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[redacted]

第８準備書面

令和３年６月１４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　[redacted]

同　[redacted]

同　[redacted]

（担当）同　[redacted]

第１　第７準備書面の訂正

第７準備書面１１頁の「＜ボードステータスを表示するランプでの判定について原告が作成したソースコード＞」について，以下の内容に訂正する。

１　訂正前

```
If AD_BoardFlag(i) And (.pbtmAD(i).Button = Button_Green Or .pbtmAD(i).Button =
Button_Yellow) Then
AD_Enabled(i) = True
```

```
.txtChannelString(i) = .ctrlAD(i).ChannelString
.lblChannelNumber(i) = .ctrlAD(i).ChannelNumber
.txtChannelString(i).Enabled = True
AD_ChN(i) = .lblChannelNumber(i)
Else
AD_Enabled(i) = False
.txtChannelString(i) = ""
.lblChannelNumber(i) = 0
.txtChannelString(i).Enabled = False
AD_ChN(i) = 0
End If
```

2　訂正後

```
For i = AD_LBound To AD_UBound
If AD_Enabled(i) Then
If AD_OpenFlag(i) = False Then AD_Open i
Ret = .ctrlAD(i).StartAI
AddStatus i, "StartAI", .ctrlAD(i).GetErrorString(Ret)
If Ret <> 0 Then
.pbtmAD(i).Button = Button_Red
AD_ErrorFlag(i) = True
AD_Start = False
AD_Stop
AD_Close
AD_Initialize
```

```
AD_Open
Else
If AD_ErrorFlag(i) Then
.pbtmAD(i).Button = Button_Yellow
Else
.pbtmAD(i).Button = Button_Green
End If
End If
End If
Next
```

第２　被告準備書面７に対する反論

１　第１について

被告は，「サブルーチン化するか否かはアイデアに過ぎない」，「コントロールの名前で引き渡すか否かについてもアイデアに過ぎない」などと主張する。

しかし，サブルーチンを作るか否かについての選択でソースコードの表現も異なり，さらにサブルーチン化する場合でも第６準備書面第１の１（１）ウの＜原告の作成したソースコード①＞及び＜原告の作成したソースコード②＞のとおり，作成するプログラマーの考えや経験によってサブルーチンについてのソースコードの表現の組み合わせは異なる。また，コントロールの名前で引き渡すことについても，コントロールを操作する場合，直接コントロールに対してプロパティを設定することや，メソッドを実行するのが一般的であるが，原告は，自らが最適と思うソースコードとして，コントロールの名前で引き渡すと

いう方法を選択している。このようにサブルーチン化やコントロールの名前で引き渡すことも原告独自の選択の結果である。

なお，被告は一定の処理方式を選択していること自体をもってアイデアにすぎないと主張しているものと思われる。しかし，問題とされるべきは「一定の結果を得るためにどのように指令を組み合せ，どの範囲で構造体（クラス）を設定し，配置・構造化するかについてどれだけの選択肢があったか」，「選択の結果である具体的な記述が，一定の意図のもとに特定の指令を組み合わせ，多数の構造体（クラス）を設定し，配置・構造化した独自のものになっているか」である。そして，本件プログラム１について多数の選択肢が存在したこと，原告の選択した具体的な記述が独自のものになっていることは原告第６準備書面第１に記載したとおりであり，創作性が認められるべきである。

２　第２について

（１）　被告は，「テキストファイルにするかデータベースにするか，データベースソフトの選択，コントロールによる方法か標準クラスによる方法か，クラス化するか否かについて・・・すべてアイデアにすぎない。」と主張する。

この点，どの選択肢にもメリットがあり，目的を実現することはできると考えられたところ，原告はプログラマーとしての経験及び技術力を総合的に考慮して，第６準備書面９頁＜図１＞の選択を行い，ソースコードを作成している。これも原告独自の選択の結果である。

そして，被告は上記と同様に，一定の処理方式を選択していること自体をもってアイデアにすぎないと主張しているものと思われるが，上述のとおり問題とされるべきは「一定の結果を得るた

