

平成13年(ワ)第17306号 著作権侵害差止等請求事件
平成14年10月2日口頭弁論終結

判 決

原告	被告	株式会社ワイビーエム
同訴訟代理人弁護士		伊関正孝
同被告	被告	高橋利全
同訴訟代理人弁護士		佐鳥電機株式会社
同同		真茶山
同同		吉谷
同同		南増

主 文

- 1 原告の請求をいずれも棄却する。
- 2 訴訟費用は原告の負担とする。

事 実 及 び 理 由

第1 請求

- 1 被告は、別紙被告製品目録記載の製品を製造、使用、販売、頒布又は輸出してはならない。
- 2 被告は、前項記載の製品を廃棄せよ。
- 3 被告は、原告に対し、金4000万円及びこれに対する平成13年9月4日から支払済みまで年5分の割合による金員を支払え。

第2 事案の概要

原告は被告に対し、別紙被告プログラム目録記載の各プログラムを（以下、総称して「被告プログラム」という。）搭載した別紙被告製品目録記載の製品（以下「被告製品」という。）を製造、販売するなどの被告の行為が、原告の有する原告プログラム目録記載の各プログラム（以下、総称して「原告プログラム」という。）に係る著作権（複製権、翻案権、譲渡権）を侵害すると主張して、被告の上記各行為の差止及び損害賠償金の支払等を求めた。

- 1 前提となる事実（証拠等は文末に付した。）

(1) 原告プログラム

ア 吉沢ビジネス・マシンズ株式会社（以下「吉沢ビジネス・マシンズ」という。）は、平成元年9月ころ、MS-DOS3.1上で作動し、AutoCAD GXⅢ版に対応する別紙原告プログラム目録の1記載のプログラム（以下「原告プログラム1」という。）を開発し、東日本旅客鉄道株式会社（以下「JR東日本」という。）千葉支社等に対し、同プログラムを複製、格納した製品の納入を開始した（甲14、弁論の全趣旨）。

イ 吉沢ビジネス・マシンズは、平成8年11月ころ、原告プログラム1をバージョンアップし、Windows上で作動し、AutoCAD R13J版に対応する同目録の2記載のプログラム（以下「原告プログラム2」という。）を開発し、JR東日本秋田支社等に、同プログラムを複製、格納した製品を納入した（弁論の全趣旨）。

ウ 原告は、平成12年11月1日、吉沢ビジネス・マシンズから、代金2億円で、原告プログラムに関する開発、販売、保守サービス等一切の事業を譲り受けるとともに、原告プログラムの著作権及び同権利の侵害により同日までに発生した損害賠償請求権を譲り受けた（甲11、12）。

(2) 被告プログラム

ア 被告は、平成9年3月ころ、Windows上で作動し、AutoCAD R13J版に対応する被告プログラム目録の1記載のプログラム（以下「被告プログラム1」という。）を開発し、JR東日本の盛岡支社等に、これを複製、格納した製品を納入した（乙3、弁論の全趣旨）。

イ 被告は、平成10年10月ころには、被告プログラム1をバージョンアップし、AutoCAD R14版に対応する同目録の2記載のプログラム（以下「被告プログラム2」という。）を開発し、さらに平成13年10月ころには、被告プログラム2をバージョンアップし、AutoCAD 2000i版に対応する同目録の3記載のプログラム（以下「被告プログラム3」という。）を開発し、それぞれを複製、格納した製品をJR東日本の盛岡支社等に納入した（乙3、弁論の全趣旨）。

(3) 原告プログラムの内容

ア 原告プログラムは、AutoCAD上で作動し、鉄道電気設計及び設備管理用の図面を作成するコンピュータ支援設計製図ソフトプログラムである。なお、AutoCADとは、オペレーティングシステム（MS-DOS又はWindows）上で作動するオートデスク社製の汎用型CADシステム（2次元又は3次元の図面の作成、変更、削除、表示、印刷等を行うプラットフォーム）である。

原告が複製権侵害と主張する部分は、原告プログラムの、以下のイ及びウである。

イ 原告「電車線—基準線作成プログラム」

（ア）原告「電車線—基準線作成プログラム」は、原告プログラムの一部であるが、ユーザーの入力するデータに従って、電車線（電気機関車及び電車に動力用の電気を供給するために使用する接触電線等）の設計図面の作図を補助するため、縦及び上下の基準線を描画するプログラムである。

（イ）原告「電車線—基準線プログラム」は、各機能ごとに記述され、以下の5つのファイルに分けられている（甲17、27）。

- a メイン部 YBJ—TR68. lispファイル（別紙1）
- b 入力部 YBJ—TQ02. lispファイル（別紙2）
- c 修正部 YBJ—TR80. lispファイル
- d 描画部 YBJ—TR79. lispファイル（別紙3）
- e 説明部 YBJ—TR78. lispファイル

（ウ）原告「電車線—基準線プログラム」は、AutoLISP言語（インタープリター方式のプログラム言語。同言語の記述は、コンパイルする必要がない。）で記述されている。なお、原告プログラムは、暗号化されたソースプログラム形式によって、原告製品のハードディスク内に複製、格納され、同プログラムについてのオブジェクトプログラムは存在しない（弁論の全趣旨）。

ウ シェイプ定義に係る記述

原告プログラムを複製、格納した製品のハードディスク内には、多数の特殊文字（フォント）や特殊図形（シェイプ）についてのシェイプファイル（拡張子shxのバイナリデータファイル）が格納されている（甲19、24）。シェイプファイルは、シェイプ定義ファイル（拡張子がshpのファイル）をAutoCADのコマンドで機械語に翻訳（コンパイル）することによって生成されるファイルである。シェイプ定義ファイルは、AutoCADのシェイプ定義文に従って記述されている（乙2）。

（4） 被告プログラムの内容

ア 被告プログラムも、AutoCAD上で作動し、鉄道電気設計及び設備管理用の図面を作成するコンピュータ支援設計製図ソフトプログラムである。

イ 被告「電車線—基準線作成プログラム」

被告「電車線—基準線作成プログラム」も、被告プログラムの一部を構成し、ユーザーの入力するデータに従って、電車線（電気機関車及び電車に動力用の電気を供給するために使用する接触電線等）の設計図面の作図を補助するため、縦及び上下の基準線を描画するプログラムである。

被告「電車線—基準線プログラム」は、別紙4のとおり、BASELINE. lispファイル上に、AutoLISP言語で記述されている（甲18）。

ウ シェイプ定義に係る記述

被告プログラムを複製、格納した製品のハードディスク内には、特殊文字（フォント）や特殊図形（シェイプ）についてのシェイプファイル（拡張子shxのバイナリデータファイル）が格納されている（乙1）。シェイプ定義ファイルは、AutoCADのシェイプ定義文に従って記述されている。

2 争点

（1） 被告プログラムは、原告プログラムを複製ないし翻案したものに当たるか（被告製品を製造、販売するなどの被告の行為は、原告プログラムに係る複製権、翻案権、譲渡権を侵害するか。）

（2） 被告は、被告プログラムを作成するに際して、原告プログラム1を、自己のコンピュータ内の記憶媒体等に複製、格納したことがあるか。

（3） 原告の損害額はいくらか。

3 争点に関する当事者の主張

（1） 被告プログラムは、原告プログラムを複製ないし翻案したものに当たるか。

（原告の主張）

被告は、以下のとおり、原告プログラムに依拠して、原告プログラムと実質的に同一であるか、又は二次的著作物に当たる被告プログラムを作成した。被告プログラムは原告プログラムを複製ないし翻案したものである。

したがって、被告プログラムを搭載した被告製品を製造、販売する被告の行為は、原告プログラムについて原告が有する著作権（複製権、翻案権、譲渡権）を侵害する。

ア 原、被告の「電車線—基準線作成プログラム」の同一性

(ア) メニュー表示部

原、被告のメニュー表示部は、処理手順ないしフローチャート等、その基本的な構造が全く同一である。

(イ) 基準線のデータ入力部

a キロ行程最初の値の入力部

「電車線—基準線作成プログラム」のうち、キロ行程の最初の値を入力する部分は、原告プログラムが「(setq V0 (getreal “ \backslash n●キロ行程の最初の値をm単位で入力<0> : ”))」であるのに対して、被告プログラムが「(setq fStartKiro (getreal “ \backslash nキロ行程の最初の値をm単位で入力<0> : ”))」であり、ほとんど同一である。また、上記「”」内に記述された日本語部分（画面表示メッセージ）もほとんど同じである。

b キロ行程オフセット値の入力部

「電車線—基準線作成プログラム」のうち、キロ行程のオフセット値（スタート地点から図面を書き出す地点までの距離）を入力する部分は、原告プログラムが「(princ “ \backslash n \backslash n●キロ行程のオフセット値（スタート値から最初のマークまでの距離）をm単位で入力<0> : ”))」であるのに対して、被告プログラムが、「(setq f0ffsetKiro (getreal “ \backslash nキロ行程のオフセット値（スタート値から最初のマークまでの距離）をm単位で入力<0> : ”))」であり、ほとんど同一である。また、画面表示メッセージは完全に一致している。

