

# レッスン①…リファクタリングの過程を一通り体験する

中林 智之

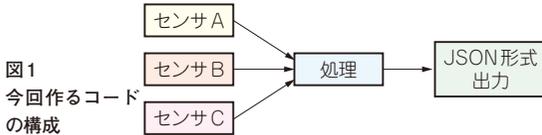


表1 サンプル・コードの仕様  
(センサ名とインターバル)

センサ名	インターバル
A	1秒ごと
B	2秒ごと
C	4秒ごと

表2 サンプル・コードに含む関数と説明

関数名	説明
main	プログラム動作確認用の最低限の機能を実装。JSONを標準出力にログ出力する
get_sensor_A_value	実行用のスタブ
get_sensor_B_value	実行用のスタブ
get_sensor_C_value	実行用のスタブ
get_sensors_report	リファクタリングの対象。センサから値を取得してJSONをバッファに出力する

リスト1 サンプル・コードのJSON出力例

```
{ "sensors": [ { "name": "A", "value": 154.25 }, { "name": "B", "value": 567.80 }, { "name": "C", "value": 3555.08 } ] }
```

リスト3 サンプル・コードの実行結果

```
{ "sensors": [ { "name": "A", "value": 154.25 } ] } ← 1行が1秒ごとのJSON出力になっている
{ "sensors": [ { "name": "A", "value": 154.25 }, { "name": "B", "value": 567.80 } ] }
{ "sensors": [ { "name": "A", "value": 154.25 } ] }
{ "sensors": [ { "name": "A", "value": 154.25 }, { "name": "B", "value": 567.80 }, { "name": "C", "value": 3555.08 } ] }
```

IoTデバイスのサンプル・コードを例にリファクタリングの流れを体験します。

解説に使用するサンプル・コードは、リファクタリングの一連の流れの雰囲気がつかめるよう、短いものを用意しました。

## リファクタリング対象のコード… 典型的なIoTデバイス

リファクタリングの対象とするサンプル・コードとその仕様を説明します。

### ● サンプル・コードの仕様

センサから値を取得して、外部システムと連携するためのJSONを出力するC言語のサンプル・コードです(図1)。

リスト1に示すようなJSONを出力します。センサはA, B, Cの3つあり、その値は全て浮動小数点数

です。また、それぞれのセンサからは表1のインターバル(時間間隔)で値を取得します。JSONは1秒ごとに出力します。

### ● リファクタリング前のサンプル・コード

リスト2(次頁)はリファクタリング前のサンプル・コードです。表2にこのコードに含まれている各関数の簡単な説明を示します。リファクタリングの対象はセンサ取得関数である、get\_sensors\_report()です。

この関数は各センサ・ドライバのソースコードに記述してある想定ですが、ここでは固定値を返すスタブで代用します。メイン関数もプログラムを実行するための最低限のものとなっています。

サンプル・コードを実行すると、出力はリスト3のようになります。1行が1秒ごとのJSONを出力した結果になっています。