めにどのように指令を組み合せ，どの範囲で構造体（クラス）を設定し，配置・構造化するかについてどれだけの選択肢があったか」，「選択の結果である具体的な記述が，一定の意図のもとに特定の指令を組み合わせ，多数の構造体（クラス）を設定し，配置・構造化した独自のものになっているか」であり，本件プログラムがこれらの点をみたすことは上記のとおりである。

3　第３について

（１）　まず，被告は「データベースの読み込みの作業を一度の処理で済ませること」及び「データの読み込みと同時処理を行うか否か」はアイデアに過ぎないと主張する。しかし，データベースを操作するＳＱＬ文は，原始的な命令から，よく使われる命令をセットにした命令が考えられ，どのような命令とするかはプログラマーの考え方によって異なる。そのため，データベースの読み込みを一度の処理で行うことはアイデアではなく原告独自の選択の結果である。そして，データの読み込みと同時処理とするか否かという点についても，原告のプログラマーとしての経験から，よりベストと思われる命令を組み合わせられる選択肢を選んでいるのであって，原告独自の選択の結果である。加えて，本件プログラム２のようにACCESS（アクセス）のデータベースを使用する場合のＳＱＬ文は，高機能データベース(オラクル等)と違い，制約があるため条件文を工夫する必要がある。条件文にどのような工夫を凝らすかもプログラマーによって対応が異なるため，他のプログラマーが作成しても同様のソースコードにはならない。

（２）　さらに，被告は「ＩＲフィルター」「ＦＩＲフィルター」「ＦＦＴフィルター」「ピークカット」「移動平均」の中から原告が

選択した加工処理方法は被告の仕様であるという旨主張する。

もっとも，原告は被告の仕様で上記のような処理を選出したのではなく，原告のこれまでのプログラマーとしての経験から本件プログラム２については，「ピークカット」「移動平均」の演算が有効ではないかと判断して選出している。その上で原告は実際に「ピークカット」と「移動平均」のフィルターを実際に試して，有効性が高かったものを組み合わせて第６準備書面２７頁乃至２８頁のソースコードを作成した。そのため，原告が作成したソースコードは，以上のように実験等をした上で組み合わせたものであるから，他のプログラマーが作成しても同様の記載になるものではなく，創作性がある。

第３　本件プログラム６の著作物性

１　本件プログラム６の概要

本件プログラム６は，鳥取自動車道智頭用瀬トンネル北工事（㈱大林組元請け）の工事現場でトンネル掘削時の発破振動を測定し，測定された振動データから報告書を作成するプログラムである。

本件プログラム６は，第３準備書面８頁で述べたとおり，事務所パソコン１台と測定パソコン１台又は２台の構成となっており，プログラムの機能としては，振動センサーから発信されるデータを測定パソコンでアナログ信号入力処理し，ハードディスクに収録した上で事務所パソコンに転送するというもの，事務所パソコンが転送されたデータを元に発破振動報告書を作成するというものがある。

２　ソースコードの構造について

本件プログラム６の主な特徴としては，アナログ信号入力処理，報

告書作成の処理，遠隔地でのリアルタイムデータ表示が挙げられる。以下，ソースコードの選択肢について（１）乃至（５）をピックアップして，詳述する。

３　ソースコードの選択肢に幅があること

（１）測定データのメモリー領域について（別紙　図①）

本件プログラム６は，センサーからの振動データをコンテック社のアナログ信号入力ボードを使用して取得しているところ，データを保存するメモリー領域について，主にＡＤＣボードに内蔵された領域を使用するようソースコードを組み合わせる方法と，プログラムで専用の領域を確保するようソースコードを組み合わせる方法が考えられた。ＡＤＣボード内蔵領域を使用する場合のソースコードは以下のものが考えらえる。

＜ＡＤＣボード内蔵領域を使用する場合のソースコード＞

```
'開始条件の設定：レベル比較
Ret = AioSetAiStartTrigger(Id, 3)
'開始レベルの設定
Ret = AioSetAiStartLevelEx(Id, 0, CSng(Text_Level.Text),
    Combo_Direction.SelectedIndex)
'メモリのリセット
Ret = AioResetAiMemory(Id)
'変換開始
Ret = AioStartAi(Id)
'変換データ取得
Ret = AioGetAiSamplingDataEx(Id, AiSamplingCount, AiData)
```