c 縮尺の入力部

「電車線—基準線作成プログラム」のうち、縮尺を入力する部分は、原告プログラムが「(setq V2 (getreal “ \backslash n \backslash n●縮尺の分母のみ（例：1/500は500）入力<500> : ”))」であるのに対して、被告プログラムが、「(setq fScScaleC (getreal “ \backslash n尺度の分母のみ（1/500のときは500）を入力<500> : ”))」であり、ほとんど同一である。また、画面表示メッセージは、実質的に同一である。

d 用紙サイズの入力部

「電車線—基準線作成プログラム」のうち、用紙サイズを入力する部分は、原告プログラムが「(princ “ \backslash n \backslash n●キロ用紙サイズ A 4（高さ210mm） A 3（高さ297mm）”))」「(setq V3 (getstring “ \backslash n A 2（高さ420mm）を入力< A 4 > : ”))」であるのに対して、被告プログラムが「(setq sPaSize (getreal “ \backslash n用紙サイズ（A 4, A 3, A 2, A 1, A 0）または高さをm単位で入力< A 4 > : ”))」であり、ほとんど同一である。また、画面表示メッセージは、実質的に同一である。

e スパン距離の入力部

「電車線—基準線作成プログラム」のうち、スパン距離を入力する部分は、原告プログラムが「(setq V0 (getstring “ \backslash n \backslash nスパンを入力<50>”))」であるのに対して、被告プログラムが「(initget 6) (setq fSpan (getreal (strcat “ \backslash nスパン距離をm単位で入力<” (DelLastZero fOldSpan) ” > : ”))」であり、ほとんど同一である。また、画面表示メッセージは、実質的に同一である。

(ウ) 基準線の描画部

a 初期設定部

「電車線—基準線作成プログラム」のうち、基準線描画部の初期設定部（変数に値を設定する部分）の構成は、原、被告プログラムのいずれも、「キロ行程最初の値」、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の4項目を採用しているという点で、全く同じである。また、原、被告プログラムの処理の流れは、いずれも、上記4項目について入力されたデータを設定ファイルから読み出した後、「キロ行程最初の値」、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の順序で変数に値を設定する点において、全く同じである。

b スパンの描画部

「電車線—基準線作成プログラム」のうち、スパン（縦基準線間の距離）描画部の処理の流れは、原、被告プログラムのいずれも、「L（スパン線の左側の補助線の描画）」、「R（スパン線の右側の補助線の描画）」、「スパン（前回のスパン位置から、指示されたスパン距離分の間隔を空けた縦線の描画）」の順に繰り返し処理を行うという点において、実質的に同一である。

c 図面の上下基準線の描画部

「電車線—基準線作成プログラム」のうち、上下基準線を描画する部分の変数設定の処理の流れは、原告プログラムが、上下基準線を描画するための変数を設定する順序について、最下位の基準線、最上部の基準線、下から2番目の基準線、それ以降は上から2番目の基準線まで昇順としているのに対して、被告プログラムが、最下位の基準線、次いで昇順に一番上の基準線としている点において、ほぼ同一である。

d キロ行程の描画部

「電車線—基準線作成プログラム」のうち、キロ行程（1キロ、0.5キロ及び0.1キロごとに、表示するマーク）の描画部は、原、被告プログラムのいずれも、①キロ行程の算出の方法、②アルゴリズム、③キロ行程に配置する図形をシンボル化しているという点において、一致している。

イ シェイプ定義に係る記述の同一性

(ア) 特殊文字データ

被告の特殊文字データのソースプログラム（シェイプ定義）は、原告の特殊文字データのソースプログラムと実質的に同一である。

例えば、文字コード「OF27C」で表示される「(((」(甲33の222頁)についての原、被告のソースプログラム（シェイプ定義）は、ペンの動き（特殊文字を書き始める起点や、書く順序）こそ異なるものの、相対的な移動を指示する部分（1つ前の地点から移動する数値を、X座標、Y座標で記述したもの）はほぼ一致しており、その結果、画面上に表示される文字の形状（変化点の座標値）は完全に一致する。このような文字が、実務で使用される文字の90%を占めている。

(イ) シェイプコードの割当領域

シェイプコードの割当領域については、被告プログラムと原告プログラムとは実質的に同じである。

すなわち、原告プログラムでは、シェイプ定義ファイル先頭の宣言文において、「5F, 60」「7B, A0」及び「E0, FE」の3つの外字の割当領域を設定している。これは、原告プログラム1を作成した当時は、OS（MS-DOS）上に外字領域がなかったため、原告が独自にASCII（アスキー）文字を転換して「5Fから60」「7Bから7F」を外字領域として採用していたが、その後、原告プログラム2にバージョンアップする際に、Windows上に存在する外字領域である「F2からFE」に外字を割り当てようになったためである。したがって、原告プログラムでは、「F2からFE」領域に割り当てられた文字は、「5Fから60」「7Bから7F」の領域に登録されたシェイプ定義と全く同じものである。

これに対し、被告プログラムでも、外字領域として「5F, 60」「7B, 7F」「81, 9F」「E0, EA」「F2, FE」の5領域を設定している。「7B, 7F」「81, 9F」は、「80」を境にして「7B, A0」の領域を分割したものであり、同様に、「E0, EA」「F2, FE」は、「E0, FF」の領域を分割したものであるから、原、被告プログラムのシェイプコードの割当領域は実質的に同じである。

ウ 依拠性について

原告プログラムを複製、格納した製品は、JR東日本に納入されており、被告は、JR東日本の各支社に出入りする立場にあったから、JR東日本の従業員らの協力を得て、原告プログラムを解析する機会があった。

また、被告プログラムと原告プログラムの間には、後記のとおり、被告プログラムが独立して開発されたのであれば、あり得ないような一致点が数多く存在する。このことから、被告は、原告プログラムに依拠して、被告プログラムを作成したことが明らかである。

(被告の反論)

原告プログラムのうち、原告が、被告プログラムとの類似性を指摘する部

分は、以下のとおり、すべて創作性を有しない。また、原、被告プログラムは、実質的に同一ではないから、仮に、原告プログラムに創作性のある部分があったとしても、被告プログラムは、原告プログラムを複製ないし翻案したものではない。

ア 電車線—基準線作成プログラムの同一性

(ア) メニュー表示部

原告のメニュー表示部のプログラムは、「princ”\n1. データファイルの作成”」というものにすぎず、何ら創作性を有しない。

また、原告プログラムと被告プログラムとでは、表示されるメニューの文言が異なり、実質的に同一とはいえない。

(イ) 基準線のデータ入力部

a キロ行程最初の値の入力部

原告のキロ行程の最初の値を入力するプログラムの記述は、創作性を有しない。すなわち、原告のキロ行程の最初の値を入力するプログラムを簡単にモード化すれば、「setq A (getreal”メッセージ<実数>”)」となるが、これは、メッセージを画面に表示し、実数の入力を求め、入力された実数をAという変数に代入するという極めて基本的なプログラムであり、誰が作成しても同一の記述となる。

b キロ行程オフセット値の入力部

原告のキロ行程オフセット値の入力部のプログラムの記述は、単純なものであり、創作性を有しない。

また、原告プログラムでは、princ関数を使用しているのに対し、被告プログラムでは、setq関数及びgetreal関数を使用している点において、両者は異なる。

c 縮尺の入力部

原告の縮尺の入力部のプログラムの記述は、創作性を有しない。すなわち、縮尺の設定において、必要とされる要素は、縮尺の入力を求めるメッセージ表示と実数値の入力であり、そのために必要な最低限のプログラムをモード化すると、「setq A (getreal”メッセージ<実数>”)」となる。原告のプログラムは、この最低限必要なプログラムと同じものであって、創作性はない。

d 用紙サイズの入力部

原、被告の用紙サイズの入力部のプログラムは、実質的に同一とはいえない。すなわち、原告プログラムではgetstring関数を使用しているのに対し、被告プログラムではinitget関数及びgetreal関数を使用している点において、両者のプログラムは異なる。また、原告プログラムでは、入力されたデータを逐一ファイルに書き出しているのに対し、被告プログラムでは、リスト処理をしてからファイルに書き出している点において、データの処理方法が異なる。

e スパン距離の入力部

原告のスパン距離の入力部のプログラムと被告の該当部分とのプログラムとは、実質的に同一ではない。

(ウ) 基準線の描画部

a 初期設定部

原告プログラムの初期設定部において使用されている「キロ行程最初の値」「キロ行程オフセット値」「縮尺」「用紙サイズ」の各入力項目は、CADにおける製図の際の必要性及びユーザーであるJR東日本の要請によって必然的に決まるものであるから、その選択には創作性がなく、また、入力項目の変数の設定の順序にも創作性はない。

被告プログラムと原告プログラムとは、入力されたデータをファイルに書き出す順序及び使用する関数が異なっているから、実質的に同一ではない。

b スパン線の描画部

原告のスパン線の描画部のプログラムの処理の流れには、創作性がない。すなわち、原告プログラムのスパン線描画部は、「L(左)側 → R(右)側 → スパン」の順に繰り返し処理を行うものであるが、スパン線の描画部において入力されるデータは、スパン間の距離を表す数値又は左右の補助線を描画することを表す「L」又は「R」だけであるから、その処理順序は、自ずから限られている。

