

情報技術イノベーション促進と 著作権エンフォースメントの 調整法理としての fair use

—Google v. Oracle 事件連邦最高裁判決を基点とした検討

南山大学法学部法律学科 教授 平嶋 竜太

要 約

2021年4月にアメリカの連邦最高裁で下された Google v. Oracle 事件判決は、コンピュータ・プログラムのアプリケーションインタフェースの利用を巡って、アメリカ著作権法における fair use 法理を適用して注目される判断を示した事案である一方で、日本法に対しても多くの示唆をもたらさうる事案であると考えられる。本稿は、情報技術を実装する上で不可欠な要素であるソフトウェア技術に係る著作権のエンフォースメントと情報技術イノベーションのあり方を考察する基点として、同判決の検討と課題の分析、さらに日本法の文脈に適用した考察を行うことで、情報技術分野におけるイノベーションの促進と著作権のエンフォースメントに係る調整法理のあり方について、その方向性を様々な観点から模索することを主たる目的とするものである。

目 次

- 1 検討の方向性
- 2 Google v. Oracle 事件について一法的紛争の背景
- 3 Google v. Oracle 事件連邦最高裁判決
- 4 連邦最高裁判決の評価と課題
- 5 日本法に置き換えた議論—APIの著作物性評価と著作権行使
- 6 展望—情報技術イノベーション促進と著作権エンフォースメントの調和

1 検討の方向性

近時の情報技術イノベーションの進展スタイルは、機械学習技術を基礎とした AI 技術や P2P 及び分散 ledger の技術を基礎としたブロックチェーン技術といった新たな技術を基にした発展を中心に牽引される側面、IoT やビッグデータ利用、メタバースの展開といった従前の技術を基にして社会への幅広い浸透・拡張を中心に牽引される側面、と多様化している。知的財産法制を構成する各種の知的財産権のエンフォースメントのあり方が、これらの情報技術分野におけるイノベーション進展に及ぼさうる影響についても多様化しているものと考えられる。また、いずれの側面においても、情報技術を実装するテクノロジーとしてのコンピュータ・ソフトウェアは不可欠な要素であるところ、ソフトウェアは、知的財産法制の下では、特許発明としての保護に加えてプログラムという表現形式についても著作権法による保護が及ぶという点で、他の技術分野に比べて特徴的であることが指摘できる。

ソフトウェア技術に係る著作権のエンフォースメントと情報技術イノベーションのあり方を展望する上で、2021年4月にアメリカの連邦最高裁で下された Google v. Oracle 事件判決は、アメリカ著作権法における特徴的な法理である fair use 法理を適用した判断事案であるにもかかわらず、日本法に対しても多くの示唆をもたらさうるものと考えられる。

本稿は、Google v. Oracle 事件連邦最高裁判決の検討と課題の分析を行うことを契機として、情報技術分野におけるイノベーションの促進と著作権エンフォースメントの調整のあり方について、その方向性を模索することを主たる目的とするとともに、併せて関連事項についても展望することを目指すものである。

2 Google v. Oracle 事件について一法的紛争の背景

2-1 事案

Google v. Oracle 事件は、Java 言語によるプログラム開発ツール（開発プラットフォーム）である Java SE の著作権者である原告 Oracle America, Inc. が、スマートフォン用 OS 及びプラットフォームである Android の開発をした被告 Google に対して、Android 上で動作するアプリケーションプログラム⁽¹⁾ 開発環境において、Java SE を構成している API（Application Programming Interface、アプリケーションプログラムインタフェース）における宣言コード部分を Google が複製して、Android の開発プラットフォームに取り入れて利用したことが著作権侵害等に当たるとして民事訴訟を起こした事案である。

2-2 背景状況

本件事案における原告著作権の基礎となる表現は、プログラミング言語 Java に関するものである。Java という語については、ソフトウェアのプログラミングに用いられるプログラミング言語体系としての Java 言語自体と Java 言語を用いたアプリケーション開発環境の両者を意味する場合で一般的には用いられているといえる。なお、Java 言語は 1995 年に Sun microsystems によって元々開発されたもので、その後、Sun が企業買収されたことに伴って本件事案の原告である Oracle America, Inc. が継承し、著作権者となったものである。2010 年に Oracle は Sun を買収した後、まもなく本件訴訟を提起した。

Java 言語による開発環境の下で開発されたプログラムのもつ最大の特徴としては、‘write once, run anywhere.’ が挙げられる。Java が開発される以前までの既存のアプリケーションソフトウェアの場合には、特定の OS（windows, iOS, Linux 等）をプラットフォームとして動作することを前提として作成されていたために、同一内容のアプリケーションソフトウェアであっても、異なる OS 毎に異なるプログラムの開発が必要であったところ、Java 言語で開発されたソフトウェアについては、これと異なり、JVM（Java Virtual Machine）上で中間コードを介してコンピュータハードウェアを動作させる構造をとっていることから、特定の OS の種類に依存することなく動作するアプリケーションソフトを開発することが可能となった。このため、個別ユーザの OS 環境に捉われず広く動作可能なアプリケーションソフトの開発が可能⁽²⁾ となり、インターネットの商業化とも相まって急速に利用が拡大し、さらに、多様なアプリ開発に際して非常に大きな影響をもたらした。それと共に、Java 言語をアプリケーション開発に用いるプログラム開発者も増加するようになってきた。現状でも、Java はデスクトップ・Web 系のアプリはもちろん、組み込み装置用のアプリ、スマホ系のアプリ等の開発に極めて広く用いられている言語であって、アプリケーション系のプログラマ人口でも Java 言語系を用いるプログラマの割合は圧倒的に多数とされている。なお、Android を OS のベースとするアプリケーションプログラムの開発言語としては、近時はあらたに開発された言語である Kotlin を用いて開発される例が急速に増えている⁽³⁾ 状況にあるが、Java 言語による開発環境は、元々オープンソースソフトウェアとして広くプログラム開発者に開放されていたこともあって、アプリケーション開発者の間で依然として非常に幅広く利用されている状況にある。

Java 言語を用いたプログラム開発環境としては、基本的には Sun microsystems が開発した Java SE と

(1) 以下、略記としてアプリという語も同義の語として用いる。

(2) 本来はネットワーク家電等への利用を目的として開発されていたようである。

(3) Google も Kotlin を 2019 年に Android アプリ開発の推奨言語に指定している。

いうプログラム開発ツール（プラットフォーム）を用いる形となっている⁽⁴⁾。Java SE は、基本的には Java 言語でのプログラム開発を行うためのツールを呼び出す API とのセットで構成されている。

Java SE の標準仕様については、Sun が設立した団体である Java Community Process (JCP) を通じて策定されるものの、策定過程自体はオープンにされており、Java SE を用いてアプリケーションプログラムを開発（実装）する事業者（ベンダー）がメンバーとして JCP に参画して標準仕様の策定に関わることが可能とされていた。むしろ参加によって標準仕様の内容が調整され、適宜アップデートされて取り入れられる形がとられてきた。このため、それぞれのベンダーは、この標準仕様を自らの製品に実装しつつ、一部については独自の仕様を実装して、差別化を図った製品を販売することが可能であった。ただし、ベンダーによる個々の製品について、Java 標準仕様に準拠していることを謳うためには、Sun による有償の認証を受けることが必要とされており、この認証に適合すると所定のロゴ使用が認められるものとなっていた。すなわち、Sun のビジネスモデルは、Java SE 自体等の利用に対するライセンスによるビジネスではなく、Java SE の標準仕様に適合している製品であることについての認定ライセンスを提供することをビジネスの柱にしていたともいえる。

他方、Java SE では多様なベンダーが標準仕様の策定にコミットできる機会を設けていたことによって、皮肉にも標準仕様自体のサイズは肥大化する一方で、Java 標準の認証を確保しつつも小回りの利くような実装を実現することは次第に難しくなってきた。そのような流れの中で、2005 年ごろになってオープンソースコミュニティの Apache が Java SE をオープンソース化したバージョンの開発を行うプロジェクト (Project Harmony) を提案して、Apache ライセンスの下で提供される Java SE に相当する開発プラットフォーム (Apache Harmony) の開発が、IBM や Intel の後押しもあって Apache Software Foundation (ASF) を中心として開始されるようになった⁽⁵⁾。

これに対して、Sun も Java SE のオープンソース化を提案したものの、開発したコードについて広く公開義務が生じるオープンソースライセンスである GPLv2 の条件下で提供されるオープンソースソフトウェアとしたために、Harmony 陣営が強く反発して紛糾する事態となり、最終的には、両者は Sun 自ら開始したオープンソース化された Java 開発プラットフォームである OpenJDK と統合することとなった⁽⁶⁾。

2-3 Google による行為

被告である Google は、スマートフォン用 OS 及びプラットフォームのために、スタートアップ企業 Android を買収して、2005 年ごろから、スマートフォン用 OS 及びプラットフォーム (Android) の開発において、Android 上で動作するアプリケーションプログラム開発環境について、できるだけフリーかつオープンなアプリケーション開発が可能な環境を開発することを目指していた。その当時、上記のような特徴から非常に多くの Java 言語によるアプリケーションプログラム開発者（当時で 600 万人程度とされる）によるアプリケーション開発が普及していたことから、Java 言語による開発環境に慣れた開発者にとってのアクセシビリティが重視されたといえる。

Google は、Android における開発環境として、自由なアプリ開発がなされるプラットフォームとなることを前提に、Java SE の携帯端末や家電用の小型版である Java ME (MicroEdition) をオープンソース化したものとするを想定していたという⁽⁷⁾。このため、Google がスマートフォン用 OS プラットフォームで

(4) これ自体については基本的にオープンソースソフトウェアとして無償利用可能である。ただし、Oracle がサポートしている Java 開発プラットフォーム (JDK) の一部については 2019 年から有償化されている。もっとも無償版もあり、主にバージョンアップ後のサポート期間等で差別化が図られているようである。

(5) Apache ライセンスによって提供されることにより、この場合は開発された個々のプログラムについては、コードの公開義務から逃れられる。他方、GPL ライセンスの下で提供される場合、コードの公開義務から逃れることができなくなる。

(6) 他方、Oracle (Sun) は元々の Java 開発プラットフォームとして JDK (Java Development Kit) も提供をしている。

(7) 当初、Google は Sun から認定を受けるつもりであったという話もあるようである。

の利用のために Java プラットフォームのライセンス供与について交渉されたが、結果的に決裂した⁽⁸⁾。そこで、Google は、100 名のエンジニアと 3 年以上の開発期間を費やし、独自に Android についての開発プラットフォームを開発するに至った。

すなわち、Android 自体についてのプログラムを実装しているコード自体は Google が独自に作成したものである。とりわけ、Java において中心的な役割を果たす JVM に相当する部分については、完全に独自に Dalvik というコンポーネントを開発している。しかしながら、Java SE を構成している API の宣言コード部分については、おおよそ 11500 行ほど複製して、Android の開発プラットフォームに取り入れて利用した⁽⁹⁾。これはそもそも Android が Java をベースにした開発環境とすることを念頭に置いていたということもあろうが、本件判決でも議論の対象となるように、Java での開発環境に慣れているプログラム開発者によって OS たる Android 上で動作するアプリケーションプログラム開発をやすくすることで、より多くのアプリ開発への参入やユーザの増加が期待されることも考慮したものと考えられる⁽¹⁰⁾。