もっとも，原告は，本件プログラム６について，ＡＤＣボードに内在された領域では十分な機能を実現できないと考え，専用のメモリー領域を作成するようソースコードを組み合わせることを選択した。甲１６号証６頁乃至１０頁のソースコードによって，リングメモリー構造の記憶領域を実現している。

したがって，メモリー領域の設定だけでも，ソースコードの組み合わせに選択肢の幅がある。

（２）測定データの保存形式について（別紙　図②）

測定データを保存する際の保存形式について，主にテキスト，バイナリー，データベースの選択肢が考えられる。テキスト，データベースを使用する場合のソースコードの組み合わせとしては以下のものが考えられる。

＜テキストを使用する場合のソースコード＞

```
Dim sw As System.IO.StreamWriter
Dim enc As Encoding = Encoding.GetEncoding("shift_jis")
sw = New System.IO.StreamWriter("D:¥DATA.txt", True, enc)
sw.WriteLine(Data)
sw.Close()
```

＜データベースを使用する場合のソースコード＞

```
' コンストラクタ(初期処理)
  conStr = connectionString
 ' DB オープン
 sqlCon = New SqlConnection(conStr)
 sqlCon.Open()
```

```
' SQL 実行
command = New SqlCommand(sql, sqlCon, tran)
adapter = New SqlDataAdapter(command)
adapter.Fill(returnDt)
' DB クローズ
sqlCon.Close()
```

データの扱いやすさという点では，テキスト及びデータベース方式でソースコードを組み合わせることが考えられたが，原告は構造の最適化の観点を重視し，専用に設計されたバイナリー形式としてソースコードを組み合わせることを選択した。原告が選択した方法で作成したソースコードは，甲１６号証７頁４２行目乃至１０頁４３行目である。

したがって，測定データの保存形式にもソースコードの選択肢の幅がある。

（３）リアル測定データの転送方式について（別紙　図③）

本件プログラム６は，上述のとおり，現場で測定するパソコンと事務所のパソコンが存在することが想定されたものであり，データ通信によって現場と事務所のパソコンで測定データをリアルタイムに表示している。このデータ通信の方法について，シリアル通信，パケット通信等の選択肢が主に考えられる。このうち，シリアル通信を選択する場合のソースコードは，以下のようなものが考えられる。

＜シリアル通信を選択する場合のソースコード＞

```
SerialPort1.Open()                    ' シリアル通信ポートオープン
```

```
SerialPort1.WriteLine("*GetData?") ' データを要求
rd = SerialPort1.ReadLine          ' 測定結果を取得
SerialPort1.Close()                ' シリアル通信ポートクローズ
```

原告は，簡単に通信を実現する方法としてはシリアルが優れていると考えたが，データエラー対策やネットワーク環境においてはパケットの方法にメリットがあると考え，パケット通信を選択した。パケット通信を選択して組み合わせたソースコードが甲１６号証８１頁２２行目乃至８３頁５５行目である。

したがって，リアル測定データの転送方法に関するソースコードにも選択肢に幅がある。

（４）ファイルの転送方式について（別紙　図④）

現場の測定パソコンで計測されたデータのファイルを事務所のパソコンに転送するにあたって選択肢として，ＯＳでのネットワークドライブ接続の方法，ＦＴＰプロトコルによる接続の方法が考えられる。

原告は，簡単に通信を実現する方法としてはネットワークドライブが優れていると考えたが，通信障害が発生する可能性があることを考慮して，ＦＴＰ接続による方法を選択した。

この方法を選択し，独自に通信ステータスとファイルを監視し，リトライすることでファイル未転送の事態を防いでいる。原告は，甲１６号証２０６頁乃至２０８頁でクラス宣言を行い，２００ページ３９行目乃至２０１頁２２行目で実行を行うよう指令を組み合わせている。

したがって，転送方式のソースコードの組み合わせの選択肢にも

幅があるといえる。

（５）報告書作成（印刷）方法について（別紙　図⑤）

本件プログラム６では，現場の測定パソコンで発破振動を測定した結果について報告書を作成しているところ，報告書の作成（印刷）方式として，標準ライブラリで行う方法，サードパーティツールで行う方法，エクセル経由で行う方法等が考えられる。これらのうち，標準ライブラリで行う方法，サードパーティツールで行う方法を選択する場合のソースコードとしては，以下のようなものが考えられる。

