

令和元年（ワ）第10940号 損害賠償請求事件

原告 森次茂廣

被告 株式会社

第2準備書面

令和2年7月17日

大阪地方裁判所第26民事部合議係 御中

原告訴訟代理人弁護士

同

同

(担当) 同

第1 原告のプログラム開発方針

1 総論

プログラムの出力結果が同じでも、プログラムコードの表現方法は作者の考え方によって様々である。

例えば、処理の順序や組み合わせに関しては、標準的な記述方法、速度重視の記述方法、最適化重視の記述方法、解読困難な記述方法などの方法が考えられる。また、表現方法に関しては、データ構造（グループ化、リレーション化など）、データの命名方法（代表的なものとしてハンガリアン記法、キャメル記法、スネーク記法）など作者によって様々である。

2 原告プログラムの特徴

原告は、フリーランスのエンジニアで、個人で設計・プログラミング、デバッグ、メンテナンスを行っている。そのため、グループでの開発を前提とするコードの記述（他人に理解しやすく、引き継ぎしやすい記載）と異なり、自らが理解できて工数を抑えることができる記述を行っている。具体的には次のとおりである。

処理順序や組み合わせについては、標準的な記述というよりは自らが理解できる記述、最適化及びデータの構造化に重きを置いた記述を行っている。

データ等の命名については、独自の識別子を使用したハンガリー記法とキャメル記法を混合して命名しているほか、変数名、メソッド名などに日本語（漢字）を多用している。

3 アナログ入力ボードのモジュール

アナログ入力ボード（Analog-to-digital converter, 以下、「ADC」という。）とは、センサーが出力するアナログ信号である電圧信号をコンピュータで処理するためのデジタル値に変換する装置であるところ、原告は、ADCを使ったシステム開発を得意としており、多くの実績とノウハウを有している。本訴訟における本件プログラム3～6でも、ADCを主要モジュールとして使用している。

なお、コンテック社のADC上でも、トリガ機能、リングメモリは搭載されているが、使い勝手が良くないため、原告が独自にプログラムを創作し、高速サンプリング、連続サンプリング、トリガ収録、プリトリガ収録を実現している。つまり、「他者が創作したサンプルプログラムの利用」ではない。

第2 本件プログラム1の著作物性

1 被告からの提案書

原告は、被告から本件プログラム1についての提案書（甲22）を受けて、被告の提案に沿うプログラムを作成しているところ、被告の提案を実現する方法として以下に述べるように複数の選択肢が考えられた。原告は、その選択肢の中から一つを選択して本件プログラム1を作成している。

2 被告の提案を受けて原告が考えた本件プログラム1の概要

(1) 管理するレコーダーの数

10台に決定。

(2) アプリケーションの形態

管理する最大10台のレコーダーをアプリケーションからどのように管理するかの方法としては、「複数インスタンスのアプリ」、「単一インスタンスのアプリでマルチフォーム」、「単一インスタンスのアプリでシングルフォーム」の3つの方法が考えられた。原告は、1本のアプリですべてのレコーダーを管理し、個々のフレームにレコーダーを表示する「単一インスタンスのアプリでシングルフォーム」の方法を採用した。

(3) レコーダーとの通信仕様

通信方法としては、HTTP通信とModbus通信があるところ、リアルタイムに情報を更新できるModbus通信を選択した。もっとも、シンプルなModbus通信を使用しても、送信要求・結果受信を繰り返すのみでは送信と受信の間で待ち時間が発生し高速処理が望めないため、処理を細分化し、処理の効率化を行っている。

(4) 収録データの管理方法

被告の提案書5ページには「テキストファイル」の記載があるが、アクセス性を考慮し、データベースソフトを採用した。データ

ベースソフトとして、Oracle（オラクル）、MySQL（マイエスキューエル）、ACCESS（アクセス）が考えられたが、コストと操作性を考慮してACCESSを採用した。

ただし、ACCESSでは、ファイル容量上限が2GBとなっているため、収録データを1ファイルに収められない可能性があった。そこで、ファイル分割（基本情報テーブルとデータテーブルを分ける）、収録インターバルの可変コントロール機能を原告の設計により追加している。

（5）画面表示

基本的にはレコーダーの画面表示に合わせるが、統合管理画面ではフォームのサイズを変更するとそれに合わせて表示内容も変わる仕様になっている。バックカラーによりデータが見えづらくなる場合には反転色に近い色で描画される。

グラフ描画は横方向に描画される仕様とし、描画方法についてはパネルオブジェクトで実現している。

3 ソースコードの構造

原告は、上記の仕様を実現するためのソースコードを作成した（甲3）。

本件プログラム1のデータ構造は以下の表のとおりとなっているところ、それぞれの構造がどのように関連しているかどうかは、別紙1及び別紙2を参照されたい。

なお、本件プログラム1は、大まかに分類すると、各レコーダーのステータスを統合管理画面に表示させる機能（別紙1の「1. 統合管理画面」参照）、個々のレコーダーの詳細なステータスをレコーダー監視画面に表示させる機能（別紙1の「2. レコーダー監視画面」参照）、高速通信を可能にする機能（別紙1の「3. レコー

「データ通信ルーチン」参照)、収録データを適切に管理する機能(別紙1の「4. データベース設計について」参照)に分かれているところ、別紙1は上記の項目ごとにソースコードがどのように関連しているかを説明したもの、別紙2は既に提出済みの甲3号証にコメントを追加したものである。

clsAccessMdb	データベース操作クラス
clsAddressInfo	メールアドレス管理クラス
clsArrange	データ表示位置クラス(数値パネル)
clsChannelInfo	チャンネル情報クラス
clsExcel	エクセルアプリケーション操作クラス
clsGraphLine	ライングラフクラス(Graphic系)
clsMeasureData	マンロックデータクラス(Base class)
clsRecorderInfo	レコーダークラス(child class)
clsSendMail	メール送信クラス
frmMain	メインフォーム画面
frmPaperlessRecorder	レコーダークラス画面
frmPropertyMain	メインプロパティフォーム

4 結論

本件プログラム1のソースコードには、原告の開発方針が散りばめられている(上記第1参照)ほか、被告の提案を実現する複数の方法から一つを選択し、具体的なソースコードとして表現しており、プログラム全体に選択の幅があるといえる。また、本件プログ

ラム1のソースコードはA4用紙で128ページにのぼり（1ページあたり60行程度のコードが記載されている）、データ構造とそれぞれの関連の仕方が複雑であること（別紙1及び別紙2参照）からも、ソースコードの組み合わせ、表現順序等が、誰が作成しても同じものになるとはいえないことは明白である。

よって、本件プログラム1のソースコードには原告の個性が表れており、著作物性が認められる。

以上