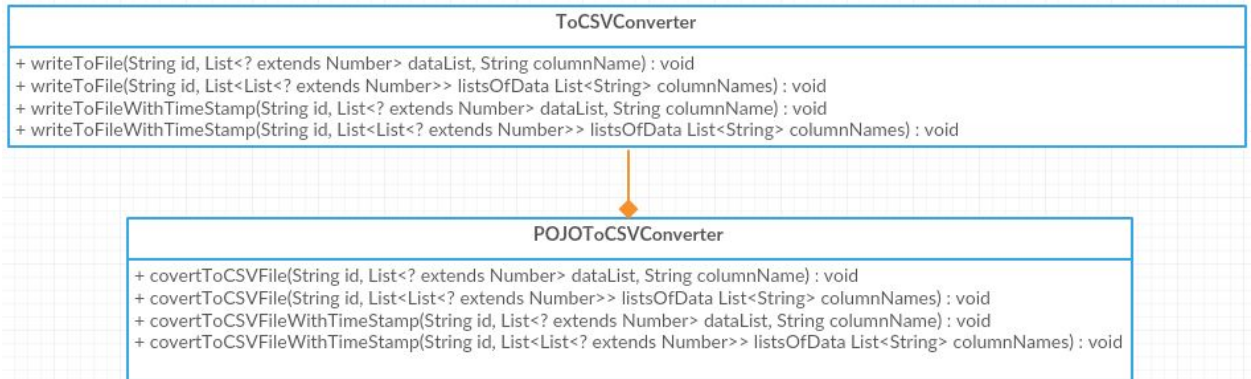
```
1 Temperature,Pressure,Humidity
2 22,25,29 #2016.05.25 23:00:06
3 1010,1100,1040 #2016.05.25 23:00:06
4 50,60,40 #2016.05.25 23:00:06
5
```

### 5. Architektura komponentu

#### 5.1. Opis architektury

Klasa ToCSVConverter jest klasą agregującą konwertery różnych formatów wejściowych, umożliwia w przezroczysty dla użytkownika sposób korzystać z konwertera. Klasa POJOToCSVConverter jest implementacją konwertera obiektów jadowych do formatu CSV.

## 5.2. Schemat



## 6. Przykład użycia komponentu

- Utworzenie danych wejściowych i zapisanie ich w formacie csv

```
List<String> columnNames = new LinkedList<>();
columnNames.add("Temperature");
columnNames.add("Pressure");
columnNames.add("Humidity");

List<Integer> temperature = new LinkedList<>();
temperature.add(22);
temperature.add(25);
temperature.add(29);
List<Integer> pressure = new LinkedList<>();
pressure.add(1010);
pressure.add(1100);
pressure.add(1040);
List<Integer> humidity = new LinkedList<>();
humidity.add(50);
humidity.add(60);
humidity.add(40);

List<List<? extends Number>> listsOfData = new LinkedList<>();
listsOfData.add(temperature);
listsOfData.add(pressure);
listsOfData.add(humidity);

//Result file location: *\\UserDir\\Documents\\CSVData\\SENSOR_TEMP_ONLY
ToCSVConverter.writeToFile("SENSOR_TEMP_ONLY", temperature, "Temperature");
//Columns: Temperature, Pressure, Humidity
//Result file location: *\\UserDir\\Documents\\CSVData\\SENSOR_3PARAMS
ToCSVConverter.writeToFile("SENSOR_3PARAMS", listsOfData, columnNames);
ToCSVConverter.writeToFileWithTimeStamp("SENSOR_TEMP_ONLY_withTimeStamp", temperature, "Temperature");
ToCSVConverter.writeToFileWithTimeStamp("SENSOR_3PARAMS_withTimeStamp", listsOfData, columnNames);
```