＜標準ライブラリの方法を採る場合のソースコード＞

```
'Image オブジェクトを作成する
Dim canvas As New Bitmap(PictureBox1.Width, PictureBox1.Height)
'Image オブジェクトの Graphics オブジェクトを作成する
Dim g As Graphics = Graphics.FromImage(canvas)
'フォントオブジェクトの作成
Dim fnt As New Font("MS UI Gothic", 20)
g.DrawString("測定データ", fnt, Brushes.Blue, 0, 0)
'リソースを解放する
fnt.Dispose()
g.Dispose()
```

＜サードパーティツールの方法を採る場合のソースコード＞

```
Dim mReport As New Report
mReport.PageStart()
mReport.Write("MeasData01", Data(0), 1)
```

```
mReport.Write("MeasData02", Data(1), 0)
mReport.Write("MeasData03", Data(2), 0)
mReport.Write("MeasData04", Data(3), 0)
mReport.PageEnd()
mReport.Output()
```

サードパーティーのツールは，別途ライセンス費用が発生し，被告に多額の費用負担が生じるという難点があった。また，標準ライブラリは，費用面の負担はないものの，編集機能が他の選択肢よりも劣ると考えられた。一方，エクセル経由の方法は，エクセルアプリを購入する必要があるものの，基本的に業務で使用するパソコンにはエクセルがインストールされていることから，別途多額の費用が発生する可能性は低く，仮に購入するとしても高額にはならないことが想定された。加えて，エクセルは編集機能も柔軟性が高く他の選択肢よりも優れていると考えられたことから，原告はエクセル経由の方法を選択した。

原告が選択した方法に基づくソースコードは，甲１６号証１２６頁乃至１３２頁，１４４頁２７行目，１４７頁５行目乃至同頁４２行目，１４７頁４９行目乃至１５０頁６行目である。具体的には，甲１６号証１２６頁乃至１３２頁は，クラス選択のソースコードであり，１４４頁２７行，１４７頁５行目乃至４２行目は実行のソースコードである。なお，報告書には波形グラフを貼り付ける仕様になっているところ，波形グラフの画像は，１４７頁４９行目乃至１５０頁６行目の指令によって作成されている。

（６）小括

本件プログラム６のソースコードの選択肢として以上の５つをピックアップしたが，（１）乃至（５）のそれぞれに複数の選択肢があり，上述のとおりどの選択をしてもソースコードの組み合わせは異なる。

したがって，本件プログラム６全体にソースコードの選択肢の幅があることは明らかである。

４　原告の個性が現れていること

（１）アナログ信号入力処理について

本件プログラム６は，上記３（１）のとおり，原告が独自にソースコードを組み合わせて専用のメモリー領域を作成するようにしているところ，原告は併せて高速サンプリング，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。これらの機能を実現したソースコードは，甲１６号証６頁乃至１０頁である。

原告は専用のメモリー領域を作成するためにこれらの独自のソースコードを作成し，高速サンプリング等の複雑な機能を実現しているため，当該ソースコードは他のプログラマーが作成しても同様のものにはならず，創作性がある。

（２）遠隔地のリアルタイムデータ表示

上記３（３）のとおり，原告は本件プログラム６によって現場の測定パソコンだけでなく事務所のパソコンでもリアルタイムの測定データを見ることができるように通信機能に関するソースコードを組み合わせている。この通信は，常時実行されているため，他の処理を妨げる可能性があったことから，原告は最小限のデータ通信となるようなソースコードを作成した。そのソースコードは甲１６号証８１頁２２行目～８３頁５５行目の箇所である。

これらのソースコードは，本件プログラム６が現場の測定パソコンと事務所のパソコンを用いられるという特性に合わせて，原告が独自に作成したものであるため，他のプログラマーが作成したとしても同様の組み合わせにはならず，創作性がある。