ところで、API とは、本件連邦最高裁判決における言い回しによると、「プログラム開発者がある機能をプログラムとして作り込む際に、最初からすべてのプログラムコードを自らあらたに書く (scratch) のではなく、あらかじめコードとして書かれているものを利用することを可能とするツール」とされている。すなわち、プログラム開発者は API を用いることによって、複雑なタスクをコンピュータに実行させるプログラムを開発するに際して、あらかじめ書かれているコードのライブラリを基に適宜組み合わせることでプログラムを記述することが可能となり、ゼロから書き始める必要があるコードの記述量を減らせることになって、プログラム開発を大幅に効率化することが可能となる。本件連邦最高裁判決では、「あるロボットに、特定のファイルキャビネットに向かい、ある引き出しを開けて、個別のレシピを取り出すように指示したとしよう。ロボットは、そのレシピをもってキッチンへ移動して、そこで調理をするコックにそのレシピを渡す。このような例が、API によるタスクについての組織的なシステムに相当する。ここで、簡単なコマンドをロボットに指示することで、ロボットは求めているレシピをコックへと手渡すことになる。ここで重要なことは、特定の料理についての詳細なレシピの内容を我々は知っておく必要はないのと同様に、API を使うに際してメソッドについての実装コードを知っている必要はないということである。」という例示的な説明がなされている。

本件連邦地裁判決で取り上げられた具体例では、2つの数字のうちいずれが大きいのかを示すプログラムを Java 言語で記述する場合、最初に `java.lang` を書くことされている。これはパッケージを指定するものであって、先のロボットの設例におけるキャビネットに相当する。さらに、`Math` を記述して、これはクラスを指定するものであって、先のロボットの設例における引き出しに相当する。さらに、`max` を記述して、これはメソッドを指定するものであって、先のロボットの説明におけるレシピに相当する。そこで、カッコを記述して、比較対象とする具体的な 2つの数字、例えば 4 と 6 をカッコに入れることになる。いままで指定した指示全体の表現 (メソッドコールと呼ばれる) としてみると、`java.lang.Math.max(4, 6)` となる。このような表現とすると、API によって、2つの数字のうち、より大きな数字がいずれであるのかを判断するようなタスクを実行するための実際のプログラムが呼び出されることになる。プログラム開発者は、このような記述を作成するだけであるが、ここでの表記自体がコンピュータハードウェアに対して何らの処理を

(8) Google は Java SE の一部の API だけのライセンス供与を求めたところ、Sun はこれを拒否したともいわれる。また、当時、Google は上記のオープンソース系の Apache Harmony をベースに Android 開発を行っていたということも背景に考えられる。

(9) Java SE に含まれる主要な部分ともいわれるが、Apache Harmony をベースに開発していたことから、そこに含まれていた API のうち Android の動作環境である携帯電話関係に必要なものだけを取り出して活用したようである。

(10) あるいはまた、元々はオープンソース化されていた Apache Harmony をベースに開発しており、そこから API を抜き出してきたとすると、Google 自体は、本件で問題となった Java SE の API に対応する実装コードについても実は必ずしも完全にすべて独自に開発をしていなかった可能性も皆無ではないとも考えられるが、真意のほどは不明である。

行っているわけではない。これらの表記とファイルキャビネット、引き出し、ファイルに相当するものとを結びつけるソフトウェアが必要となる。これがAPIというソフトウェアに当たる。つまり、個々のAPIについては、「上記のようなメソッドコールと特定のタスクに係る実装プログラムを結びつける宣言コード」および「実際に実行を行う実装コード」の両方を含むものから構成されているといえる⁽¹¹⁾。

ところで、APIという言葉自体は多義語であり、OSやアプリケーションプログラムが他のソフトウェアの有する機能を呼び出したりする場合のインタフェース（の仕様）を意味したり、近時では、単にAPIという場合、Web上でHTTPを通じて他のサーバ上のプログラムで提供されているサービスを用いて処理をさせるためのインタフェース（WebAPI）を示している場合も少なくない。本件事案におけるAPIとは、そのような意味ではなく、より細かいレベルで、例えば一つのアプリケーションソフトウェア内での処理を実現するソフトウェアの部品（コンポーネント）を相互に呼び出しあって、やり取りするためのインタフェースを意味するものである。APIの法的保護という問題を考えるに際して、APIといっても具体的にどのようなものを前提とするのかによって、法的な意味での議論内容も大きく異なってくる可能性があることには注意を要するであろう。

ところで、Java APIにしても、AndroidのAPIにしても、APIの宣言コードはそれぞれのAPIの構造において主要な部分を構成する要素といえるものであるが、Googleは、Java APIから37のパッケージについての宣言コード部分をそのまま複製していた。すなわち、この37パッケージについては、特定のタスクに対して個別に与えられている名称と、これらの多数のタスクを一定のまとまりをもったクラスやパッケージにグループ化するやり方、の両者について複製をしていたといえる。これによって、Java言語を用いたプログラム開発者としては、既に知っているタスクをその名前を用いてコールする仕組みを使って、Android上で動くアプリのプログラム開発がしやすくなったことになる。すなわち、Androidプラットフォームにおけるプログラム開発に際しても、特定のタスクをコールするに際して、Javaと同じ宣言コードを使うことができることによって、プログラマは既に慣れ親しんだメソッドコールを使うことができる。逆にみれば、仮にJava APIの上記部分のコピーがAndroidのシステムにおいてなされなかったとすれば、Android上で動作するアプリケーションプログラムの開発を行う者は、Java言語で記述されている内容と全く同様のタスクをコールするに際しても、（Android固有に用意された）別途の名称のメソッドやクラス、パッケージの名前を最初から覚えなおす形で、全く新たなシステムの構成のやり方を新たに習得する必要があることになるわけである。

Googleが宣言コードを複製した37のパッケージには、モバイルデバイス用のアプリケーションを開発す

(11) この点について、よりくだけた例を基に説明をすると、例えば次のようになるであろう。

いま、タスクを「(コーヒーが入っている)カップを持ち上げて(コーヒーを)飲む。」とする。

クラス「物体を移動させる動作系」として、以下のメソッド1-3を含むものとする。

メソッド1 = 「水平」 - 物体を (口のある位置の近傍まで) 水平に移動させる。

メソッド2 = 「垂直上げ」 - 物体を (口の高さに近い適切な高さまで) 垂直に持ち上げる。

メソッド3 = 「傾斜」 - 物体を (カップのふちが開いた唇に接触する適切な角度まで) 傾ける。

■部分の具体的内容については適切な対象物や数値範囲をその都度指定すればよい。

以上のようなメソッドを前提とすれば、まったく異なるタスクを構成する際にも、一からメソッドを新規に作成せずとも、メソッド1-3、すなわち「水平」「垂直上げ」「傾斜」は利用できることになる。

例えば、「床にある衣類ケースを棚にしまう。」というタスクを構成する場合にも、クラス「物体を移動させる動作系」からメソッド1及び2を呼び出して、衣類ケースをメソッド1「水平」で移動させて、メソッド2「垂直上げ」で床から持ち上げて、棚板までメソッド1「水平」で移動させれば、新たに「垂直に下す」というメソッド4「垂直下げ」を付け加えるだけで、タスクを構成することができる。

このとき、メソッド1-3について、例えば、それぞれ「水平」「垂直上げ」「傾斜」という名称を付けた場合に、これらの記述に著作物性はあるのか否かという問題こそが、本件事案で議論になっている著作物性の問題を極めて端的に示しているものと考えられる。

これらのメソッドの記述として、上記以外にも「横に移動する」「縦に上げる」「傾ける」あるいは「ヨコスベリ」「アップ」「ナナメ」といった他の表現で記述する選択肢は（無数に）あることも確かといえるであろう。

るに際して非常に有用なタスクが含まれていた⁽¹²⁾。これらの個々のパッケージにはさらに数多くのクラスが含まれており、さらに個々のクラスには多数のメソッドが含まれている階層的な構造となっている。そしてそれぞれについて宣言コードが付与されていることになることから、Google は 600 以上のクラス、6000 以上のメソッドを複製して利用したことになる。ただし、これらの宣言コードに対応する実装コードについては複製しておらず⁽¹³⁾、Google が独自に作成したものによってハードウェアを動作させてタスクを実行することになる。

Android のプラットフォームは 2007 年に最初にリリースされ、成功をおさめてアメリカ国内における膨大なシェアを獲得し、2015 年までに 420 億ドル以上の収益を達成しているという。

2-4 訴訟経緯

訴訟の流れとしては、最終的に連邦最高裁判決に至るまで 2 期に分ける見方もあり、複雑な流れをとっている。

まず、下級審での争点として、当初は、Oracle は著作権侵害、特許権侵害の両者で提訴し、連邦地裁では、著作権侵害、特許権侵害、損害額算定の 3 つに分けて審理され、陪審では特許権侵害の主張は否定され、著作権侵害の成立だけを限定的に肯定⁽¹⁴⁾して、fair use については判断されなかった。また、連邦地裁判決では、API の宣言コードは著作権法による保護が及ぶものではないと判断した。Google は Java 言語に慣れたプログラム開発者がタスクを使いやすいように宣言コードや組織構造だけを複製したのであって、そのような部分は、102 条 (b) における「システムや操作方法」であって著作権法では保護されないと判断した⁽¹⁵⁾。

これに対して控訴審では、CAFC はこの判断を覆した。すなわち、API の宣言コードと組織的構造 (SSO) のいずれもが著作権法の下で保護されると判断した。Google は実装コードを自ら書いたと同様に、宣言コードについても自ら書くことができたはずであったし、Google は、プログラム開発者によってコールされるタスクを分割してラベル付けして、全く新たなシステムを作り出すことも基本的には行えたといえるのであるから、Java API を構成した宣言コードは著作物として保護されるものであったといえる、等を理由としている。Java API を利用することが fair use に当たるかについての判断も併せて求められていたが、この点については、判断は示されなかった。CAFC では、fair use とは著作権法の硬直的な適用を避けるために認められるものであるとして、fair use の主張を審理するに足る事実があるものではないとして、この点について更なるトライアルを行うべく、連邦地裁に差戻しをした⁽¹⁶⁾。Google は、CAFC による著作物性の判断に対し、certiorari の受理申立てをしたが、認められなかった⁽¹⁷⁾。

差戻し後、連邦地裁において、トライアルがなされた。Google が複製した 37 の Java API パッケージに含まれる宣言コード部分および Java API の構造の Android における利用は著作権法の下での fair use を構成するのか否かという点に関して、陪審は肯定的な回答を示した。そこで、Oracle は再度 CAFC に控訴した。CAFC の判決では、Java API についての著作物性を肯定し、すべての事実問題については Google による

(12) 地裁の認定でも、これらパッケージのうち 3 つのパッケージは Java 言語の利用にとって、もっとも基本となるものであったことが認められている。具体的には以下のパッケージである。

java. awt. font, java. beans, java. io, java. lang, java. lang. annotation, java. lang. ref, java. lang. reflect, java. net, java. nio, java. nio. channels, java. nio. channels. spi, java. nio. charset, java. nio. charset. spi, java. security, java. security. acl, java. security. cert, java. security. interfaces, java. security. spec, java. sql, java. text, java. util, java. util. jar, java. util. logging, java. util. prefs, java. util. regex, java. util. zip, javax. crypto, javax. crypto. interfaces, javax. crypto. spec, javax. net, javax. net. ssl, javax. security. auth, javax. security. auth. callback, javax. security. auth. login, javax. security. auth. x500, javax. security. cert, javax. sq

(13) ただし、争いのない事実によれば、9 行のコードから成る rangeCheck 関数、8 つのデコンパイルされたセキュリティファイルは除くものとされており、すなわち当該プログラムについては実装コードを含めて複製していることになろう。

(14) 37 パッケージ及び“rangeCheck”関数と呼ばれるルーチンについての著作権侵害に限って肯定した。

(15) Oracle Am., Inc. v. Google Inc., 872 F. Supp. 2d 974 (N.D. Cal. 2012)

(16) Oracle Am., Inc. v. Google Inc., 750 F. 3d 1339 (Fed. Cir. 2014)

(17) 576 U. S. 1071 (2015)