被告プログラムと原告プログラムとは、スパン線描画部のプログラムの具体的記述において異なっているから、両者は実質的に同一ではない。

c 図面の上下基準線の描画部

被告プログラムと原告プログラムとは、上下基準線を描画する上下基準線を描画する順序及び使用する関数が異なっているから、両者は同一ではない。

d キロ行程の描画部

被告プログラムと原告プログラムとは、キロ行程の描画部のアルゴリズムが異なっているから、両者は同一ではない。

イ シェイプ定義に係る記述の同一性

(ア) 特殊文字データ

a 原告が主張する「特殊文字データのソースプログラム」は、著作権法上の保護の対象となるプログラムではない。

すなわち、著作権法（以下「法」という場合がある。）が保護するプログラム著作物とは、「電子計算機を機能させて一の結果を得ることができるようにこれに対する指令の組み合わせたものとして表現したもの」（法2条1項10号の2）である。しかし、原告が主張する「特殊文字のソースプログラム」は、単なる座標値データであって指令の組み合わせではない。

b 仮に、特殊文字のシェイプ定義のうち、ペンの動き（文字を書く順序）に関する指令情報が著作権法上のプログラムに該当するとしても、被告の特殊文字のシェイプ定義と原告の特殊文字のシェイプ定義とは、ペンの動き及び変化点の座標値において異なるから、両者は同一とはいえない。

(イ) シェイプコードの割当領域

原告プログラムのシェイプコードの割当領域の選択には、創作性が無い。

ウ 依拠性について

被告プログラムは、被告がJR東日本から開発依頼を受けて、平成7年ころから、自ら開発を続けてきたものであり、原告プログラムに依拠したものではない。

(2) 被告は、被告プログラムを作成するに際して、原告プログラム1を、自己のコンピュータ内の記憶媒体等に複製、格納したことがあるか。

(原告の主張)

被告は、過去のいずれかの時点において、JR東日本の各支社に出入りする立場にあったことを利用するなどして、JR東日本に設置されているコンピュータ内のハードディスクに格納された原告プログラムを、自己のコンピュータ内の記憶媒体等に複製して格納したことがある。被告がこのような行為を行ったことは、以下の事実に照らして明らかである。

被告の上記複製行為は、原告プログラムについて原告が有する著作権（複製権、翻案権）を侵害する。

ア 特殊文字データの変化点の座標値の一致

原告プログラムの特殊文字の形状と、被告プログラムの特殊文字の形状とは、約90%以上がほぼ完全に一致している。このことは、これらの特殊文字のフォント定義において記述されている変化点の座標値が一致していることを意味するが、変化点の座標値は、ソースプログラムを見なければ分からないものであるから、被告プログラムの特殊文字のソースプログラムは、原告プログラムを複製し、解析して作成したものである。

イ シェイプコードの割当領域の一致

原告プログラムと被告プログラムのシェイプコードの割当領域は、ほぼ完全に一致している。また、原、被告プログラムのいずれにおいても、「F2からFE」領域に割り当てられた文字は、「5Fから60」「7Bから7F」の領域に登録されたシェイプ定義と全く同じである。原告プログラムでは、特殊文字が二重に定義されているのは、原告プログラム1のOSが外字領域を有しないMS-DOSであったためであるが、Windows上で作動する被告プログラムでは、「5Fから60」「7Bから7F」の領域に登録された特殊フォントは不要であるにもかかわらず、これらの文字が原告プログラムと同じく二重に定義されている点で不自然である。

このような一致点が存在することは、被告が、原告プログラムを複製し、これを解析して被告プログラムを作成したことを示している。

ウ 補助線の数の一致

原告の電車線-基準線プログラムと被告の電車線-基準線プログラムとは、描画される横の補助線の数が一致している。補助線の数は、沿革的な理由で残されているにすぎないにもかかわらず、この数が一致するということは、被告が、

原告プログラムを複製し、これを解析して被告プログラムを作成したことを示している。

エ エスケープコードの一致

原、被告プログラムは、いずれも「"} ”をエスケープコードとして使用している。

原告は、信号設備台帳プログラムやキロ行程の描画部において、小さい上付き文字を表記するためのエスケープコードとして、「"} ”を採用しており、たとえば「M} ”とAutoCADに指令すれば、「M」の小さい上付き文字がAutoCAD上に描かれる。原告プログラムで、このような方法を採用したのは、OSがMS-DOSであった時代には、外字領域がなかったため、アスキー文字の「"} ”を転用して、特殊文字コントロール記号としたためである。

これに対し、被告プログラムでも、原告プログラムと同様に、「"} ”をエスケープコードとして「小さい上付き文字」を表示している。被告プログラムが開発された当時、OSはWindowsであり、Windowsには外字領域が定義されているのであるから、アスキー文字を特殊文字コントロール記号として転用する必要が全くなかったことからすれば、被告プログラムで、このような方法を採用するのは、不自然である。

このような一致点が存在することは、被告が原告プログラムを複製し、これを解析したことを示している。

オ 改行コードの一致

原告の信号ケーブルプログラムでは、改行コード（入力した文字列を図面上に表示する際の改行を制御するためのコード）として、「086BA」を使用しているところ、被告プログラムでも、全く同じ「086BA」を改行コードとして使用している。また、改行量も同じであり、このため、原、被告プログラムで作成した文字列の図面上の配置位置（座標値）は、完全に一致する。

このような改行コードが偶然一致することはあり得ず、これは、被告が、原告プログラムを複製し、解析したことを示している。

カ 変数に値を設定する順序の一致

上記のとおり、電車線一基準線プログラムのうちの初期設定部は、原、被告プログラムのいずれも、入力されたデータを設定ファイルから読み出した後、変数に値を設定する順序が、「キロ行程最初の値」、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の順であるという点において、全く同じである。

変数に値を設定する順序は、プログラマーが自由に決定できるものであるから、被告プログラムが独立して開発されたものであれば、両者において変数に値を設定する順序が一致する確率は極めて小さい。特に、被告プログラムでは、入力されたデータを設定ファイルに書き出す順序は、「縮尺」、「用紙サイズ」、「キロ行程最初の値」、「キロ行程オフセット値」の順であるにもかかわらず、変数に値を設定する順序がこれと異なるのは、不自然である。

このことは、被告が原告プログラムを複製し、解析したことを示している。

（被告の反論）

被告は、被告プログラムを作成するに際して、原告プログラム1を自己のコンピュータ内の記憶媒体等に複製して格納をしていない。

原告は、原、被告プログラムでは、「特殊文字の形状」、「シェイプコードの割当領域」、「補助線の数」、「エスケープコード」、「改行コード及び改行量」、「変数に値を設定する順序」等が一致することをもって、被告が原告プログラムを複製したと主張する。しかし、以下のとおり、上記各一致点は、原告プログラムを直接複製した上で解析しなくても、他の方法により認識し得る事項であるから、被告が原告プログラムを複製した根拠とはならない。

ア 原告の主張ア及びイについて

特殊文字の形状及びシェイプコードの割当領域は、原告プログラム上でフォント一覧表を作成するLISPプログラムを実行し、印刷した文字フォント一覧表から分かるものであり、被告は、被告プログラムと原告プログラムとの互換性を図るために、同一覧表を作成し、これに基づいて、原告プログラムのシェイプコードと同じコードの特殊文字を作成した。

イ 原告の主張ウについて

補助線の数は、原告プログラムが作成した図面上分かるものであり、被告は、JR東日本から補助線の数について変更要請がなかったため、従前と同じ本

数を採用した。

ウ 原告の主張エについて
エスケープコードは、原告プログラムが作成した図面上から分かるものであり、被告は、原告プログラムと操作上の互換性を図るために、同じエスケープコードを採用した。

エ 原告の主張オについて
改行コードは、原告プログラムが作成した図面上から分かるものであり、改行量については、異なる改行量を数回図面上で試行することにより、原告プログラムが定義した改行量を探し当てることができる。

(3) 原告の損害額はいくらか。

(原告の主張)

被告は、平成7年ころから現在まで、被告プログラムを複製、格納した製品を少なくとも合計100台製造し、販売した。上記製品の1台当たりの利益額は40万円を下らない。

したがって、原告プログラムに係る著作権を侵害されたことにより生じた原告の損害額は、4000万円を下らない。

(被告の反論)

否認ないし争う。

第3 当裁判所の判断

1 争点1 (被告プログラムは、原告プログラムを複製ないし翻案したものに当たるか。) について

(1) プログラムの創作性の有無及び同一性の判断について

ある表現物が、著作権法の保護の対象になる著作物に当たるというためには、思想、感情を創作的に表現したものであることが必要である。そして、創作的に表現したものであるためには、当該表現が、厳密な意味で独創性のあることを要しないが、作成者の何らかの個性が発揮されたものであることは必要である。

この点は、プログラム(電子計算機を機能させて一の結果を得ることができ、この点にこれに対する指令を組み合わせたものとして表現したもの)形式で表現されたものであっても何ら異なることはない。プログラムは、具体的記述において、作成者の個性が表現されていれば、著作物として著作権法上の保護を受ける。