（３）電文フォーマットについて

原告は，ＲＳ－２３２Ｃのシリアル通信の経験を活かし，上記３（３）で述べたパケット通信の電文にシリアル通信の電文を採用している。パケット通信であれば，エラー処理等は内部で自動的に行われるため，本文のみの通信でも問題はないところ，エラー処理やデータの開始判定及び終了判定を行うため制御コードを付加しており，データ通信が確実に行うことができるようソースコードを組み合わせた。具体的には，以下の２種類のソースコードにより，有効な通信データを取得（甲１６号証８１頁３行目乃至２１行目）及び通信データに制御コードを付加したデータ（甲１６号８２頁１７行目乃至４９行目）の送出を実現している。

＜有効な通信データ取得のコード＞

```
'TcpIP 受信バッファより有効文字列取得
Private Function GetReceiveBuffer(ByVal vIndex As Integer) As String
Dim mBuf0 As String, mRtnStr As String = ""
mBuf0 = pvReceiveBuffer(vIndex Mod 256)
Dim mPos1 As Integer = InStr(mBuf0, cSTX)
If mPos1 > 0 Then
Dim mBuf1 As String = mBuf0.Substring(mPos1 - 1)
Dim mPos2 As Integer = InStr(mBuf1, cETX)
If mPos2 > 0 Then
```

```
Dim mBuf2 As String = mBuf1.Substring(0, mPos2 + cETX.Length - 1)
mRtnStr = mBuf2.Substring(cSTX.Length, mBuf2.Length - (cSTX + cETX).Length)
mBuf0 = mBuf1.Substring(mPos2 + cETX.Length - 1)
Else
mBuf0 = mBuf1
End If
End If
pvReceiveBuffer(vIndex Mod 256) = mBuf0
Return mRtnStr
End Function
```

＜通信データに制御コードを付加して送出するソースコード＞

```
Public Sub WriteReadText(ByVal cl As ClientHandler, ByVal text As String)
'Listクラスが保持しているClientHandlerの中のSocketクラスの
'Handleを比較し、クライアントを識別します。
Dim no As Integer = 0, mIndex As Integer, tmp As String, mSendStr As String
Dim mGetTime As Date
For i As Integer = 0 To lstClientHandler.Count - 1
If lstClientHandler(i) Is cl Then
'クライアントのハンドルが一致した
no = CInt(lstClientHandler(i).ClientHandle)
mIndex = i
Exit For
End If
Next
```

```
'送られたメッセージをクライアントの情報と時間と共に書き込む
Dim mBuf As String
mBuf = "R" & no.ToString("0000") & "-" & mIndex.ToString("00") & ":" & text
PutStatus(Me.gridTcpReceive, mBuf)
PushReceiveBuffer(mIndex, text)
For i = 0 To 9
tmp = GetReceiveBuffer(mIndex)
If tmp = "" Then Exit For
If tmp.Substring(0, 10) = "GetNowData" Then
mSendStr = cSTX & "GetNowData," & GetNowData() & cETX
TcpIpSend(no, mSendStr)
ElseIf tmp.Substring(0, 7) = "GetData" Then
mGetTime = CDate(tmp.Substring(8))
mSendStr = cSTX & "GetData," & GetData(mGetTime) & cETX
TcpIpSend(no, mSendStr)
End If
Next
End Sub
'/**** 送信処理****/"
```

これらのソースコードは，原告が将来システムを拡張する上でイーサーネットによるパケット通信のみならず，他のインターフェイスを見据えた仕様を考えていたことから，当時の原告の経験及び独自のポリシーにもとづいた組み合わせとなっている。そのため，ソースコードも，原告が独自のプロトコルを立案し，プログラムコー

ドで実現しているものであるため，他のプログラマーが作成したとしても同様の組み合わせにはならない。

（４）振動波形グラフについて

原告は発破振動で測定した波形データを見える化するために，グラフを作成している（上記３（５）参照）。報告書の作成について，上記３（５）のとおり原告はエクセルを使用することを採用していることから，グラフを作成するためにエクセルにデータを渡し「折れ線」グラフを挿入することも考えられるが，原告はグラフのクオリティを高めるために独自でグラフを作成している。そのソースコードは以下のとおりである（甲１６号証１４７頁５０行～１５０頁７行）。