主張を肯定的に捉えることを前提としつつも、他方で、これらの事実により fair use を構成するものであるか否かという問題は法律問題であるとして、Google による Java API の利用行為について、「著作物の表現そのものを用いて、競合するプラットフォームにおいて元の著作物と同一の目的および機能のために利用することについてはフェアと評価する要素はどこにもない。」として、著作権法の下での fair use には当たらないとの判断を示した。そして、再度連邦地裁に差し戻して、損害額算定のため、再度トライアルに付されることとなった⁽¹⁸⁾。これにより、Google 側の行為が著作権侵害と判断される可能性が高まったといえる。

これに対して Google は、再度、連邦最高裁に対して、certiorari の受理申立てをした。すなわち、以下の2点、(1) Google が複製した当該プログラム部分に著作物性はあるのか？ (whether Java SE's owner could copyright the portion that Google copied.), (2) 当該部分についての著作物性があるとして、Google による複製行為は fair use を構成するのか？ (if so, whether Google's copying nonetheless constituted a "fair use" of that material, thereby freeing Google from copyright liability.), についての判断を求めるものであって、連邦最高裁はこれを受理した⁽¹⁹⁾。

3 Google v. Oracle 事件連邦最高裁判決

3-1 多数意見の概要

連邦最高裁判決は、2021年4月5日に下され、原審であるCAFCがGoogleの行為は著作権法上のfair useに当たらないとした判断を破棄し、差し戻す判断を示した⁽²⁰⁾。Breyer判事をはじめとする多数意見が6名、Thomas判事及びAlito判事の2名が反対意見、Barrett判事は審理及び判決に加わらなかった。

多数意見では、まず、著作権法の基本的な概念と本件事案の争点、fair use 法理の位置付け及び考慮されるファクターについての確認を行って、「急速に変化する技術、経済、ビジネス環境を所与とすると、当事者の議論を解決するに際して必要となる以上の回答を行うべきではないものとする⁽²¹⁾。」として、本件判決の「射程」を捉える上でも示唆的ともいえる姿勢を示している。その上で、「純粹に議論の目的のため、Java APIは著作権で保護される対象の定義に属するものとする⁽²²⁾。」として、certiorariにおいて判断が求められた第一点である、Java APIの著作物性について具体的な判断を提示するものではなく、あくまでも著作物性があるという仮定の下に判断を示す姿勢を明らかにした。そして、「他方、GoogleによるAPIの部分的な利用がfair useに当たるかについて検討することとする。CAFCの結論と異なって、これはfair useに当たると結論する⁽²³⁾。」「当該部分の著作物性があるとしても、Googleによる複製利用行為についてfair useは成立するため、Googleの行為は著作権侵害行為ではない⁽²⁴⁾。」として、Googleによる行為がfair useに該当することで著作権侵害行為は成立しないという構造によって結論を導いている。

次に、多数意見におけるfair useの判断構造を検討する。

まず、fair useについての一般論として、アメリカ著作権法107条のfair useの条項は、判例⁽²⁵⁾を起源とすることから柔軟に適用しうるものであって、文脈によってその適用は変化するため、当然ながら著作物の

(18) Oracle Am., Inc. v. Google Inc., 886 F.3d 1179 (Fed. Cir. 2018)

(19) Google LLC v Oracle America, Inc., 140 S. Ct. 520 (Mem.) (2019)

(20) 593 U.S. ___ (2021), 141 S. Ct. 1183 (2021)

(21) "Given the rapidly changing technological, economic, and business-related circumstances, we believe we should not answer more than is necessary to resolve the parties' dispute."

(22) "We shall assume, but purely for argument's sake, that the entire Sun Java API falls within the definition of that which can be copyrighted."

(23) "We shall ask instead whether Google's use of part of that API was a "fair use." Unlike the Federal Circuit, we conclude that it was."

(24) "we assume, for argument's sake, that the material was copyrightable. But we hold that the copying here at issue nonetheless constituted a fair use. Hence, Google's copying did not violate the copyright law."

(25) Folsom v. Marsh, 9 F. Cas. 342, 348 (No. 4,901) (CC Mass. 1841)

種類によって、創作的なものか実用性の高いものかといった点によっても変わりうる。そして、著作物を充足しない事柄に制約を受ける場合には著作権法による保護は「薄い」ものとなるとされてきたこと、一般的には、コンピュータプログラムについては、機能的な目的に供されることから言語著作物に比して性質が異なっている⁽²⁶⁾こと、CONTUにより著作権法による保護が望ましいとされたものの、必要以上に保護を与えることへの注意や著作権法上の様々な法理によりイノベーションを阻害しないことが確認されたことを提示する。そして、fair use 法理も保護の範囲を画定するに際して重要な役割を果たすものであることを確認している。

さらに、fair use の成否判断が法律問題であるのか事実問題であるのか、という議論については、事実問題と法律問題が混在した問題として位置付け、裁判所は陪審の事実認定にしたがうべきであるが、その事実が fair use に当たるかという最終判断は裁判所によってなされる問題であるとした（原審である）CAFC の理解に同意するとしつつも、本件の場合には、法的問題が主たるものといえるとした。もっとも、fair use を適用するに際しては、例えば市場への影響の程度やどの程度の量が複製されたのかといった形で、従属的に事実問題についての判断を伴うことを指摘している。そして、本件では CAFC は慎重にこれらの分類を行った上での判断を行っているといえるのであって、CAFC の判断は合衆国憲法修正第 7 条に違反するという Google の主張⁽²⁷⁾は当てはまらないとした。

以上のような前提を踏まえて、Google による API の複製行為は fair use を構成するのか、という点について、アメリカ著作権法 107 条で規定されている 4 つのファクターそれぞれにつき、第 2 ファクターから検討を行って判断を示している⁽²⁸⁾。

第 2 ファクター（著作物の性質）について。

まず、Java API はユーザインタフェースであって、ユーザであるプログラム開発者が目的とするタスクを実行するプログラムを効率的に作成できるようにするための手段といえ、そのタスク自体について著作物が生じる余地はないことを確認している。また、Java API はメソッドコールというコマンドで個々のタスクを呼び出すものであるが、Oracle はこのようなコマンドの利用自体がなされることにつき著作権侵害を争っているものではなく、Java API は、実装コードを含むコンピュータ内の特定の「場所」とメソッドコールを結びつけるコードである宣言コード部分を含んでおり、Google は実装コードについては自ら作成していることから、Oracle は宣言コード部分の Google による複製利用について著作権侵害と主張しており、宣言コード部分は、プログラム的一种ではあるが、他のプログラムコードとは異なって、（著作物とは主張しえない）システム全般やタスクの制約を密接に受けるものである⁽²⁹⁾こと、そして、実装コードにも密接に制約される⁽³⁰⁾ことについて指摘する。そこで、宣言コードは、その創作性の観点からは、実装コードとは異なる種類のものであって、Sun 開発者たちは直感的に思い出しやすいような名前を付けることにより、プ

(26) その違いについては、「ピースがかみ合わないジグソーパズルを合わせるようなものだ」とされることもある。（Lotus Development Corp. v. Borland Int'l, Inc., 49 F. 3d 807, 820 (CA1 1995) (Boudin, J., concurring)）

(27) 陪審による事実認定の権利を害して、裁判所が事実問題について審理していることを根拠とした主張である。

(28) なお、反対意見を示した Thomas 判事は、著作物の性質を考慮するに際しては、プログラムというタイプのなかで区別をしないのが筋であって、fair use における判例では事案の文脈に応じるものとされているとしても、fair use 法理の分析に際しては、あるファクターが他のファクターよりも優先して検討されるべきである、という見解を示している。これに対して、多数意見では、プログラムが fair use の適用に際して他の著作物と異なった扱いをされるべきものであるとは理解していないし、他の著作物と同様に規定されて、fair use の適用においてもしかりとの見解を示している。

(29) キャビネット、引き出し、ファイルにタスクを組織的に配列するというアイデア、すなわち著作物性のないアイデアの制約を密接に受け、プログラム開発者に知られている特定のコマンドの利用、メソッドコールとして知られているものに密接に制約を受けるとする。そして、この点については Oracle も争っていないことを指摘する。

(30) 実装コードを書くためには、証人が陪審に述べているように、コンピュータがどの程度速くタスクを実行できるか、メモリのサイズがどの程度なのかといった考慮すべき点をバランスさせることが必要となる。バッテリーで動作するデバイスのパワーマネジメントのような事柄についても気にしだすと、API 開発者による創作性はまるで「魔法」のようであるとある証人は述べている。デスクトップやラップトップの PC ではなく、スマートフォンという非常に異なる文脈で用いられる Android を開発するために必要とされる創作性とはまさにそのようなものといえる、としている。

プログラム開発者たちがシステムを学びやすく、開発がしやすく、他の言語を用いたくなくなるようにしようとしていた点⁽³¹⁾に注目している。

このような特徴を捉えて、ユーザインタフェースの一部としての宣言コードはプログラムの主要な動作とはかなり異なるものであって、他のプログラムのように機能的ではあるが、他方、その利用に際して、著作物性のないアイデアに本質的に制約を受け、その重要な価値は、著作権の生じない部分すなわち、プログラム開発者がAPIのシステムを学ぶ時間や労力を節減できるという点にあるとする。さらに、その価値とは、プログラム開発者がGoogleにより複製されていない実装コードを用いてシステムを学ぶこと、利用することを奨励しようとする役割にあるといえるとした。

著作権で保護されるもののうちには、著作権による保護の中核的な位置にあることを認識する必要があるものが存在するものの、宣言コードについては著作物性が肯定されるとしても、実装コードのようなプログラムに比べて、著作権による保護の中核からは離れているといえ、この点で、反対意見やCAFCの判断で提示された懸念⁽³²⁾は最小限にとどめられるとの見解を示した。これにより、「著作物の性質というファクターについてはfair useを肯定する方向へ向けられることになる」との判断を示した。

第1ファクター（著作物利用の目的と特徴）について。

fair useの判断に際しては、判例では、異なる目的や特徴を目指して当該著作物の表現を変えることによって、新たな表現、意味、メッセージを添加するような利用であるか否かを従前から考慮してきており、創作性を刺激するという著作権法の目的を実現するものであるのか、という問われ方がなされる場合もある。判例⁽³³⁾では、新しい重要な要素を添加する形での複製の利用を呼ぶ際に“transformative”（トランスフォーマティブ）という言葉を用いてきており、著作物利用がトランスフォーマティブなものか評価できるか否かという問題として扱われる。これが第一ファクターにおける評価の問題であることを確認している。

そして、「GoogleはJava APIの一部を正確に複製している、これはプログラム開発者が特定のタスクを実現する実装プログラムを呼び出せるようにするためである。とはいえ、著作権法による保護を受けるプログラムのあらゆる無許諾な利用とは実質は同様のことを行っていることになるし、このような行為をやめさせることは、プログラムの機能的な側面におけるfair useの範囲を著しく狭めることになるものといえる。むしろ、“transformative”であるかを判断する際には、複製行為についての「目的および特徴」についてより特定されたものについて検証しなくてはならない。」として、「目的および特徴」の詳細な特定を行い、「Googleによる利用とは、Androidをベースとしたスマートフォンの有用性を拡張することにある。プログラム開発者に対して、高度に創作的かつイノヴェーティブなツールを提供するものである。Googleがプログラム開発者に容易に使いやすいプラットフォームを作るべく、Java APIを利用している範囲では、著作権に係る合衆国憲法上の目的でもある、創作的進歩というものに一致しているものといえる。」との判断を示した。さらに、異なるプログラム間で互いにやり取りするには共有されたインタフェースが必要であること、プログラム開発者が既存のスキルを利用するにはインタフェースを再実装することが必要となること、APIの再利用は産業界においては一般的であること、Sun自体もJavaの開発に際して既に存在していたインタフェースを使用していたこと、Sunの首脳もJava言語がスマートフォンも含めた幅広い利用がなされることによって利益がもたらされると考えていたこと、等は陪審に説明されており、Googleを支持するamicusも同様の点を指摘しているという事実を摘示して、「これらの事実より、Googleによる複製の目的