ところで、プログラムは、その性質上、表現する記号が制約され、言語体系が厳格であり、また、電子計算機を少しでも経済的、効率的に機能させようとする、指令の組合せの選択が限定されるため、プログラムにおける具体的記述が相互に類似することが少なくない。仮に、プログラムの具体的記述が、誰が作成してもほぼ同一になるもの、簡単な内容をごく短い表記法によって記述したもの又は極くありふれたものである場合においても、これを著作権法上の保護の対象になるとすると、電子計算機の広範な利用等を妨げ、社会生活や経済活動に多大の支障を来す結果となる。また、著作権法は、プログラムの具体的表現を保護するものであつて、機能やアイデアを保護するものではないところ、特定の機能を果たすプログラムの具体的記述が、極くありふれたものである場合に、これを保護の対象になるとすると、結果的には、機能やアイデアそのものを保護、独占させることになる。したがって、電子計算機に対する指令の組合せであるプログラムの具体的表記が、このような記述からなる場合は、作成者の個性が発揮されていないものとして、創作性がないというべきである。

さらに、プログラム相互の同一性等を検討する際にも、プログラム表現には上記のような特性が存在することを考慮するならば、プログラムの具体的記述の中で、創作性が認められる部分に対比することにより、実質的に同一であるか否か、あるいは、創作的な特徴部分を直接感得することができるか否かの観点から判断すべきであつて、単にプログラムの全体の手順や構成が類似しているか否かという観点から判断すべきではない。

以下、上記のような点を総合考慮して、原告プログラムの創作性の有無及び被告プログラムとの対比について検討する。

(2) 電車線—基準線作成プログラムについて

ア メニュー表示部

(ア) 原告プログラムの内容

原告プログラム1の電車線—基準線作成プログラムのメニュー表示部の内容は、以下のとおりである(甲17, 25)。

原告プログラム1の電車線—基準線作成プログラムのメインプログラ

ムは、画面上に、「1. データファイルの作成」、「2. データファイルの修正」、「3. 基本線作成」、「4. データファイルの文法説明」、「0. 終了」というメニュー一覧を表示し、ユーザーが入力したメニュー番号に応じて、各機能を実行するファイルを呼び出す（ロードする）機能を有するものである。

そのプログラムの記述は、別紙1のとおりであり、1から28行目までの合計28行から構成されている（空行がある。また、1から4行目は、プログラムの内容や作成年月日などのメモである。）。

a 6行目は、「(setq B1 0)」であり、変数B1を初期化することを指示する。setq（セットキュー）関数は、AutoLISP言語における基本的な代入関数である。

b 8から20行目は、主として、「(princ "\n【メニュー名】")」、「(princ "\n")」という構文を6回繰り返すものであり、【メニュー名】の部分には、順次、「<<JR-CAD>>【電車線キロ行程、基準線作成】」、「1. データファイルの作成」、「2. データファイルの修正」、「3. 基本線作成（データファイル作成後）」、「4. データファイルの文法説明」、「0. 終了」が表示されている。princ（プリンシー）関数は、「"」内の記述を、そのまま画面上に表示することを指令する。「\n」はMS-DOS版AutoCADにおける改行コードである。

c 21行目は、「(setq B0 (getint "\n目的の番号を入力<0>:")) (if (=B0 nil) (setq B0 0))」であり、「目的の番号を入力<0>:」というメッセージを画面に表示した後、ユーザーが入力した整数値（メニュー番号）を、変数B0に設定することを指令する。

d 23から27行目は、「(if (=B0 【メニュー番号】) (load "【ファイル名】"))」という構文を4回繰り返し、最後に「(if (=B0 0) (setq B1 1))」と記述され、【ファイル名】の部分には、順次、データ入力部、修正部、描画部、説明部を実行するファイル名が記述されている。この部分は、ユーザーの入力するメニュー番号に応じてデータ入力部、修正部、描画部、説明部のいずれかのファイルをAutoCADに読み込み、ユーザーがメニュー番号「0」（終了）を選択した場合には、変数B1に1を設定し、処理を終了することを指令する。

(イ) 創作性の有無

原告プログラムのメニュー表示部のプログラム記述は、全体として短く、その大部分が、AutoLISP言語で定められた一般的な関数を用いて、簡単な指令を組み合わせたものにすぎない。したがって、原告プログラムは、制作者の個性が発揮された表現とはいえず、創作性がない。

また、原告プログラムのメニュー表示部における処理の流れは、①画面上に、データ作成（入力）、修正、描画、説明、終了の順に各メニューメッセージを表示し、②ユーザーにいずれかのメニュー番号を選択（入力）させ、③ユーザーが入力したメニュー番号に応じて、各機能を実行するファイルを呼び出すものであるが、これらの流れは、法10条3項3号所定の「解法」に当たるといふべきであって、著作権の保護が及ばない。

以上のとおり、原告プログラムのメニュー表示部は創作性がない。

イ 基準線のデータ入力部

(ア) キロ行程最初の値の入力部

a 原告プログラムの内容

原告プログラムのキロ行程最初の値の入力部の主たるプログラム記述（原告が、被告プログラムの該当部分と同一であると主張する記述。後記(イ)ないし(オ)においても同じである。）は、「(setq V0 (getreal "\n●キロ行程の最初の値をにm単位で入力<0>:"))」の1行であり（別紙2の13行目）、これにより、キロ行程の最初の値の入力を指令するメッセージを画面上に表示し、入力された実数を、変数V0に設定することを指令する（甲17, 25）。

原告の記述部分を構文にすると「setq V0 (getreal "メッセージ")」となる。Getreal（ゲットリアル）関数は、AutoLISP言語で実数値を入力させる際に使用される関数であり、関数の後に記述される「"」で囲まれた文字列を、そのまま画面に表示する。¥nは、改行を意味するコードである。

b 創作性の有無

原告プログラムのキロ行程最初の値の入力部の記述は、画面上に「●キロ行程の最初の値をにm単位で入力<0>:」という文字列を表示した上、ユーザーが入力した実数値を、AutoLISP言語の一般的な関数を用いて、変

数に設定するという極めて簡単な内容を、ごく短い構文で表現するものである。したがって、キロ行程最初の値の入力部の記述は、制作者の個性が発揮された表現とはいえ、創作性はない。

(イ) キロ行程オフセット値の入力部

a 原告プログラムの内容

原告プログラムのキロ行程オフセット値の入力部の主たるプログラム記述は、「(princ"¥n¥n¥n●キロ行程のオフセット値(スタート値から最)")」「(setq V1(getreal"¥n初めのマークまでの距離)をm単位で入力<0>:"))」の2行であり(別紙2の17, 18行目)、これにより、キロ行程オフセット値の入力を指令するメッセージを画面上に表示し、入力された実数を、変数V1に設定することを指令する(甲17, 25)。

原告の記述部分を構文にすると、「setq V0(getreal"メッセージ"))」となるが、画面表示メッセージがゲットリアル関数に続く「'''」内にすべて記載されておらず、メッセージの一部は、前行のプリンシー関数を用いて画面表示する。

b 創作性の有無

原告プログラムのキロ行程オフセット値の入力部の記述は、画面上に「●キロ行程のオフセット値(スタート値から最初のマークまでの距離)をm単位で入力<0>:」という文字列を表示した上、ユーザーが入力した実数値を、AutoLISP言語の一般的な関数を用いて、変数に設定するという極めて簡単な内容を、ごく短い構文で表現するものである。

したがって、キロ行程オフセット値の入力部の記述は、制作者の個性が発揮された表現とはいえ、創作性はない。

(ウ) 縮尺の入力部

a 原告プログラムの内容

原告プログラムの縮尺の入力部の主たるプログラム記述は、「(setq V2(getreal"¥n¥n¥n●縮尺の分母のみ(例:1/500は500)入力<500>:"))」の1行であり(別紙2の22行目)、これにより、縮尺の分母の入力を指令する「●縮尺の分母のみ(例:1/500は500)入力<500>:」というメッセージを画面上に表示し、入力された実数を、変数V2に設定することを指令する(甲17, 25)。

原告の記述部分を構文にすると、上記「キロ行程最初の値の入力部」及び「キロ行程オフセット値の入力部」の記述と同じく、「setq V0(getreal"メッセージ"))」となる。

b 創作性の有無

原告プログラムの縮尺の入力部の主たるプログラム記述は、キロ行程最初の値の入力部及びキロ行程オフセット値の入力部の記述と同じく、制作者の個性が発揮された表現とはいえ、創作性はない。

(エ) 用紙サイズの入力部

a 原告プログラムの内容

原告プログラムの用紙サイズの入力部の主たるプログラム記述は、「(princ"¥n¥n¥n●キロ用紙サイズA4(高さ210mm)A3(高さ297mm)")」「(setq V3(getstring"¥n A2(高さ420mm)を入力<A4>:"))」の2行であり(別紙2の26, 27行目)、用紙サイズの入力を指示するメッセージとして、「●キロ用紙サイズA4(高さ210mm)A3(高さ297mm)A2(高さ420mm)を入力<A4>:」を画面上に表示し、入力された文字列を、変数V3に設定することを指令する(甲17, 25)。