＜原告が独自でグラフを作成したソースコード＞

```
mImageFile = mImageFolder & "¥" & mImgWork1
pvExcel.AddPicture(mImageFile, "d5", 430, 350)
pvExcel.ExcelClose(, True)
pnlExcelRun.Visible = True
pnlExcelRun.Width = 920
Dim proc As Process = Process.Start("Excel.exe", Chr(34) & gReportInfo.GetExcelPath &
Chr(34))
proc.WaitForExit()
pnlExcelRun.Visible = False
Cursor.Current = Cursors.Default 'マウスカーソル元に戻す
End Sub
Private Sub DrawFlame(ByRef g As Graphics, ByVal vChNo As Integer)
Dim buf As String, mY As Double, a As Double
```

```
Dim mX1 As Integer, mX2 As Integer, mY1 As Integer, mY2 As Integer
Dim mChInfo As clsChannelInfo
Dim mTitle(3) As String
mTitle(1) = "Z 鉛直方向[gal]"
mTitle(2) = "X 東西方向[gal]"
mTitle(3) = "Y 南北方向[gal]"
'Penオブジェクトの作成(幅3黒色)
Dim p As New Pen(Color.Green, 1)
'フォントオブジェクトの作成
Dim fnt As New Font("MS UI Gothic", 20)
p.Color = Color.Black
p.DashStyle = Drawing2D.DashStyle.Dot
p.Width = 1
For a = -1 To 1 Step 0.5
mY = pvAxisMaxY * a
mY1 = PosY(mY)
g.DrawLine(p, pvRect.X, mY1, pvRect.X + pvRect.Width, mY1)
buf = StrRight(" " + mY.ToString("##0.0"), 5)
g.DrawString(buf, fnt, Brushes.Black, pvRect.X - 70, mY1 - 10)
Next
For a = -5 To 25 Step 5
mY = pvRect.Y + pvRect.Height + 5
mX1 = PosX(a) : mY1 = pvRect.Y + pvRect.Height
g.DrawLine(p, mX1, pvRect.Y, mX1, mY1)
buf = StrRight(" " + a.ToString("##0"), 2)
```

```
g.DrawString(buf, fnt, Brushes.Black, mX1 - 2, mY1 + 5)
Next
Dim t As Double
mX1 = 0 : mX2 = 0 : mY1 = 0 : mY2 = 0
p.DashStyle = Drawing2D.DashStyle.Solid
p.Color = Color.Red
p.Width = 1
mChInfo = pvDataFile.ChInfo(vChNo + 1)
For i = 0 To pvDataFile.GetRecScanN - 1
t = -5 + i / 2000.0
mX2 = PosX(t)
'mVolt = pvDataFile.Data(i, vChNo)
'mData = mChInfo.CompData(mVolt)
'mY2 = PosY(mData)
mY2 = PosY(pvDataFile.Data2(i, vChNo))
If i > 0 Then
g.DrawLine(p, mX1, mY1, mX2, mY2)
End If
mX1 = mX2 : mY1 = mY2
Next
fnt = New Font("MS UI Gothic", 20)
buf = mTitle((vChNo Mod 3) + 1)
g.DrawString(buf, fnt, Brushes.Black, pvRect.X + 50, pvRect.Y - 30)
buf = "最大加速度：" & pvDataFile.GetMax(vChNo). ToString("#0.00") & "(gal)"
g.DrawString(buf, fnt, Brushes.Black, pvRect.X + 1000, pvRect.Y - 30)
```

```
p.DashStyle = Drawing2D.DashStyle.Solid
p.Color = Color.Black
p.Width = 3
g.DrawRectangle(p, pvRect)
End Sub
```