(31) 元々Javaを開発してきたSunの戦略も、プログラム開発者にAPIを利用させることの重要性を強調しており、APIをオープンにして個々の事業者は実装の部分で競争することを求めていた、とする。また、トライアルにおける証言でも、利用者を中心に据えた宣言コードとイノヴェーティブな実装コードの間に明確な線を引く証人の例が数多くみられたことも指摘する。

(32) すなわち、fair useの適用によって著作権によるプログラムの保護が減じられてしまうという懸念を意味している。

(33) Campbell v. Acuff-Rose Music, Inc., 510 U. S. 569 (1994)

や特徴は、このファクターについても fair use を肯定する方向に重み付けをする程度に transformative なものであったと考えられる⁽³⁴⁾。」とする。

なお、さらに考慮すべき2点として、商業性 (commerciality) と good faith を挙げて補足している。商業性については、アメリカ著作権法 107 条では非商業的利用につき例示しているものの、商業的性質がないことは、fair use を肯定する方向へ傾かせることは疑いの余地はない反面、その逆は必ずしも成り立つとは言えない⁽³⁵⁾として、Google の利用が商業的性質のあるものである—この点はいずれの当事者も議論していないものの—としても、第1ファクターが肯定されないというものでなく、Android という新たなシステムにおいて再実装が果たしている、本質的に transformative な役割という観点からは肯定されるとの見解を示した⁽³⁶⁾。さらに、good faith については、Campbell 事件連邦最高裁判決では、fair use の分析において、何らかの役割を果たすものであるかという点については懐疑的な見解を示しているところであって、このような懐疑的な見方も正当性のあるものといえるとの見解を示す。すなわち、著作権は品行に対する特権のようなものでなく、一般的な問題として good faith が意味のある着眼点であるかについて論じるには及ばないとする⁽³⁷⁾。

第3ファクター (利用された部分の量と実質性) について。

fair use の成否に際して考慮される第3ファクターは、利用された著作物の量と利用の実質性に関するものである。

多数意見では、Google によって複製された宣言コードは、数百もの異なるタスクを呼び出すために必要な実質的にすべての宣言コードに相当するものであって、37 パッケージ (そこに含まれる 6000 以上のメソッド) における宣言コードは 11500 行と膨大なものといえる一方で、Java API の全体を考慮すると、Java API のコードの全体は実装コードも含めると 286 万行となって、そのうち 11500 行とはわずかに 0.4% に過ぎないことから複製された割合としてはわずかなものであるとして、著作物の量として問題となるのは、11500 行をそれ自体膨大なものとみるのか膨大な全体の一部としてみるのかという点であるとした。

そして、これまでの裁判例を踏まえて、「元の著作物の創作的表現のうち中心をなしているのであれば、少ない量の複製でも fair use の範囲から外れうるとしてきた。他方で、膨大な量の複製行為であっても、複製された部分が創作的表現をわずかにしか顕していないか、複製を行った者の有意なる目的にとって中心的なものである場合には、fair use の範囲に含まれるのである⁽³⁸⁾。」とした。

その上で、Google による複製行為の幾つかの特徴として、複製された分量をみるよりも、Google が複製しなかった数百万行を考慮することを示唆して、Java API で実装されているタスクに不可避的に制約を受けているということと、その目的は実装コードを呼び出すことにあって、Google は創作性や美のためにコードを複製しているのではない⁽³⁹⁾ことを指摘する。さらに、Google の基本的な目的とは、(Java とは) 異なっ

(34) “These and related facts convince us that the “purpose and character” of Google’s copying was transformative-to the point where this factor too weighs in favor of fair use.”

(35) fair use が問題となる多くの場合は明らかに商業的な利用であって、例えば、107 条の条文上の例として、ニュース報道といったものがあるが、これが商業的利益を目的になされることはよくあることである、としている。

(36) これは、商業的利用は、本件事案における利用の場合、第1ファクターの評価にとっては問題とならない、として評価しているようにも考えられる。

(37) このような見解とは、good faith は第1ファクターの評価には考慮しないということを実質的に意味する考え方を示しているようにも捉えられるところである。

(38) “We have said that even a small amount of copying may fall outside of the scope of fair use where the excerpt copied consists of the “heart” of the original work’s creative expression. Harper & Row, 471 U. S., at 564-565”

“On the other hand, copying a larger amount of material can fall within the scope of fair use where the material copied captures little of the material’s creative expression or is central to a copier’s valid purpose.”

(39) プログラム開発者が Java API を用いて作業をすることを既に学んでいることを前提とすると、これらのコードなしに、多くのプログラム開発者が Android 上のアプリケーションシステムを構築することを魅力的なものであるようにすることは極めて難しいものであったであろう。

たコンピュータ環境（スマートフォン）用の異なるタスク関連システムを作成して、その目的を実現し広めることを助けるプラットフォーム（Android プラットフォーム）を作ることであったことも摘示する。

これらより、「複製される量が有意かつ transformative な目的に向けられている場合においては、「実質性」のファクターは一般的にフェアユースを肯定する方向へウエイトを置かれることになる。」（下線部著者）とした⁽⁴⁰⁾。

そして、Google はわずか 170 行のコードのみを複製することで Java との互換性を実現しえた、とする CAFC の結論には賛成しないとして、このような結論は、Google の適法なる目的をあまりに狭く捉え過ぎているとする。すなわち、Google の基本的な目的とは、単に Java 言語を Android システム上で利用できるようにすることだけではなく、Android システムを備えたスマートフォン用の新しいプログラムを書く際に、プログラム開発者が Java API を用いる知識や経験を活用できるようにすることでもあって、さらには、宣言コードとは、プログラム開発者の創作エネルギーのロックを外すカギのようなものであり、イノヴェーティブな Android システムを作り、改良するためにはそのようなエネルギーを必要としていたという見解を示して⁽⁴¹⁾、結果的に、第 3 ファクターは fair use を肯定する方向へウエイトを置くものとした。

第 4 ファクター（市場への影響）について。

fair use の成否を検討する第 4 のファクターは、著作物の市場や価値における複製行為のもたらす影響に焦点をあてるものである。

コンピュータプログラムの場合、このファクターの考慮はより複雑なものとなる⁽⁴²⁾としつつ、Campbell 事件連邦最高裁判決⁽⁴³⁾でも指摘されたように、パロディーでも原作に対する需要を減じることもありうることから、潜在的な損失をすべて考慮するというものではなく、損失の量ではなく損失が生じる源を考えなくてはならないのであって、さらに、複製行為が生み出す公益を考慮に入れなくてはならないとする。

本件において考えられる損失について、陪審は、Android は Java SE についての現実のあるいは潜在的な市場に対して損害をもたらさなかったと判断しえたといえ、Google が API の一部を複製したか否かにかかわらず、Sun (Oracle) 自体がこれらの市場にうまく参入しえなかったものと判断しえたとする。その根拠として、第一に、トライアルにおける証拠によると、Android のスマートフォンの存否にかかわらず、Sun はモバイルフォン市場において優位な地位になかったことが明らかである⁽⁴⁴⁾こと、第二に、Google の Android プラットフォームを用いるデバイスは Sun の技術をライセンスしたデバイスとは異なるものであ

(40) このような見解は、第 3 ファクターにおける量や実質性についての評価内容は第 1 ファクター（目的や特徴）の正当性と連動して考えることを意味しているようにも考えられる。

(41) “Google’s basic objective was not simply to make the Java programming language usable on its Android systems. It was to permit programmers to make use of their knowledge and experience using the Sun Java API when they wrote new programs for smartphones with the Android platform.”

“In a sense, the declaring code was the key that it needed to unlock the programmers’ creative energies. And it needed those energies to create and to improve its own innovative Android systems. We consequently believe that this “substantiality” factor weighs in favor of fair use.”

このようなロジックは第 3 ファクターの検討を実質的に第 1 ファクターの議論とすり替えて、形を変えて第 1 ファクターにおける根拠をそのまま用いているに過ぎないようにも考えられる。

(42) 先例では、原作を完全にそのまま複製することは、原作に対して代替的な市場を作り出すことになるとして、原作の本に対する映画を作成することも同様であるといえる。これらの損失とは、創作性のある表現を保護する排他的権利を著作者に提供するという著作権法の目的と抵触することになるとする。

(43) Supra note 33

(44) Java SE の主な市場はラップトップ、デスクトップパソコンであるという十分な証拠が示されており、モバイルフォン市場へ参入しようとした Sun の多大な努力はうまくいかなかったことも陪審には示されている、との見解を示している。

また、Android の公開以前の 2006 年までに遡ると、Sun の役員は、スマートフォンの急展開によって、モバイルフォン部門の収益が減少していることを提示しており、Sun の前 CEO は、Sun がスマートフォンを立ち上げることに失敗したのは、Google の Android の発展によるものであるのかどうか、直接にたずねられた際に、そうではないと答えていることを指摘する。このようにモバイルフォン製品を開発するビジネス上のチャレンジについて Sun は道をふさがれていた状況にあったことを示す証拠を提示されることで、陪審は以上のような評価をすることに納得がいったといえる、との見解を示している。

ることを、陪審では繰り返し説明されている⁽⁴⁵⁾こと、さらに、Androidのような新しいプラットフォームにおいてJava言語が幅広く利用され、Javaの訓練を受けたプログラム開発者のネットワークが広がることから、Sunは利益を得ることも予見されたという証拠も陪審に対しては示されており、いいかえれば、陪審は、AndroidとJava SEという2つの異なる市場におけるものとして理解しえた⁽⁴⁶⁾ことを挙げて、いずれにしても、陪審のfair useに係る判断に際して、少なくともGoogleがJava APIの一部を利用するか否かということが、Sunのスマートフォン市場参入が困難であったことを示す証拠に対して何ら影響することはなかったことを意味しているとする。

他方、Googleによる複製は同社がAndroidプラットフォームから莫大な利益を得る助けとなったことから、Oracleは、Java APIについての著作権のエンフォースメントをもって、その利益の相当の分配を求めているともいえるところ、Oracleに何故このような利益が分け与えられるべきことになるのかという理由を考えることが重要であるとする。そして、APIや表計算プログラムのような新しいインタフェースを最初に市場に投入すると、その表現上の性質によって、新たなユーザを引き付けることになりうるが、時間経過に伴って、プログラム開発者を含めたユーザがそれに慣れてくるということで異なる価値をもつことを指摘して、Androidの収益は、第三者すなわちプログラム開発者のJavaプログラムにおける投資に大きく関係しているものの、一方で、Java APIを作るSunの投資とはほとんど関係がないとする。これについて、「我々は、著作権法が、創作物を動作させる方法を学ぶに際しての第三者の投資を保護するものと信じることはできない⁽⁴⁷⁾。」との見解を示す。

さらに、Java APIを学ぶためにプログラム開発者による投資がなされることを前提として、Oracleによる著作権行使を許容することは、公共に対して害悪をもたらすリスクがあるとする。同様な内容をもった別のAPIを新たに作る手間やコストを生じさせることになれば、著作権のエンフォースメントを肯定することが、Java APIの宣言コードを活用して新しいプログラムについての創作を将来的に行うことを制限する足かせとなって、Oracleだけがその足かせを外すカギをもつこととなり、Oracle（あるいはインタフェースの著作権を有する事業者）にとって極めて大きな利益をもたらすことになるが、そのような利益とは、インタフェースを使うことを学んできたユーザによって開発された創作的な改良や新しいアプリケーションや新たな用途から生まれるものであって、その限りでは、このような足かせは著作権の基礎となる創作性に係る趣旨を害するものとする⁽⁴⁸⁾。そして、「著作権とはアイデアを作り出して、世に広めるという両方についての経済的インセンティブを供給している」⁽⁴⁹⁾ことから、ユーザインタフェースの再実装化を認めること

(45) 例えば、スマートフォンとフィーチャーフォンを区別してきたことを証人が説明していること、Sunが製作したソフトウェアを利用した特定のデバイスについては、タッチスクリーンがなく、別のものはキーボードがないといった説明がなされていたこと、Amazon キンドルのように、よりシンプルな製品が提示され、より進んだキンドルファイヤのような装置ではAndroidを用いていたこと、といった証拠によると、Sunの製作したコードを大型のコンピュータから小型のものへと再目的化するよりも、GoogleのAndroidプラットフォームは、むしろ、より進んだ別の市場に属するものの一環であったことを示しているという結論を導いている。また、Androidのプラットフォームは、アプリケーションプログラミングの枠組みであるJava SEとは非常に異なるタイプの製品であるといえること、証拠からは、Sunのモバイルフォンのビジネスは衰退していて、他方で、市場はSunが決して導入しえなかったような新しい形のスマートフォン技術への需要が増大していたといえることを指摘している。

(46) 問題となっている2つの市場が存在することから、一方の市場（スマートフォン）で仕事をするためにJava言語を学んでいるプログラム開発者は、他の市場（ラップトップPC）においても自らの才能を生かすことができる、との見解を示した。

CAFCは「市場への影響」というファクターについてはfair useの成立を阻害するものとして現に判断しており、これは、SunがAndroid市場に参入しようと試みていたからである（SunはGoogleとライセンス交渉を進めていたとする）とするOracleの主張に対して、このようなライセンス交渉は宣言コードの37パッケージ以上のものに関わるものであり、両企業間における「Javaの実装」とか「ブランディングや協調」といった事柄に関わるものであるとした。

(47) “We have no reason to believe that the Copyright Act seeks to protect third parties’ investment in learning how to operate a created work.”