原告の記述部分を構文にすると、「setq V3(getstring"メッセージ"))」となり、用紙サイズに関しては、入力されるデータは、「A4」「A3」といった文字列であるため、文字列の入力に対応するAutoLISP言語の関数であるgetstring(ゲットストリング)関数を使用している。また、メッセージの一部は、前行において、プリンシー関数を用いて画面表示する。

b 創作性の有無

原告プログラムの用紙サイズの入力部の記述は、画面上に「●キロ用紙サイズA4(高さ210mm)A3(高さ297mm)A2(高さ420mm)を入力<A4>:」という文字列を表示した上、ユーザーが入力した文字列を、AutoLISP言語で文字列の際に通常用いられる関数を用いて、変数に設定するという極めて簡単な内容を、ごく短い構文で表現するものである。

したがって、原告の用紙サイズの入力部の記述は、制作者の個性が発揮された表現とはいえ、創作性はない。

(オ) スパン距離の入力部

a 原告プログラムの内容

原告プログラムのスパン（縦の基準線間の距離）の入力部の主たるプログラム記述は、「setq V0 (getstring "¥n¥nスパンを入力<50>")」の1行であり（別紙2の36行目）、これにより、スパンの入力を指示するメッセージとして、「スパンを入力<50>」を画面上に表示し、入力された文字列を、変数V0に設定することを指令する（甲17, 25）。

原告の記述部分を構文にすると、用紙サイズの入力部の記述と同じく、「setq V0 (getstring "メッセージ")」である。スパンに関しては、入力されるデータは、スパン（実数値）だけではなく、スパンの左右に補助線を描画することを指示する文字列（「L」又は「R」）も含まれるため、文字列の入力に対応するゲットストリング関数を使用している。

b 創作性の有無

原告プログラムのスパンの入力部の記述は、画面上に「スパンを入力<50>」という文字列を表示した上、ユーザーが入力した文字列を、AutoLISP言語で文字列の際に通常用いられる関数を用いて、変数に設定するという極めて簡単な内容を、ごく短い構文で表現するものである。

したがって、スパン距離の入力部の記述は、制作者の個性が発揮された表現とはいえ、創作性はない。

ウ 基準線の描画部

(ア) 初期設定部

a 原、被告プログラムの内容

(a) 原告プログラムの初期設定部は、ユーザーが入力部で入力したデータに基づいて基準線を描画するため、入力部でデータファイルに書き込んだデータを読み出して、変数に設定する役割を有する。原告プログラムの初期設定部の記述は、描画部ファイル（別紙3）の記述のうち22から53行目までの32行であり、その構成は、以下のとおりである。

まず、「キロ行程の最初の値」について、①「(setq 【変数】 (read-line F1))」という構文により、データファイル内に書き込まれているデータを先頭から1行分読み込み、変数に設定し、②rtos関数及びatof関数（文字列を実数に変換する関数）などを使用して、上記①の変数を、作図の際に使用できる実数値又は文字列等に変形させ、その結果を、セットキュー関数で新たな変数に設定し直し、③新しい変数の設定結果をプリンター関数で画面上に表示し、その後、①から③までの処理を、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の各入力項目ごとに、繰り返す。

原告プログラムの初期設定部において、変数を設定する順序は、入力部においてデータをファイルに読み込んだ順序と同一で、「キロ行程の最初の値」、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の順序である。これは、原告プログラムが、入力部において、「(write-line 【変数】 F0)」という構文を使って、入力されたデータを、入力された順に逐次データファイルへ書き込んでいくところ（入力部ファイルの30から34行目）、初期設定部のデータの読み込みに使われているread-line関数は、データがファイルに書き込まれた順に先頭から読み出すことしかできないためである（甲17, 25）。

(b) これに対し、被告プログラムの初期設定部に該当する部分は、被告の電車線ー基準線作成ファイル（別紙4）の284から291行目までの8行であり（DrawBaseLine関数というローカル関数の定義内に記述されている。）、その構成は、以下のとおりである。

まず、「キロ行程の最初の値」について、①lsBaselineDataというリスト（複数の値をもった変数）の中から、必要なデータを読み出し、②データをそのままatof関数、fix関数（小数部分を切り捨てて整数変換する関数）などで評価し、③評価した数値をセットキュー関数で変数に設定し、上記①から③の処理を、順次、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」という順序で行う。

被告の上記処理の具体的記述は、「(setq 【変数】 (【関数 (atof関数など)】 (nth 【リスト内の順序】 lsBaselineData)))」という1行の構文を主として、これを各入力項目（変数）ごとに、繰り返す。「(nth 【リスト内の順

序】IsBaseLineData)」は、リストの中身のデータを読み込む指令であり、Read-line (リードライン) 関数のように、データファイルの先頭からだけではなく、データファイル内のうち、【リスト内の順序】で指定された位置にあるデータを読み出すことができる。なお、被告の電車線—基準線プログラムでは、IsBaseLineData内のデータは、縮尺、用紙サイズ、キロ行程の最初の値、キロ行程オフセット値、スパン距離の順序で保持されている(甲18, 25, 27)。

b 原告プログラムの創作性の有無及び対比

(a) 原告プログラムにおける入力項目として何を用いるかという点は、アイデアであり、著作権法上の保護の対象となるものではない。また、「キロ行程最初の値」、「キロ行程オフセット値」、「縮尺」、「用紙サイズ」の順序で変数に値を設定するという処理の流れも、法10条3項3号所定の「解法」に当たり、著作物としての保護を受けない。

(b) 仮に、原告プログラムの初期設定部の具体的記述に、創作性が生じると解する余地があるとしても、前記認定の原告プログラムの内容に照らして、創作性の範囲は極めて狭いものというべきである。そして、被告プログラムと原告プログラムとは、初期設定部に用いられている構文上の相違によって具体的記述が大きく相違する。被告プログラムの初期設定部の具体的記述は、原告プログラムの初期設定部の記述と実質的に同一とはいえないし、原告プログラムの創作性を有する本質的な特徴部分を直接感得することもできない。

(c) したがって、原告プログラムの初期設定部について複製権又は翻案権侵害があるとは認められない。

(イ) スパン線の描画部

a 原、被告プログラムの内容

(a) スパン線の描画部は、ユーザーから入力され、データファイルに書き出されたスパン(縦基準線間の距離)に関するデータ(スパン間の距離を表す数値又は左右補助線を描画することを表す「L」又は「R」)を基に、縦の基準線、左右補助線、スパンの数値、スパン線(中央の横基準線)の描画を行う役割を有する。

原告プログラムのスパン線の描画部の記述は、原告の描画部ファイル(別紙3)の59から105行目までの47行であり、その構成は、以下のとおりである。

まず、①変数の初期化等の処理の後、画面左端の基準線の作図を指令し、②データファイルに入力されているスパン距離に関するデータを、「(setq 【変数】(read-line F1))」という構文で、ファイルの頭から1行分読み込み、③読み込んだデータが空ではない場合に、後続の処理を行うように指令し、④「(if (= (substr S0 1 1) "L")）」という指令で評価して、読み込んだデータが「L」の場合には、AutoLISP言語のコマンドである「command "line"(list X座標1, Y座標1)(list X座標1, Y座標2)）」という構文を用いて、左補助線を描画するよう指令し(X座標1は、1つ前の基準線のX座標値から-4.0オフセットした値、Y座標値1及び同2は、それぞれ左端基準線の両端の点のY座標値である。)、⑤読み込んだデータが「R」の場合には、同様に右補助線を描画するよう指令し、⑥読み込んだデータが実数値(スパン距離)の場合には、前回のスパン位置からスパン距離分右側の位置に縦基準線を描画し、かつ、スパン距離を表示するよう指令し、⑦上記処理が終了した後は、データファイルからさらにデータを読み込んで変数を設定するよう指令し、データがあれば上記③の処理を繰り返すが、データがなければ(変数が空になれば)、処理を終了する(甲17, 25, 27)。

(b) これに対し、被告プログラムのスパン線描画部の記述は、被告の電車線—基準線作成ファイル(別紙4)の329から363行目までの34行であり、その構成は、以下のとおりである。

まず、①画面左端の基準線の作図を指令し、作図した結果の図形情報を、AutoLISP言語の関数であるputlayer関数を用いて、更新し、②初期カウンタ(変数)を「4」に設定し、リスト内のデータの5番目以降(ユーザーが入力し、リスト内に保持したデータのうち、5番目以降がスパンに関連するデータであるため)を処理対象とするよう指定し、③リストの全体の長さを評価して、カウンタがリストの長さに達しない限り、後続の処理を繰り返すよう指令し、④カウンタの順序(最初は5番目)に位置するデータをリストから読み出してセットキ