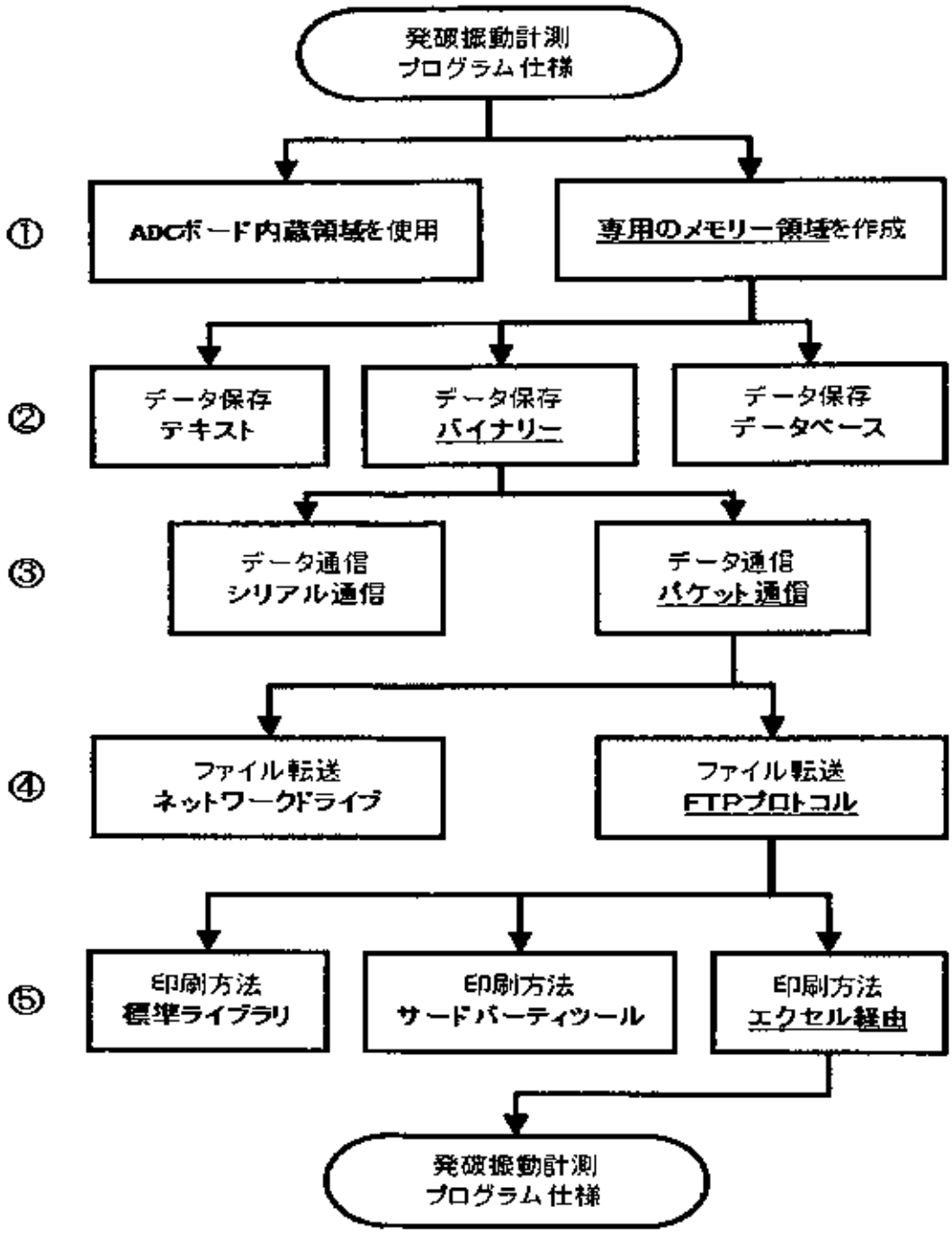
上記のソースコードにより，プログラム上でＪｐｅｇのイメージファイルを作成し，そのイメージファイルに対してゼロからグラフを描画し，描画したイメージデータをエクセル上に画像として挿入することが可能となっている。また，ドット単位の細かなカスタマイズも可能となっている。これらのグラフに関するソースコードも，原告がグラフのクオリティを向上させるために設計から作成まで独自の考えによって行ったものである。そのため，他のプログラマーが作成したとしても同様の記載にはならない。

（５）小括

以上より，本件プログラム６には原告の個性が現れている。

５　よって，本件プログラム６に著作物性が認められる。

別紙


発破振動計測
プログラム仕様
①
ADCボード内蔵領域を使用
専用のメモリー領域を作成
②
データ保存
テキスト
データ保存
バイナリー
データ保存
データベース
③
データ通信
シリアル通信
データ通信
パケット通信
④
ファイル転送
ネットワークドライブ
ファイル転送
FTPプロトコル
⑤
印刷方法
標準ライブラリ
印刷方法
サードパーティツール
印刷方法
エクセル経由
発破振動計測
プログラム仕様