(48) この点は、Java APIについては既に世の中で広く用いられ利用されていることから、それ以上のインセンティブ付与のための法的保護は不要ということの意味しているように解される。

は創作的な新しいプログラム（コンピュータコード）がより容易に市場へ参入していくことを可能にするとの見解を示す。

以上のような検討から、Sun が Android 市場で競争する能力が不確実であったこと、損なわれた収益の源、創作性に関わる面で公共に対してもたらず害悪、これらのものを併せて考えると、「市場への影響」という第4ファクターについても、fair use を肯定する方向へウエイトを置くことを確信させるものといえるとの見解を示した。

4つのファクターについての以上の検討を踏まえて、「本件では、ユーザが、新しいトランスフォーマティブなプログラムに取り組むべく、その才能をいかせるように必要とされていたものを得るためだけに、Google はユーザインタフェースを再実装しており、Google による複製は法律問題として fair use に当たる。CAFC の判断を覆した上で、本結論と適合すべく、さらなる手続のために差し戻す⁽⁵⁰⁾。」との結論を導いている。

なお、コンピュータプログラムは機能的なものであって伝統的な著作権の概念の適用が難しいといえるが、適用に際して、これらの概念の性質を何ら変えているものでもなく、fair use に係る初期の裁判例を覆したり、修正したりしているものではないとしている⁽⁵¹⁾。

3-2 反対意見の概要

一方、Thomas 判事による反対意見の構造としては、まず、多数意見が Java API の宣言コードの著作物性について判断することなく、fair use により非侵害との判断に至った点を批判しており、その上で、Java API の著作物性があることを前提に、fair use による著作権侵害を否定するとした結論自体も誤っているとの見解を示している。

宣言コードの著作物性については、著作権法の文言(101条)より肯定されることは明らかとの立場を示す。すなわち、「宣言コード→機能的な表現→操作方法(102条(b))に該当→著作物性なし」というロジックは間違いであって、機能的という意味では宣言コードも実装コードも変わらないし、宣言コードを用いるというアイデアは保護されないが、そのアイデアの特定の表現は著作物となりうるのにもかかわらず、多数意見は、立法の趣旨に反して、宣言コードと実装コードの区別を主たる根拠としている⁽⁵²⁾と批判する。

次に、fair use の判断における各ファクターの適否については、以下のような見解を示す。まず、第2ファクターについて、多数意見では宣言コードと実装コードの区別を行うために用いており、宣言コードはアイデアと本来的に結びついていることから著作権法による保護の中心から離れているという誤った裏付けをしているとする。第4ファクターについては、本件では、結論としては潜在的な市場への影響はあるとの見解を示す。すなわち、一つは、Android プラットフォームが無料で利用できることにより Java API を利用することに伴う料金をユーザは支払わなくなったという点であり、もう一つは、スマホ市場における Java API プラットフォームへのライセンス機会が阻害されたこと、つまり多数意見では Java API が Android で用いられるに際して Oracle はライセンスもなしえたことが検討されていない、ということ指摘する。また、多数意見は Oracle による権利行使による影響を懸念しているが、本件事案による著作権行使が問題となる

(49) Harper & Row v. Nation Enterprises, 471 U. S. 539 at 558

(50) “We reach the conclusion that in this case, where Google reimplemented a user interface, taking only what was needed to allow users to put their accrued talents to work in a new and transformative program, Google’s copying of the Sun Java API was a fair use of that material as a matter of law. The Federal Circuit’s contrary judgment is reversed, and the case is remanded for further proceedings in conformity with this.”

(51) むしろ、fair use のような法理の適用については、立法府と裁判所の協調的努力によるものであることが長らく明らかにされてきたことを認識しており、そこで、107条で規定され、初期の裁判例で形成された原則に立ち返って、異なる種類の著作物に対しても適用してきたのである、とする。

(52) The majority holds otherwise-concluding that every factor favors Google-by relying, in large part, on a distinction it draws between declaring and implementing code, a distinction that the statute rejects.

Androidとは2014年11月までにリリースされたバージョンに限られるのであって、Oracleによる著作権行使がAndroidのその後のバージョンにまで影響するものではないとする。第1ファクターについては、多数意見では商業的利用をもって fair use を否定する根拠とならないとするが、権利者の膨大なる利益を破壊するような複製行為について fair use は認められないのであって、Googleの利用内容とは、新しい何かを付け加える目的というような transformative なものではなく、同一の目的で利用しているものであることから、多数意見は、transformative の理解を誤っている⁽⁵³⁾と批判する。第3ファクターについては、Googleは宣言コードというプログラム開発者を引き付ける要素をそのまま複製しているのであって、fair use は否定される方向に作用するとする。多数意見は必要最小限の複製であると評価するが、その利用内容自体が transformative なものではないのであるから、この分析は誤っているとする。仮に transformative なものであるとしても、その量は相当のものであって、質・量とも実質的であるといえるとする。以上のような、各ファクターについての検討から、結論として、宣言コードの著作物性についていずれにしても判断を示す必要があるのであって、多数意見は、誤った fair use 法理の分析を行ったことによって、議会の政策判断を骨抜きにしているという批判をしている。

4 連邦最高裁判決の評価と課題

4-1 多数意見の検討

連邦最高裁判決の多数意見について、これまでの内容の検討分析を基に、幾つかの観点から検討を行う。

まず、本件判決は事例判決と位置付けられるものであるのか、という問題が挙げられる。この点、多数意見では、その冒頭で、「当事者の議論を解決するに際して必要となる以上の回答を行うべきではない」として、本件判決における判断があくまで事例判断であるかのような姿勢を示していることから、本件判決で提示された論理を先例として幅広く適用することの合理性はあまり見いだせないように考えられる。

また、多数意見では、原々審判決である連邦地裁判決および原審である CAFC 判決でも明確に判断が示された API の著作物性についての判断を回避している点が注目される。すなわち、著作物性については、「純粋に議論の目的のため」という前置きの下で肯定されるとして、fair use の適否の検討へ進んでいる。したがって、多数意見では本件事案で問題となった Java API の著作物性自体について肯定する判断を示したものと捉えることは当然できないし、これに限られることなく、一般論として API の著作物性についても何ら判断されなかったものとして理解せざるをえないと考えられる。そして、この点については、学説における本件判決に対する批判としてもみられるところである。Google を支持する amicus としては合計で 27 件提出されたところであるが、そのうち 24 件は Java API の著作物性が否定されることを根拠として示されているものであって、fair use を根拠とするものはわずか 2 件だったことから、API の著作物性について、連邦最高裁判決で何らかの判断基準が提示されることは強く期待されていたことがうかがえる。

次に、fair use 法理の条文上規定された各ファクターの当てはめにおける特徴について検討する。fair use 法理の適用に際して、107 条で列挙されている 4 つのファクターの適用については、多数意見では、これらが網羅的なものでないことは認識しつつも、4 つのファクターすべてについての評価を行っている。その順序については、第 2 → 第 1 → 第 3 → 第 4 という順序であって、Thomas 判事が反対意見で批判するようにその順序自体が入れ替わっていること自体が大きな意味を有すると評価できるか否かについては必ずしも断定できないとしても、検討内容自体が、これまでの fair use 法理に係る事例における取扱いと比べて、かなり特異的なものであることは明らかであると考えられる。本件判決に対する見解としても、第 2 ファクターにつ

(53) "Ultimately, the majority wrongly conflates transformative use with derivative use. To be transformative, a work must do something fundamentally different from the original. A work that simply serves the same purpose in a new context—which the majority concedes is true here—is derivative, not transformative."

いて積極的に判断を示して fair use を肯定したという点が非常に特徴的であると指摘するものがみられる⁽⁵⁴⁾。すなわち、fair use について 107 条で列挙されている 4 つのファクターのうち、第 2 ファクターは従来からは重要性の低いファクターであるとして判例・学説で認識されていたところ、本件判決では第 2 ファクターの重要性を明らかにしたところに大きな意義があるとされる⁽⁵⁵⁾。また、従前の fair use 法理の適用に係る裁判例（1978—2019 年までの裁判例）における判断事例における各ファクターの分析についての網羅的な実証研究⁽⁵⁶⁾によると、第 4 ファクターが fair use の成否判断において圧倒的に大きな影響を及ぼしており、第 2 ファクターについては、fair use の成否判断に対するインパクトとしてはもっとも少ないものであることが明らかにされていることから、本件判決の特徴として、確かにこの点を挙げることは肯定できるであろう。

次に、各ファクターにおける評価内容について検討する。

第 2 ファクター（著作物の性質）について、本件事案で著作権侵害が問題となっている対象とは Java API を構成する宣言コードの部分であって、ハードウェアを実際に動作させるプログラム部分（実装コード）とは明確に異なることを認識しており、宣言コードの役割機能上、その表現形式としては創作性が高いものではないことを確認した上で、加えて、宣言コードの価値は表現自体ではなく、プログラム開発者が API のシステムを学ぶ時間や労力を節減できるという（アイデア自体に）価値があることを肯定して、実装コードのようなプログラムに比べて、著作権による保護の中心となるものより周辺的なものであることをもって、fair use 成立の肯定へ傾くというロジックを採っている。この点、API を構成する宣言コードについては、ハードウェアに対する指令を構成しているコード部分（実装コード）と比べて、顕著に異なるという指摘は的確なものであると考えられ、またそのようなコードとは、要は特定の言語ベースでの開発スキルの有効活用といったところに価値が見いだされるのであって、表現としての保護という観点からは、そのような性質を考慮することの意義はあるといえるであろう。そして、これらの性質を踏まえると、著作物の性質という第 2 ファクターにおいては fair use 肯定の方向に作用するという評価を行ったことについては妥当性があるものと考えられる。

第 1 ファクター（利用の目的と特徴）については、先例と同様に、Google による利用は transformative な利用であるか否かという基準をもって判断を行っているといえる。Google はプログラム開発者に容易に使いやすい Android プラットフォームを作るべく、Java API を利用していることから創作的進歩というものに一致するとして、さらに、宣言コードに対応する実装コードは自ら再実装していること、API の再利用自体は産業界においては一般的であること、Sun も Java 開発で他のインタフェースを再利用していたこと、Sun は Java 言語が幅広く利用されることを望ましいものと考えていたこと、等を根拠として、fair use を肯定できる程度に transformative なものであるとするが、これについては、以下のような観点から、若干の疑問が生じる。