ユー関数で変数（「vItem」）に設定し、⑤「（＝（strcase vItem）」という構文で評価（変数が大文字に変換される。）した結果、変数がLである場合は、左補助線を描画するよう指令し、⑥上記⑤と同じ評価の結果、変数がRである場合には、右補助線を描画するよう指令し、⑦「（＜0.0（setq fSpan（atof vItem）」という構文で評価（数値であれば、実数に変換が行われる。）した結果、変数が実数である場合は、縦基準線及びスパンの数値を描画するよう指令し、⑧最後に、処理カウンタに1を足した後、上記③に戻り、③以降の処理を繰り返す（甲18、25、27）。

b 原告プログラムの創作性の有無及び対比

(a) 原告プログラムにおいて、読み出したデータを「L」、「R」、「スパン」の順序に評価し、描画をするという処理の流れは、法10条3項3号所定の「解法」に当たり、著作物としての保護を受けないというべきである。

(b) 仮に、原告プログラムのスパン描画部の具体的記述に、創作性が生じると解する余地があるとしても、前記認定のプログラムの内容に照らして、創作性の範囲は極めて狭いものというべきである。そして、被告プログラムと原告プログラムとは、その具体的な変数の評価方法や、全体の処理を繰り返すための方法手順が異なる。被告プログラムのスパン描画部の具体的記述は、原告プログラムのスパン描画部の記述と実質的に同一とはいえないし、原告プログラムの創作性を有する本質的な特徴部分を直接感得することもできない。

(c) したがって、原告プログラムのスパン線描画部の記述について複製権又は翻案権の侵害があるとは認められない。

(ウ) 図面の上下基準線の描画部

a 原、被告プログラムの内容

(a) 上下基準線の描画部は、画面上部に3本、画面下部に6本、合計9本の上下基準線の描画を行う機能を有する。

原告プログラムの上下基準線の描画部の記述は、原告の描画部ファイル（別紙3）の108から123行目までの16行であり、その構成は、主として、「command "line" (list X座標1 Y座標1) (list X座標2 Y座標2)""」という構文を、最下部の基準線、最上部の基準線、下から2番目の基準線、以降は昇順に上から2番目の基準線まで、順次9回繰り返すというものであり、これにより、各基準線の始点と終点を、XとYの座標値で特定し、2点間に線を描くことを指令している（甲17、25）。

(b) これに対し、被告プログラムの上下基準線の描画部の記述は、被告の電車線一基準線作成ファイル（別紙4）の365から407行目までの43行であるが、あらかじめそれぞれの基準線のオフセット値（基準線間の間隔。変数の設定の順序は、最下位の基準線から昇順である。）及び縦線の高さの半分の値を変数として設定されており（293から303行目）、上下基準線の描画部においては、「command "Line" 変数1 変数2 ""」という構文の指令を繰り返し、1つ前に描画した基準線の座標にオフセット値（変数）を加算して、次の基準線の座標（変数1、変数2）を求めるという方法で、上下基準線を、画面中央部上側（上から3番目の基準線）から順に上方向へ3本、次いで中央部下側（下から6番目の基準線）から下部方向へ合計6本の基準線（合計9本）を描画している（甲18、25）。

b 原告プログラムの創作性の有無及び対比

(a) 原告プログラムにおいて、上下基準線の座標値を下の基準線から上の基準線へ昇順に設定し、描画するという処理の流れは、そもそも、法10条3項3号所定の「解法」に当たり、著作物としての保護を受けないというべきである。

(b) 仮に、原告プログラムの上下基準線描画部の具体的記述に、創作性が生じると解する余地があるとしても、前記認定のプログラムの内容に照らして、創作性の範囲は狭いものというべきである。そして、被告プログラムと原告プログラムとは、前記認定のとおり、基準線を描画する順序や、基準線の座標値を計算する方法に関するプログラムの記述において大きく相違する。被告プログラムの上下基準線描画部の記述は、原告プログラムの上下基準線描画部の記述と実質的に同一とはいえないし、原告プログラムの創作性を有する本質的な特徴部分を直接感得することもできない。

(c) したがって、原告プログラムの上下基準線描画部の記述についての複製権又は翻案権が侵害されていると認めすることはできない。

(エ) キロ行程の描画部

a 原, 被告プログラムの内容

(a) キロ行程の描画部は, 1キロメートル, 0.5キロメートル, 0.1キロメートルごとに, それぞれ印(マーク)を描画する機能を有する。原告プログラムのキロ行程の描画部は, 描画部ファイル(別紙3)の129から164行目までの36行であり, その構成は, 以下のとおりである。

まず, マークを一定間隔(1キロメートルごと, 500メートルごと, 100メートルごと)で設定するために, ①現在処理中のキロ行程の値を, 一定の間隔の数値(たとえば, 1キロメートルマークについては1000)で除し, その結果を切り捨て, さらに同数値(1000)を乗じ, その計算結果が, もとのキロ行程の値と一致するときには, 1キロメートルマークのシンボルを表示し(プログラムとは別に定義してある図形を読み込む), ②一致しない場合には, 続けて500メートルマークを表示する場合に当たるかを, 同様の方法(ただし, 除したり乗じたりする数値は, 500になる)で判定し, ③一致しない場合には, 続けて100メートルマークを表示する場合に当たるかを, 同様の方法(ただし, 除したり乗じたりする数値は, 100になる)で判定し, ④上記処理を終了後, 現在処理中のキロ行程の値に0.1メートルを加算した後, 上記①へ戻って同じ処理を繰り返し, 現在処理中のキロ行程が, 事前に変数として設定してある「最大のキロ行程」と一致するに至ったとき(キロ行程が画面の右端に達したとき)は, 終了する(甲17, 25)。

(b) これに対し, 被告プログラムのキロ行程の描画部の記述は, 被告の電車線—基準線作成ファイル(別紙4)の410から449行の合計40行であり, その構成は, マークを設置する一定間隔(1キロメートルごと, 500メートルごと, 100メートルごと)の判定をする方法として, rem(レム)関数を用いており, 現在処理中のキロ行程を, 一定の間隔の数値(例えば, 1キロメートルマークについては1000)で除し, その余りが0のときは, 1キロメートルマークのシンボルを表示する(プログラムとは別に定義してある図形を読み込む)という計算方法を採用している(甲18, 25)。

b 原告プログラムの創作性の有無及び対比

(a) 原告プログラムにおける「アルゴリズム」は, そもそも法10条3項3号所定の「解法」として, プログラムの著作権で保護されるものではないし, プログラムとは別に定義(シンボル化)された図形を読み込むという方法も, アイデアに当たり, 著作権法上保護されるものではない。

(b) 仮に, 原告プログラムのキロ行程の描画部の具体的記述に, 創作性が生じると解する余地があるとしても, 前記認定のプログラムの内容に照らし, 創作性の範囲は狭いものというべきである。そして, 被告プログラムと原告プログラムとは, キロ行程の値の評価方法が相違することにより, プログラムの記述において大きく相違する。被告プログラムのキロ行程の記述は, 原告プログラムのキロ行程の記述と実質的に同一とはいえないし, また, 原告プログラムの創作性を有する本質的な特徴部分を直接感得することもできない。

(c) したがって, 原告プログラムのキロ行程の記述について複製権又は翻案権が侵害されていると認めることはできない。

エ 以上によれば, 被告の「電車線—基準線プログラム」が, 原告の「電車線—基準線プログラム」の複製又は翻案である旨の原告の主張は, 理由がない。

(3) シェイプ定義に係る記述について

ア 特殊文字データ

証拠(甲6ないし8, 19, 24, 33, 乙1, 2, 4, 8ないし12)によれば, 以下の事実が認められる。

(ア) AutoCAD文字フォント等の概要

a AutoCADの文字フォント

(a) AutoCADの文字フォントには, 大きく分けて, 半角フォント, ビッグフォント, TrueTypeフォントの3種類がある。ビッグフォントは, 漢字など非ASCII文字を表現するための特殊な形式のシェイプ定義ファイルである。一般的に, コンピューターで扱うすべての文字は, すべて1つ1つ「文字コード」が割り当てられているが, コンピューターの文字コードは基本的には256個しかなく, 漢字等多数の文字を割り振るには不十分であるため, ビッグフォントでは, 2バイトコード(文字コードを2つ連結し, 1つの文字を指定する。)

によって文字を表現することができる。2バイトコードを用いる際には、コンピューターが、1つ目のコードを独自の文字コードであると誤認識しないよう、滅多に使用しない特定のASCIIコードを「エスケープコード（コンピューターが次の文字とあわせて漢字と判断する文字コード）」として、ユーザーが選択することができる。

AutoCADには、ビッグフォントファイルとして、あらかじめBIGFONT.shx及びEXTFONT.shxが添付されており、同ファイル内の文字の定義は、公開されており、これを改変・参照して、文字を独自の仕様とすることが可能である。

(b) ビッグフォントのシェイプ定義ファイルの最初の行は、「*BIGFONT nchars, nranges, b1, e1, b2, e2, . . .」という構文に従って記述（領域宣言）される。

「nchars」は、その記述に続いて定義されている文字定義数の近似値である。

「nranges」は、エスケープコードとして使用される連続範囲の数を指定する。

「b1, e1」は、エスケープコードとして使用される連続範囲における先頭及び最終コードを指定する。

b AutoCADシェイプ定義に係る記述

シェイプ定義ファイルは、AutoCADの規定する文法に従って記述されており、その規約は、AutoCADに標準添付されるマニュアルに記載されている。なお、AutoCADのフォントは、ビットマップのように字形を「点の集まり」で表すのではなく、線で表すベクトルフォントである。