なるほど、Google が Android に Java API の宣言コードを取り入れることによって、既存の Java によるアプリ開発に慣れたプログラム開発者にとってみれば、Android プラットフォーム上でのアプリ開発に際してのアクセシビリティが向上したという点は、確かに創作的進歩と評価できる余地はあるといえるのかもしれない。しかしながら、それはあくまでも Android プラットフォームの構造についての設計という面でユーザへの「バリアフリー」な進歩（工夫？）を取り入れたということに過ぎないと捉えることもまた可能なのであって、Google による Java API の宣言コードの導入とは、表現としての側面における transformative な利用と評価するよりも、実質的にはむしろアイデアとしての側面における transformative な利用を意味

(54) Michaels, Andrew C., *Functionality's Role In Oracle Copyright Ruling Isn't So Novel* (May 7, 2021). Law 360, 2021. Available at SSRN: <https://ssrn.com/abstract=3842052>

(55) Michaels, Andrew C., *supra* note 54

(56) Barton Beebe, *An Empirical Study of U. S. Copyright Fair Use Opinions Updated, 1978-2019*, 10 N. Y. U. Journal Of Intell. Prop. & Ent. Law1 (2020)

しているものと理解されるべきではないかと考えられる。すなわち、Android上で動作するアプリ開発をプログラマーが行うに際してJava APIにおける宣言コードの表現をそのまま利用可能とすることは、Androidプラットフォームが元の宣言コードの表現自体に対して、表現として何らかの新しいものを付け加えたのかという評価に対して積極的な意味を有するものと捉えることとは別の意味であると考えられ、そのような視点からの評価検討については多数意見では全くなされていないものと考えられる。

結局のところ、第1ファクターの検討の結論とは、プログラム開発におけるインタフェースの再利用はプログラム開発の当事者に広く受容されており、本件の著作権者であるOracle側もそのことについては認識していたということをもって、fair use成立の肯定に傾けられる評価をしたことを意味しているに他ならないものと考えられる。

第3ファクター（利用された部分の量と実質性）の評価に際しては、Java APIは実装コードを呼び出すという目的に制約を受けており、Java APIを既に学んでいる多くのプログラム開発者がAndroidのスマートフォンシステムを構築することを惹きつけるようにすることは極めて難しいなかで、複製されている部分は異なるコンピュータ環境（スマートフォン）用のプラットフォーム（Androidプラットフォーム）を構築するという意味のあるtransformativeな目的に向けられていることを根拠に、第3ファクターについてfair use肯定へ傾く、とする判断を導いている。ところで、多数意見における第3ファクターの評価においては、利用された部分の実質性の検討について、第1ファクターの検討、とりわけtransformativeに係る評価にほとんど依存してなされており、端的にみれば、第3ファクターに対する実体的な評価はなされているとはほとんどいいがたく、事実上は第1ファクターの議論にとって代わられているものと考えられる。また、利用された部分の具体的な量の評価に至っては事実上全くなされていないといえる。したがって、これらの検討を基にして導出された第3ファクターについてもfair use肯定とする結論の合理性については、かなり疑問の余地があるものと考えられる⁽⁵⁷⁾。

宣言コードについて著作物であることを肯定することを前提とすれば、量や実質性の評価とは、あくまでもAPIのうち宣言コード部分について、全体に対してどの程度複製されたのか、複製された宣言コードがどのような利用がなされているのか、という評価を中心に行えば足りるものと考えられるのであって、そうすると、Java APIを構成している宣言コードの相当部分をGoogleが複製・利用している事実自体については否定しえないようにも考えられる。すなわち、APIのうち宣言コード部分という「著作物」だけに限ってみたら、宣言コードという「表現」自体は、まさに実装コードを呼び出す目的のためにそのままの形で複製され、利用されているといえるであろう。ここで、API全体のコード行からすれば、複製された部分は全体のわずか0.4%に過ぎないということは問題とならないであろう。なぜなら、本件ではGoogleによって独自に作成された実装コード部分についても広義にはAPIに包含されるものである以上、量的にはこれがAPI全体の大きな部分を占めることになるのは当然であって、実装コードを呼び出す機能に過ぎない宣言コードの行数の割合が全体比としてみれば少なくなることは至極当然といえる。すなわち、第3ファクターの観点からみれば、Java APIを構成する166パッケージのうち37パッケージ、加えてそれらがJava APIにおいて非常に基本的な部分を占めているということを踏まえる限り、実質的に宣言コードの表現を利用するべく相当の量について複製・利用はなされているものと評価せざるをえないものと考えられる。そうすると、少なくとも第3ファクターに関してみると、fair use成立の肯定には傾きにくいものと考えられる。

第4ファクター（市場への影響）については、概ね以下の3点、第一にAndroidはJava SEについての現実のあるいは潜在的な市場に対して損害をもたらすことなく、Googleによる複製・利用行為にかかわらず、SunはAndroidの市場にうまく参入しえなかったことからAndroidとJava SEはそもそも市場が異なっているということ、第二にJava APIの著作権のエンフォースメントによりもたらされる利益とは、プログラ

(57) 端的には、第3ファクターについての検討・評価は実質的になされていないという印象を持たざるをえない。

ム開発者を含めたユーザがAPIに慣れてくるということによって生じるものであって、著作権法はこのような利益を保護するものではないこと、第三にJava APIの著作権のエンフォースメントを許すことは、Java APIの宣言コードを活用することで将来へ向けた新しいプログラムの創作を制限することになって公益を害するものであること、を理由として、fair use肯定へ傾くという論理をとっている。

確かに、SunがスマートフォンOS市場に参入する可能性自体はそもそも高いものではなかったとはいえるかもしれない。ただし、Android上のアプリ開発については、本件で問題となったJava APIの利用によって、事実上Java言語をベースとして行うことが可能になったことで、アプリ開発を行うユーザ側としては(Sunが提供してきた)Java SEを用いることを完全にバイパスすることが可能となったものと捉えることはできるであろう。すなわち、Android上のアプリ開発を行うユーザとしては、SunによるJava標準仕様認証といったプロセスに捉われることなく、Android上で動作するアプリ開発を自由にできることになったともいえるのであって、Java SEがそもそもアプリケーションプログラムの開発ツールとしての製品であったことを考えると、その製品の「市場」については、Googleの行為によって少なからず影響がもたらされているものと評価する余地があるものと考えられる。加えて、OracleはJava MEというモバイル版に特化したJava SEの亜流バージョンを提供していることもあるので、モバイル機器上で動作するアプリケーション開発ツールとしての製品市場において、Android上のアプリ開発と一部重なっているともいえるであろう。このようなことから、Googleの行為によってOracleにとっての著作物市場への影響はないとは断じることが妥当な評価とはいえないものと考えられる。

また、多数意見が指摘するような、Oracleによる著作権のエンフォースメントを認めることが、ユーザが既に学習したり、「慣れている」ことから派生的に生じる利益を著作権者に対して確保することを直ちに意味するともいえないと考えられる。すなわち、宣言コードが著作物であることを前提とする以上、当該著作物に係る表現を複製利用することに伴う対価を得る機会が原則として保護されることに立ち返って考えるのであれば、Googleによる「Java SEからユーザがバイパスする機会の提供」を生じさせるような利用行為によって、Java SEに係る市場に対しても何らかの影響がもたらされたものとして評価しうる事柄と考えられる。

なるほど、多数意見が指摘するように、OracleがJava APIの宣言コードについての著作権を行使することによってAndroidプラットフォームにおけるJava APIの利用についてユーザが制約を受けることとなつて、このことがAndroid上のアプリ開発という局面での新たな創作に対して制限を課す可能性があることも否定できないであろう。そして、第4ファクターの評価において公益性の観点を取り入れるのであれば、この点は否定できないものと考えられる。

以上の検討から、第4ファクターについては、公益性の観点からはfair use肯定を根拠付けることができるとしても、むしろGoogleによる複製・利用行為が著作権者の著作物市場へもたらさう影響については無視しえないようにも考えられ、第4ファクターの評価の結論についても、結局のところ、宣言コードの性質上の特徴付けが色濃く影を落としているように考えられる。

このように、第1から第4ファクターについての多数意見の判断内容について検討を行った結果、fair useの成否に係る多数意見によるロジックとは、実質的には第2ファクターを中心として構成されているものであるといえるのであって、他のファクターにおける適合性については、いずれも説得力は乏しいものとする。端的にいえば、第2ファクターの充足性だけを実質的な決め手としてfair useの成立を肯定しているものであるように考えられる⁽⁵⁸⁾。

Googleによる著作権侵害を否定した本件判決の最終的な結論自体については、大きく妥当性を失するものではないと評価できるかもしれないが、Java APIについての著作物性を肯定することを前提とした上でfair use法理の適用及び肯定によって著作権侵害を否定するものとした多数意見のロジックは、これまでの判例におけるfair use法理の適用と比べるとかなり異なる⁽⁵⁹⁾ものであって、疑問の余地が生じる場所も少

なくないと解せざるを得ないであろう。

もっとも、逆にみれば、このように個々のファクターについて積極的に肯定しえなくても、特定のファクターにおける充足性にウエイトを置いて評価を行って、総論としては fair use の成立を肯定するものとして、柔軟な評価によるエンフォースメント制限を可能にする点が fair use 法理の最大のメリットであることが、本件判決によって改めて確認されたともいえるのかもしれない。

また、本件事案における Java API 固有の背景事情に着目すると、先に検討したように、元々は Sun 自らが Java API の標準化作業の中心的な役割を果たしながらも、個別の仕様についてはコミュニティの参画を通じて形成されてきたという経緯もあり、さらにその後は Oracle 自体もオープンソースライセンスの下で Java API の提供を始めたという経緯もみられることから、このような一連の形成過程における「特殊性」についても留意すべきであろう。すなわち、Java API 自体の利用が社会に対して広く開放された状況の下で発展・形成されてきたという点をもってすれば、API 自体についての著作物性が肯定されるという仮定の下でも、fair use 法理を適用して結論として Java API を構成する宣言コードについての著作権行使を否定するという結論自体の違和感は緩和されてくるようにも思われる。しかしながら、Java API 固有の利用と発展の形成過程と異なって、仮に一つの企業によって完全に独立・完結した形で開発されて proprietary に提供されているプログラムを想定した場合にも、そのプログラムの著作物性を肯定しつつも、多数意見のようなロジックによって fair use 法理の適用を肯定しうるのかという形で捉えなおすとすれば、その妥当性が同様に担保されえるのかということには、やはり大いに疑問が残るものとも考える。より一般的に適応可能な論理構成を考えるのであれば、むしろ API 自体の著作物性を一律に否定するか、一定条件を充たす API における著作権に限って権利行使を定型的に否定するという論理構成をもって結論を導出する方が普遍的かつ説得的なものであったといえるのかもしれない。

4-2 学説等における評価

学説における本件判決の評価については、未だ十分に多くのものが公表されている状況ではないと考えられるものの、結論として著作権行使が否定されたことについては、肯定的な評価を示しているもの⁽⁶⁰⁾や少なくとも否定的ではない見解を示しているものが多い⁽⁶¹⁾ことは注目される。さらに、本件において連邦最高裁判決のレベルで fair use 法理の適用による著作権侵害を否定したことによって、下級審も積極的に fair use 法理の適用を行い、情報技術分野のイノベーションが促進される可能性を指摘するもの⁽⁶²⁾などもみら

(58) この点は、上記の学説 (Michaels, Andrew C., supra note 54) も第2ファクターにおける著作物の性質付けが fair use の分析全体を強く肯定寄りに傾けているという見解を示している。なお、本件事案における各ファクターについて私見も含めてまとめると以下になるだろう。