AutoCADのシェイプ定義ファイルのシェイプ記述の構文は、以下のとおりであり、シェイプ定義ファイルの各行には、最大128文字まで記述できる。

```
*shapenumber, defbytes, shapename  
specbyte 1, specbyte 2, specbyte 3...0
```

(a) 「*shapenumber」とは、ファイルに対して与えられる一意な番号（シェイプ番号）で、シェイプファイルと呼び出す場合は、このシェイプ番号を指定する。文字フォントの場合は、ASCIIコードの中の各文字の値に対応する固有の番号が必要である（なお、シェイプ定義ファイルでは、文字以外の図形等のシェイプを記述することも可能であり、その場合には、任意の番号を割り当てることことができる。）。

(b) 「defbytes」は、シェイプを記述するのに必要となるデータバイト数で、上限は、各シェイプにつき2000バイトである。

(c) 「shapename」は、シェイプ名である。

(d) 「specbyte」とは、シェイプ指定バイトであり、各指定バイトは、ベクトルの長さ及び方向又は特別コード番号の1つを定義するコードである。特別コード番号には、0からEまでの15種類があり、例えば、「0」又は「000」は「シェイプ定義終了」、「1」又は「001」は「描画モードをアクティブにする（ペンダウン）」、「2」又は「002」は「描画モードを非アクティブにする（ペンアップ）」、「3」又は「003」はその次に記述されるバイトでベクトルの長さを除し、「4」又は「004」はその次に記述されるバイトでベクトルの長さを乗じるという意味を有する。

シェイプ指定バイトにおいて、各変位点を記述する方法としては、①ベクトルの長さ及び方向コードを3文字の文字列で表現する方法（最初の文字がゼロで、2番目の文字がベクトルの長さを、3番目の文字がベクトルの方向を指定する）と、②特別指定コード「8」又は「9」に続けて、座標値を「(X変位, Y変位)」で指定する方法がある。①の記述方法では、ベクトルの長さとして指定できる有効範囲は1（1単位長）からF（15単位長）の15種類であり、方向を指定できる有効範囲は、0からFまで（90度方向から反時計回りに順に16等分した方向）の16種類である。②の記述方法では、XY変位値として指定できる範囲は、-128から+127までである。

(イ) 原、被告シェイプ定義に係る記述の内容

a 原、被告のフォント定義に係る記述

(a) 原告のフォント定義に係る記述

原告のSUG-BIG1.shp（シェイプ定義ファイル）冒頭

の宣言文は、「*BIGFONT 10369, 3, 05F, 060, 07B, 0A0, 0E0, 0FF」であり（甲19），これは、エスケープコードとして使用される連続範囲として、「5F, 60」「7B, A0」「E0, FF」の3つの外字領域を設定したことを意味する。

原告は、原告のフォント定義ファイルにおいて、AutoCADで標準添付されていない漢字、アルファベット、下付き文字、上付き文字、文章中で用いる特殊記号などについて、独自の定義文を作成している。

(b) 被告のフォント定義に係る記述

被告プログラムの特殊文字には、被告が、AutoCADに添付されているフォントを利用せずに、独自に作成した特殊フォントが118文字ある。このうち、「5F, 60」「7B, A0」「E0, FE」の3つの外字領域内にコードが指定されている文字（原告の特殊文字と共通のシェイプコードを有する文字）で、かつ、原告も、AutoCADに添付されているフォントを利用せずに、独自に作成した特殊文字は、合計54文字である。

54文字のうち、10文字は、原、被告の文字の形状が異なり、したがって、シェイプ定義の変化点の座標値も異なっている。1文字は、変化点の座標値及び筆順はほぼ同じであるが、文字の起点が異なっている結果、文字の形状は重ならない。38文字は、形状は同じで、ほぼ重なるが、筆順（原告のシェイプ定義は、反時計回り又は下から上へ、被告のシェイプ定義は時計回り又は上から下へと記述している。）が異なるため、シェイプ定義中の変化点の座標値は異なっている。38文字のうち1つは、後記文字コード「OF27C」の文字「（（（」である。5文字は、文字の形状も筆順も一致しており、したがってシェイプ定義の変化点の座標値も一致するが、シェイプ定義の冒頭で記述するベクトルの倍率の表示が異なるなど、シェイプ定義全体の具体的記述には異なっている部分がある。上記5文字の形状は、「×」「／」「|」「 \emptyset （上付き小文字のエル）」「 \emptyset （下付き小文字のエル）」である。

b 原、被告の文字コード「OF27C」の記述

(a) 原告のシェイプ記述

文字コード「OF27C」についての原告のシェイプ定義の記述は、以下のとおりである。

「*OF27C, 107, [{B=特殊（（（
2, 3, 5, 4, 3, 2, (8, 7, -3), 1, (8, -2,
2), (8, -2, 3), (8, -2, 4), (8, -1, 6), (8, 1,
6), (8, 2, 4), (8, 2, 3), (8, 2, 2), 2, (8, 0, -3
0), (8, 5, 2), 1, (8, -2, 2), (8, -2, 3), (8, -1,
2), (8, -1, 6), (8, 1, 6), (8, 1, 2), (8, 2, 3),
(8, 2, 2), 2, (8, 0, -28), (8, 5, 4), 1, (8, -2,
2), (8, -2, 4), (8, -1, 4), (8, 0, 2), (8, 1, 4),
(8, 2, 4), (8, 2, 2), 2, (8, 0, -26), (8, 6, 3),
3, 3, 4, 5, 0]

したがって、原告のシェイプ記述においては、描画モードが作動（アクティブ）になった後は、「(-2, 2), (-2, 3), (-2, 4), (-1, 6), (1, 6), (2, 4), (2, 3), (2, 2), (0, -3 0), (5, 2), (-2, 2), (-2, 3), (-1, 2), (-1, 6), (1, 6), (1, 2), (2, 3), (2, 2), (0, -28), (5, 4), (-2, 2), (-2, 4), (-1, 4), (0, 2), (1, 4), (2, 4), (2, 2)」の変化点の順に、線を描画しており、これは、画面上、「(」を下から上方向へ、左側から右方向へと描く動きとなる。

(b) 被告のシェイプ記述

文字コード「OF27C」についての被告のシェイプ定義の記述は、以下のとおりである。

「*OF27C, 95, (((
3, 100, 4, 60, 2, 14, 8, (-8, -25), 2,
5, 8, (7, 27), 1, 9, (-2, -2), (-2, -3), (-2, -
4), (-1, -6), (1, -6), (2, -4), (2, -3), (2, -
2), (0, 0), 2, 8, (5, 28), 1, 9, (-2, -2), (-2, -
3), (-1, -2), (-1, -6), (1, -6), (1, -2), (2, -

3), (2, -2), (0, 0), 2, 8, (5, 24), 1, 9, (-2, -2), (-2, -4), (-1, -4), (0, -2), (1, -4), (2, -4), (2, -2), (0, 0), 2, 6, 8, (23, 0), 2, 14, 8, (-15, -6), 4, 100, 3, 60, 0」

したがって、被告のシェイプ記述においては、描画モードが作動（アクティブ）になった後は、「(-2, -2) (-2, -3), (-2, -4), (-1, -6), (-1, -6) (2, -4), (2, -3), (2, -2), (5, 28), (-2, -2), (-2, -3), (-1, -2), (-1, -6), (1, -6), (1, -2), (2, -3), (2, -2), (5, 24), (-2, -2), (-2, -4), (-1, -4), (0, -2), (1, -4), (2, -4), (2, -2)」の変化点の順に、線を描画しており、これは画面上、「(」を上から下方向へ、左側から右方向へと描く動きとなる。

(c) 文字コード「OF27C」で表示される「(」について、原、被告のシェイプ定義いずれで表示された場合でも、画面上の形状は同一であり、両者は重なり合う。

(ウ) 原告シェイプ定義に係る記述の創作性の有無及び対比等

a 法2条1項10号の2所定のプログラム該当性の有無

原告のシェイプ定義の記述が、法2条1項10号の2所定のプログラムに該当するか否かについて、念のため判断する。

原告のシェイプ定義の記述は、例えば「2」や「0」のような数字から構成され、これにより、「ペンアップ」「シェイプ終了」という動作が電子計算機により行われる。同記述は、AutoCAD上に存在し、シェイプ記述を定義するプログラム（例えば、「2」は「ペンアップ」を意味することを定義するプログラム）に読み込まれ、協働して機能することによって初めて、電子計算機に対する指令としての意味を持つ。そうすると、シェイプ定義の記述は、AutoCADによって読み込まれる情報を記載した単なるデータであるから、それ自体独立し、電子計算機を機能させて一の結果を得ることができるようこれに対する指令の組み合わせたものとして表現したもの」に当たらないと解する余地もなくはない。しかし、たとえ、当該記述が、独立性がなく、個別的に利用することができないものであったとしても、データ部分を読み込む他のプログラムと協働することによって、電子計算機に対する指令を組み合わせたものとして表現したものとみることができるのであるから、そのような記述も、同号所定のプログラムに当たると解して差し支えない。