	私見	多数意見	反対意見
第2	○	○	×
第1	△	○	×
第3	△ or ×	○	×
第4	△	○	×

(○：充足性判断は合理的 △：充足性判断にやや疑問 ×：充足性判断に疑問)

(59) 以下の4-2でも検討するように、本件判決の論理構成については、従来の fair use 法理の適用事例と比べてかなり異なるものであることは学説によっても認識されているものと考えられる。

(60) Menell, Peter S., Google v. Oracle and the Grateful (API) Dead: What a Long Strange Trip Its Been (April 12, 2021). DAILY JOURNAL (Forthcoming), Available at SSRN: <https://ssrn.com/abstract=3824442> or <http://dx.doi.org/10.2139/ssrn.3824442>

(61) 例えば、後に提示する、本件判決における fair use 法理の適用手法については批判的な見解を提示している学説 (infra note 69, 70, 71) においても、本件判決の結論自体の妥当性については必ずしも批判していないように捉えられる。

(62) Tran, Jasper and Kido, Kristen, Google v. Oracle: Copying Declaring Code Is Fair Use (April 17, 2021). George Washington Law Review On the Docket, Vol. 89, 2021, Available at SSRN: <https://ssrn.com/abstract=3828651>; Copyright Act of 1976-Intellectual Property-Fair Use-Google LLC v. Oracle America, Inc., 135 Harv. L. Rev. 431 (2022)

れる。

また、法的な見解のみならず、Java 言語の開発者である James Gosling 自身も連邦最高裁判決の結論は正しいものであって、もし Oracle が勝訴していたら却って彼らは損害を被っていたであろうという見解を示したとされている⁽⁶³⁾。その理由としては、仮初にも Oracle が勝訴したとすれば、その根拠をもって、Oracle が自らのビジネスの要としているクラウドで用いている API についても、逆に Oracle は他社から権利行使を一齐に受けかねないからというものである。

他方、本件事案は API の著作物性そのものを否定することで結論が導かれるべきであったとする見解が学説では当初から広く展開されている点も注目される⁽⁶⁴⁾。そもそも、コンピュータプログラムにおける互換性 (compatibility/interoperability) 確保のために、著作権侵害の否定やプログラムにおける一定部分について著作物として保護することを否定する考え方は判例上も以前から確立されていた⁽⁶⁵⁾といえる。その論拠として、scènes à faire 法理、merger 法理、著作権法 102 条 (b) といったものが挙げられており、本件事案において、Google も Java API の宣言コードは 102 条 (b) の「操作方法」に該当するという主張を積極的に行ってきた。これまでの学説でも、interoperability 確保のため API の著作物性についてはア priori に否定されることが妥当であることを提示する説はかなり有力といえるのであって、このような立場からは、本件判決の結論は妥当とするものの、著作物性についての明確な判断を示さなかった点について批判⁽⁶⁶⁾が生じるであろう。

もっとも、本件事案における Google による Java API の一部の複製利用行為とは、従前の判例で問題となったような純粋な意味での互換性確保やリヴァースエンジニアリングといったものとは、かなり性質が異なるものであるようにも考えられ、このような従前の議論の延長線上に本件事案の評価をすることについては若干慎重であるべきものとする。すなわち、Java API とは、あくまでもアプリ開発等のためのプログラミングを支援するツールといえるのであって、片や Android はスマホ用 OS を核としたプラットフォームで、同 OS 上で動作するアプリ開発のためのツールとしての観点から、Java API との間における、いわば「バリアフリー化」を図ることでアプリ開発者の層を拡大することを大きな目的として、Java API の宣言コードが Android に取り込まれたといえる。互換性という概念について、それが実現されることによって、相互にメリットを享受しうるものであると観念するのであれば、Google にとってみれば Java API の宣言コードを Android に取り込むことには明らかに多大なメリットを発生させることが予想できたといえるものの、反対に、Java API を基にした Java SE 等の認証をビジネスとしている Oracle からみて、Android 上のアプリ開発に Java API の宣言コードが活用されることが直ちに大きなメリットをもたらしたといえるのかという点では大いに疑問も残るところである。Java 言語をプログラミングに用いるユーザは元々相当程度存在

(63) Daryll K. Taft, Supreme Court ruling on Java APIs eases developer worries (theserverside.com) April 06, 2021

(64) 例えば、Peter S. Menell, "RISE OF THE API COPYRIGHT DEAD?: AN UPDATED EPITAPH FOR COPYRIGHT PROTECTION OF NETWORK AND FUNCTIONAL FEATURES OF COMPUTER SOFTWARE", 31 Harv. J. L. & Tech 305 (2018), Pamela Samuelson & Catherine Crump, WHY 72 INTELLECTUAL PROPERTY SCHOLARS SUPPORTED GOOGLE'S COPYRIGHTABILITY ANALYSIS IN THE ORACLE CASE, 36 Berkeley Tech L. J. 413 (2022)

(65) ソフトウェアの互換性と著作権の関係性についての判断を示した主な事例として、Computer Associates Int'l, Inc. v. Altai, Inc. 982 F. 2d 693 (2d Cir. 1992) (いわゆる "abstraction-filtration-comparison" (AFC) test を提示して、異なるソフトウェア間における compatibility に係る要素における類似性がある場合でも、著作権侵害を否定)、Sega Enterprises Ltd. v. Accolade, Inc., 977 F. 2d 1510 (9th Cir. 1992) (ソフトウェアのリヴァースエンジニアリングに関連して、ソフトウェアの構成要素のうち一定要素について、著作権法 102 条 (b) により著作物性を否定し、全体として fair use 法理の適用により権利行使も否定。)、Sony Computer Entm't, Inc. v. Connectix Corp., 203 F. 3d 596, 606 (9th Cir. 2000) (テスト目的での複製について、fair use が成立することを肯定。)、Lotus Development Corp. v. Borland International, Inc., 49 F. 3d 807 (1st Cir. 1995) (表計算ソフト階層構造について、著作権法 102 条 (b) により著作物性を否定。)

(66) Lemley, Mark A. and Samuelson, Pamela, Interfaces and Interoperability After Google v. Oracle (August 2, 2021). Available at SSRN: <https://ssrn.com/abstract=3898154>

していたことから、Android 上のアプリ開発に Java 言語を利用することが可能となること自体が Oracle にとって直ちにそのビジネスモデルの上でメリットが生じるとは簡単には評しえないからである⁽⁶⁷⁾。

したがって、従来議論されてきたコンピュータプログラムにおける互換性 (compatibility/ interoperability) の問題だけを根拠として、その著作物性をアプリオリに否定する考え方を、本件事案における宣言コードを巡る事案にそのまま適用することがもっとも合理性の高い解決策であるとは言えないように考えられる。その意味では Thomas 判事の反対意見にも相応の合理性は見いだせるものと思われる。また、宣言コードを「操作方法」と同様に単なる技術的アイデアと同質のものとして捉えることによって、解釈論上も類型的に著作物性を否定するという考え方についても、理論的には否定することはできないとしても、パッケージ、クラスとしての分類も含めた宣言コード自体の名称記述については本来的には無限に存在する多種多様な記述を選択肢となしうるところ、創作性が高いとはいいがたいもののアプリオリに著作物性がないものと評価することにもやや疑念が残るところである。

その他、学説における興味深い指摘としては、本件判決が fair use 法理の適用によって Oracle の著作権行使を否定したことについて、fair use 法理の遠大なる拡大 (a far-reaching expansion of the fair use doctrine) として、fair use 法理適用の分析において著作物性について検討されるべき事柄を取り入れているという点で、プログラムにおける fair use 法理の新しい地平を拓いたものと位置付けて、とりわけもっとも遠大なる点とは、その結論のために fair use に依存したという点であるとする見解が挙げられる⁽⁶⁸⁾。この見解によると、著作権法 102 条 (b) により宣言コードの著作物性を否定する論理構成を採った場合、プログラムのうち著作物性を否定されるものとそうでないものを分けることになり、結果的にあらゆるコンピュータプログラムについても著作物性を否定する結果に至るのではないかという懸念が oral argument を通じて連邦最高裁判事間である程度醸成されており、そのような条文解釈による著作物性を示すことに対する懸念と著作権行使に否定的な IT 業界の意見との間を「調和」させるべく、fair use 法理の中に著作物性を否定するような論理が取り込まれたものが連邦最高裁判決の多数意見であった、とする分析が示されている。

また、本件判決の fair use 法理の適用手法や内容に対して、批判的な見解も少なくない。例えば、本件判決は、商業的な目的で著作物を無許諾で複製した営利企業の法的責任について fair use 法理を適用し否定した初めての事例として異例な事例と位置付ける学説⁽⁶⁹⁾、本件判決の fair use 法理の適用手法は予見可能性を今まで以上に損なわせるもので、とりわけ、第 1 ファクターの good faith の理解と第 4 ファクターの評価について批判的な見解を提示する学説⁽⁷⁰⁾、そもそも第 4 ファクターは著作物の市場における影響に係る分析を前提とするものであるにもかかわらず、本件判決では公益 (public benefit) への影響を行っていることを指摘して、他方でそもそも fair use 法理自体が公益性と著作権者の利益の調整を図る概念であることから、理論的には第 5 ファクターとして公益 (public benefit) への影響を考慮する新たな fair use 法理のフレーム

(67) もっとも、Lemley, Mark A. and Samuelson, supra note 66 は、そもそも interoperability を巡っては、一方向 (oneway) 的なものとなる可能性があることは当然に想定するという理解をしているようである。

(68) “On its own, the majority opinion broke new ground on fair use and its application to computer software, by effectively incorporating considerations normally associated with copyrightability into the fair use analysis. Yet what is even more far-reaching, even if entirely hidden, was the reason for the majority’s reliance on fair use for its result.” Balganes, Shyamkrishna, *The Institutional Turn in Supreme Court Copyright Jurisprudence* (May 5, 2021). 2021 Supreme Court Review (forthcoming), Available at SSRN: <https://ssrn.com/abstract=3842276>

(69) Mossoff, Adam, *Declaring Computer Code Uncopyrightable with a Creative Fair Use Analysis* (September 8, 2021). *Cato Supreme Court Review*, vol. 20 (2020-2021), pp. 237-261, George Mason Legal Studies Research Paper No. LS 22-15, Available at SSRN: <https://ssrn.com/abstract=4057783>

(70) Gary Myers, *Muddy Waters: Fair Use Implications of Google LLC v. Oracle America, Inc.*, 19 NW. J. TECH. & INTELL. PROP. 155 (2022)

ワークを提示する学説⁽⁷¹⁾等がみられ、本件判決における fair use 法理の適用手法自体については理論的にも多くの問題点が認識されるものであることが示唆される。

5 日本法に置き換えた議論—APIの著作物性評価と著作権行使

ところで、仮に日本法を前提とした場合には、本件事案における Java API についての著作物性はどのように評価されるべきであろうか。Java API 自体については、Java 言語という特定のプログラム言語で記述された実装コードとしてのプログラムを前提として、一定のタスクを実行させるための特定の処理手順に対応した個々のパーツのようなプログラム（メソッドと呼ばれている）であって、それらの個々に対して特定の名称を付与し、さらにそれぞれの機能ごとに適切な集合たる、クラス、パッケージといった群に分類を行うことによって体系的に構成された類型を形成しているものといえる。

まずは、日本の著作権法 10 条 3 項では、プログラム著作物を作成するために用いるプログラム言語、規約及び解法には著作権法による保護は及ばない旨を定めており、Java API については、「規約」と解釈できるのか否かがとりわけ問題となるであろう。同規定における「規約」とは、「特定のプログラムにおける前号のプログラム言語の用法についての特別の約束」とされており、例えば、特定のプログラムと連結して使うために則る必要のある一定のルール、データを複数の電子計算機で処理するための共通の約束事⁽⁷²⁾が挙げられている。学説では、規約とはプログラムにおける約束事であって、インタフェースについては一般論として「規約」に該当すると解する見解⁽⁷³⁾もみられるところであるが、ここでいう「インタフェース」の概念として、Java API のようにプログラミングに際して共通に用いられるソフトウェアの部品の集合体のようなものまで包含されるのかについては、学説上の明確な見解がみられるとはいえない状況にある。当然ながら Java API が言葉としてはインタフェースという呼称を含むものであることをもって直ちに著作権法上の「規約」と評価できるとは断じ得ない。