そうすると、原告のシェイプ定義の記述は、具体的な記述に創作性がある限りにおいて、著作権法の保護の対象になるというべきである。

b 創作性の有無

原告のシェイプ定義の記述方法は、シェイプ定義ファイルを実行するプログラムであるAutoCADによって規定されており、その記述は、ベクトルの始点と終点を定める変位点の座標値(-128から+127まで)又はベクトルの長さ及び方向コードを表現する3文字の文字列と、これらの座標値間の移動を指示する1から10までの特別指定コードの組合せにより表現されている。そして、シェイプ定義に係る文法によれば、特定の形状のフォント又はシェイプを、通常の筆順で記述しようとするれば、その変位点の座標値の記述方法について、制作者の選択の幅は極めて狭いものであるから、シェイプ記述に創作性が生じるものということはできない。そして、原告の特殊文字のシェイプ定義に係る記述は、当該文字を描く筆順として通常の筆順に従っているから、原告のシェイプ記述の座標値の記述に、創作性があるとはいえない。

c 対比

仮に、原告のシェイプ記述の具体的表現に創作性が認められると解する余地があるとしても、上記のとおり、原、被告が独自に作成した特殊文字合計54文字のうち、49文字については、変化点の座標値又は筆順が異なっていることに照らすならば（たとえば、原告が同一であると主張する「(」の原、被告のシェイプ記述の筆順は、被告のシェイプ記述が上から下方向へ書いているのに対し、原告のシェイプ記述は、下から上方向へと書いている点において相違があり、シェイプ定義の具体的記述は相違する。）、両者の具体的表現方法が実質的に同一とはいえないし、また、原告シェイプ定義記述の創作性を有する本質的な特徴部分を直接感得することもできない。

なお、上記54文字中、5文字（「×」「／」「|」「0（上付き小文字のエル）」「0（下付き小文字のエル）」）については、上記シェイプ記述の表現は、ほぼ必然的に決まるというべきであるから、5文字に関するシェイプ定義に創作性が生ずる余地はない。

c したがって、原告の特殊文字コードのシェイプ記述について、複製権又は翻案権侵害があるとは認められない。

イ シェイプコードの割当領域について

原告は、被告プログラムと原告プログラムとは、シェイプコードの割当領域において実質的に同一である旨主張する。しかし、原告プログラムにおけるシェイプコードの割当領域の選択は、アイデアに当たり、その点において、原告プログラムないし原告シェイプ定義記述の創作性を基礎付けることはできない。この点における原告の主張は失当である。

2 争点2（被告は、被告プログラムを作成するに際して、原告プログラム1を、自己のコンピュータ内の記憶媒体等に複製、格納したことがあるか。）について

原告は、被告プログラムと原告プログラムとの間に、共通点が存在することを根拠として、被告が、原告プログラムを被告のコンピュータ内の記憶媒体等に複製して格納したことがあると主張する。

しかし、以下のとおりの理由から、原告プログラムと被告プログラムとに共通する点があったとしても、被告は、被告のコンピュータ内の記憶媒体等に複製する方法以外の方法によって、原告プログラムの内容を認識することができるから、これらの共通点が存在することをもって、被告が原告プログラムを複製したと認めることはできない。

(1) 特殊文字データの変化点の座標値の一致について

原告は、原告プログラムの特殊文字の形状と、被告プログラムの特殊文字の形状の90%がほぼ完全に一致していることをもって、被告が原告プログラムを複製したことがある旨主張する。

しかし、原告プログラムの特殊文字の形状は、原告プログラムで使用されている「フォント一覧表」を作成するLISPプログラムを、原告プログラム上で実行し、作成した「フォント一覧表」を見ることにより認識できるものであるし（乙4、8、10）、上記認定のとおり、原、被告が独自に作成した54文字のうち、シェイプ定義（変化点の座標値）が完全に一致するものは5文字しかなく、その5文字は、いずれも座標値が一致しても不自然ではないような簡単な形状のものであるから、原、被告の特殊文字の形状が一致していることをもって、被告が原告プログラムを複製したと認めることはできない。

(2) シェイプコードの割当領域の一致

原告は、原、被告のシェイプコードの割当領域が、ほぼ完全に一致していること及び当該割当領域に割り当てられている特殊外字が、2重定義されている部分も含めて、すべて一致していることをもって、被告が原告プログラムを複製したことがあると主張する。

しかし、証拠（乙4、8、10）によれば、原告プログラムの特殊文字のシェイプコード及びその割当領域は、原告プログラムにおいて使用されているフォントを表示した「フォント一覧表」を見ることにより認識することができ、被告は、原告プログラムで作成された図面を被告プログラムで開くことができるようにするために、同表に基づき、原告プログラムに存在する特殊文字に対応する特殊文字に、原告プログラムにおいて割り当てられたシェイプコードと同じシェイプコードを割り当てたことが認められるから、原、被告の特殊文字のシェイプコードの割当領域が一致していることをもって、被告が原告プログラムを複製したと認めることはできない。

(3) 補助線の数の一致

原告は、被告の電車線—基準線プログラムで描画される横の補助線（上下基準線）の数が、原告の補助線の数と一致していることをもって、被告が、原告プログラムを複製したことがある旨主張する。

しかし、上下基準線の数は、原告プログラムによって作成された図面を見ることによって容易に分かることであるし（甲3）、被告が、JR東日本の便宜のために、従前の図面と一致する数とすることは、何ら不自然なことではないから、このことをもって、被告が、原告プログラムを複製したと認めることはできない。

(4) エスケープコードの一致

原告は、被告プログラムが使用するエスケープコード（「””」）が、原告プログラムで使用されているコードと同じであることをもって、被告が、原告プログラムを複製したことがある旨主張する。

しかし、原告が使用するエスケープコードは、当該エスケープコードを使用して作成した原告プログラムを、被告プログラム上で開けば、エスケープコードとしての役割を果たさずに、そのまま当該コードの文字として、画面上に表示されるものであるから（乙4）、原告プログラムを複製することなく知り得るものであるし、被告が、ユーザーであるJR東日本が操作する上で便利のように、原告プログラムと同じコードをエスケープコードとすることは、何ら不合理なことではないから、エスケープコードが一致していることをもって、被告が原告プログラムを複製したと認めることはできない。

(5) 改行コードの一致

原告は、被告プログラムが使用する改行コード及び改行量が、原告プログラムが使用する改行コード及び改行量と同じであることをもって、被告が、原告プログラムを複製したことがある旨主張する。

しかし、原告が使用する改行コードは、当該改行コードを使用して作成した原告プログラムを、被告プログラム上で開くと、可読不可能な文字（「・」）として画面上に表示されるが、そのコードは、当該文字を他のアプリケーションに貼り付けることによって調べることが可能であり（乙4）、その改行量も（原告の改行コードのシェイプ定義では、変位点の座標値が（0, -30）であり、これは下方向へ30単位移動することを意味する。）、画面上の表示内容から、原告の改行コードを用いた場合の改行量と同じ改行量を探すことは可能であるし、被告が、JR東日本の操作上の便宜のため、原告プログラムと同じコードをエスケープコードとし、改行量を一致させること自体は、何ら不自然なことではないから、エスケープコードが一致していることをもって、被告が原告プログラムを複製し、解析したと認めることはできない。

(6) 変数に値を設定する順序の一致

原告は、被告の初期設定部のプログラムでは、変数に値を設定する順序（「キロ行程最初の値」「キロ行程オフセット値」「縮尺」「用紙サイズ」の順）が、原告プログラムで変数に値を設定する順序と同じであることをもって、被告が、原告プログラムを複製したことがある旨主張する。

しかし、上記順序は、変数を設定する順序としては、何ら不自然なものではなく、この点が一致していることをもって、被告が原告プログラムを複製したと認めることはできない。

(7) 以上のとおり、原告が主張する各一致点をもって、被告が原告プログラムを複製したことがあると認めることはできず、その他、原告の主張を認めるに足りる証拠はない。

したがって、この点に関する原告の主張は理由がない。

3 以上によれば、原告の請求は、その余の点を判断するまでもなく、いずれも理由がない。よって、主文のとおり判決する。

東京地方裁判所民事第29部

裁判長裁判官 飯村敏明

裁判官 今井弘晃

裁判官 大寄麻代

(別紙)
被告製品目録

原告プログラム目録

1 AutoCAD GXⅢ版JR-CAD
「電車線-基準線作成プログラム」のうち、メニュー表示部のプログラム、入力部のプログラム、描画部のプログラムは、それぞれ別紙1ないし3のとおりで

ある。

2 AutoCAD R13J版JR-CAD II

別紙1 別紙2 別紙3

被告プログラム目録

1 AutoCAD R13J版 Quite 鉄道編

「電車線-基準線作成プログラム」のうち、メニュー表示部のプログラム、
入力部のプログラム、描画部のプログラムは、別紙4のとおりである。

2 AutoCAD R14版 Quite 鉄道編 Ver1.1

3 AutoCAD 2000i版 Quite 2000 鉄道編 Ver1.0

別紙4