まず、Java 言語を用いたプログラム開発に際しては、定型的な機能を実現するプログラムの部品としてのメソッドやクラスが既に用意されており、それを用いて実現できる役割機能である場合には、Java 言語を用いてわざわざ一から独自に別途のコードを記述するという作業が行われることはおそらく限られているものと考えられ、Java API に含まれているパッケージ、クラス、メソッドを呼び出して用いるということの方がむしろ一般的であると考えられる。このため、Java API について、このようなパッケージ、クラス、メソッドといった階層的な構造で形成するという構成自体については、Java 言語における用法の一種として解することは解釈論としてもさほど無理はないものと考えられるかもしれない。このようなことから、Java API の「構成」自体については、Java 言語というプログラム言語における「用法についての特別の約束」に近似するものと解することによって、「規約」として著作権法による保護は及ぶものではないという結論を導くことも一般論としては可能であるものと考えられる。

ところで、著作権法 10 条 3 項とは、そもそも技術的アイデア自体が著作物として保護されないとする考え方を踏まえた確認規定と解されている⁽⁷⁴⁾ところでもあることから、Java API 自体をもって技術的アイデア自体と評価することができるのか検討することも要すると考えられる。Java API については、一定のタスクを実行させるための特定の処理手順に対応している個々のパーツのようなプログラム（メソッドと呼ばれている）に振り分けて、個々に対して特定の名称を付与し、さらにそれぞれの機能ごとに適切な集合たる、

(71) Berlowitz, Joshua, The Five-Factor Framework: A New Approach to Analyzing Public Benefits in Fair Use Cases (April 8, 2022). Columbia Journal of Law & the Arts, Forthcoming. Available at SSRN: <https://ssrn.com/abstract=4079122>

(72) 加戸守行・著作権法逐条講義（七訂新版）129 頁

(73) 中山信弘・著作権法（第 3 版）137 頁

(74) 加戸・前掲注 72・135 頁

クラス、パッケージといった群に分類を行うことで体系的な類型とするという構成自体については、まさに Java 言語によるプログラミング体系についてのアイデアそのものと考えられることから、そのような体系的構成を Google がそのまま Android に取り入れることについては、著作権侵害を構成するものとは評価しえないと考えられる。

他方で、本件事案で Google による複製行為が問題となったのは、Java API の体系的構成を Android に取り入れたということではなく、むしろ、Java API に包含される個々のプログラムのパーツを呼び出すために必要となる名称である宣言コードの部分についての複製行為であることに注意すべきである。宣言コードとして、メソッドやクラス、パッケージに付与された名称としての記述については、主としてその機能を示す単語や文章を短縮化したような記述表現が多くみられることから、一般的には創作性の高い表現と評価できるものではないと考えられる。とはいえ、宣言コードの表現自体に限ってみれば、Java 言語というプログラム言語における「用法についての特別の約束」、すなわち著作権法上の「規約」として評価されるのかという点では疑問が生じる。宣言コードは、本来であれば Java 言語の基本体系から記述される特定の機能について記述された「文章」を、いわばショートカットするような形での「略称」として記述されたものであって、創作性の高い表現とはいえないものの技術的アイデア自体やスポーツやゲームのルールと同質のものとして評価することについては躊躇が生じる。加えて、多種多様な表現形態をとっている宣言コードの記述について、画一的に「規約」に属するものと解することには慎重であるべきであるものと考えられる⁽⁷⁵⁾。

また、異なる観点から、宣言コードについては、そもそも著作権法の下での「プログラム」ではないと解せるのか、という点も併せて検討する必要がある。著作権法の下でのプログラムの定義（法2条1項10号の2）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をいう。」とされており、あくまでコンピュータに対しての指令を構成するものに限定されると解される。Java API の場合、これまでの検討から明らかなように、パッケージやクラスに含まれる各機能がメソッドにまで「分解」され、最終的には、対応する実装コードが実行されることによって「電子計算機に対する指令」となっている。その意味では、Java API の宣言コードについては直接的には電子計算機に対する指令となるものではなく、プログラム著作物であることを否定できるのかということが一応問題となる。この点で、IBF ファイル事件⁽⁷⁶⁾では、「電子計算機によるプログラム処理に当たり、あるシステムにおけるプログラムを稼働させ一定の処理をさせるためには、そのプログラムの他、それに処理情報を与えるデータが必要であるが、システムの効率上、データを本体プログラムとは別個のファイルに記録させることがよく行われる。その場合、該ファイルは、プログラムに読み取られその結果電子計算機によって処理されるものではあるが、電子計算機に対する指令の組合せを含むものではないので、著作権法上のプログラムではない。」「データを記述するに当たり、プログラム自身が規定した一定の記号又は文字（以下「記号等」という。）が記述されていれば、プログラムがそれを読み取ってその記号等に意味付けられた処理を行うとしても、それは、プログラムがその記号等をデータとして読み取り所定の処理を行うものに過ぎず、その記号等をもって電子計算機に対する指令であるということとはできない。したがって、また、そのような記号等が付されたデータをもって、著作権法上のプログラムであるということとはできない。」とされている。Java API の宣言コードについても、直接的にはハードウェアに対する「指令」ではなく、一見上記の「データファイル」であるかのようにも思われるが、宣言コード→Java 言語によるソースコード→オブジェクトコードと変換されることにより、最終的には実装コードとして、ハードウェアによって実行される命令群に変換される以上、データと解することはできないであろう。

(75) 個別の API についてみれば、ほぼ「規約」に相当するものとして、著作権法による保護は肯定しえないと解せるものの中には存在しているといえよう。

(76) 東京高決・平成4年3月31日・平成3年（ラ）142号・知的裁集24巻1号218頁

以上のように、日本法の下ではJava APIの構成自体は「規約」と評価できるかもしれないし、宣言コードそれぞれについても著作物性を充足しないものとして評価されるものは個別的にみれば少なからず存在するものと考えられる。しかしながら、他方で、Java APIの宣言コードすべてについて、画一的に著作物性がないものとして著作権法の下での保護を否定することについては慎重であるべきと考えられる。もちろん、個々の宣言コードの著作物性の評価の結論としては、ありふれた表現として著作物性が否定されるものも数多く存するとは考えられるものの、宣言コードといえども創作性は低いものの著作物性を肯定できるものも存在する余地は否定できないと考えられる。したがって、著作物性を肯定されうる宣言コードについても一部には存在するものと解すべきことになろう。すると、このような場合、本件事案におけるGoogleのような行為を行う者に対する著作権エンフォースメントをどのように考えるべきであろうかという問題が生じる。創作性の低い著作物であるとしても、一般的にはデッドコピー行為に対して複製権による権利行使は肯定されるものと考えられ、著作物性が肯定されるAPIの宣言コードをデッドコピーしてAndroidに利用したGoogleの行為に対して、著作権行使は日本法の下では許容されるものであろうか。

そこで、日本版fair use規定と位置付けられることの多い著作権法30条の4の適用が目されることとなる。しかしながら、本件事案で問題となったような事実状況が、同法の規定する「当該著作物に表現された思想又は感情を自ら享受し又は他人に享受させることを目的としない場合」に該当するものと解することは困難であると考えられる。同規定にいう思想又は感情の「享受」については、行為者の主観、利用の行為態様、利用に至る経緯等の客観的・外形的な状況も含めて総合的に考慮するものとされている⁽⁷⁷⁾ところ、立法趣旨としては、プログラム著作物の場合、プログラムを実行することを通じて、その機能の効用を得ることに向けられた行為であるかという観点から評価するものとされている⁽⁷⁸⁾ところである。

そうすると、Java APIの宣言コードに表現された「思想」を享受するとは、まさに、プログラム開発者が、個々のAPIによって実現されている特定の役割機能を自らが作成しようとするプログラムに取り入れるに際して、宣言コードの表現を通じて、個別のクラスやメソッドを認識して活用するというにほかならないものと考えられるのであって、GoogleがJava APIの宣言コードを複製して、Androidに取り込んで利用する行為とは、宣言コードという表現の下にある思想又は感情を享受する目的での利用行為と評価せざるを得ないものと考えられる。したがって、本件事案でGoogleによってなされたような、著作物性が肯定される宣言コードの複製利用行為については複製権侵害を構成して、著作権法30条の4の適用により非侵害との結論を導くことは困難であるように考えられる。

他方で、一部の（著作物性がある）宣言コードについての複製権侵害を前提とした著作権行使、特に差止請求権の行使をそのまま肯定することは、Java APIの一部の宣言コードの使用を排除することを意味するものであって、結果的にAndroidで動作するアプリケーション開発をJava言語で行う行為自体を広く抑止・排除する効果にもつながる可能性も懸念され、妥当ではないと考える。

そもそも、Java言語を用いたアプリ開発は従前から非常に広く定着していたということ、著作権者であるOracle(Sun)はJava API自体を広く公開した上で、事実上、これを取り入れた個別の実装について他社が独自に開発提供すること自体は認めており（認定を受けるかは別の話として）、その限りでは、APIに係る著作権が仮に成立するとしても、その権利行使については、広範な範囲で積極的に行わない意思があったと解する余地はあったこと、Googleが自らの開発したOS(Android)の開発に際して、宣言コードの複製利用によってJava APIの「取り込み」を行ったことについては、その結果がGoogleの事業活動として成功裡に成果を得られたか否かという評価のいかんにかかわらず、Java言語を用いたスマートフォンにおけるアプリ開発という形での「バリアフリー」化を進めて、社会における技術コミュニティに寄与をしたと

(77) 加戸・前掲注72・282頁

(78) 加戸・前掲注72・283頁

いう側面で公益に資する要素は相応にあったことは否定できないのであって、情報技術におけるイノベーション促進に間接的に寄与しうるものであったこと、等を考慮すると、複製権侵害による差止請求権の行使については否定する結論を導出することの合理性は十分に見いだせるように考えられる⁽⁷⁹⁾。

したがって、日本法の下での本件事案の帰趨を考えるとすれば、APIにおける宣言コードの部分についての著作物性を画一的に否定することは困難と考えられるため、一部の著作権侵害（複製権侵害）の成立については否定しえないものと考えられる。しかしながら、そのエンフォースメントという局面では、そもそも Java API について著作権者自身が広く公開して、オープンなプログラム開発環境を醸成してきたこと、それに伴って Java API に係る著作権行使については一定範囲で留保する意思が存したことが見いだせること、Java API における宣言コードの部分の著作権行使による差止請求を肯定した場合に Android で動作するアプリケーション開発へもたらされる社会的・技術的影響が非常に大きいこと、といった観点から評価すると、FRAND 宣言された標準必須特許における特許権行使による差止請求の問題に近似するものと理解できることから、結論としては、権利濫用論⁽⁸⁰⁾によって差止請求権行使は否定されることが妥当であると考ええる。反面で、著作物性が肯定される一部の宣言コードの複製権侵害に伴う損害賠償請求については、必ずしも否定されるべきものではないようにも考えられる。

Google v. Oracle 事件連邦最高裁判決から得られる日本法への示唆として、API の著作物性のいかんにかかわらず、その権利行使を fair use 法理の適用によって否定したという結論だけ見ると、fair use の成立を積極的に肯定することで著作権のエンフォースメントを情報技術イノベーション促進に歩調を合わせるべく一定の制約を課した判決として肯定的に捉えられる余地もあるかもしれない。しかしながら、そのロジックを検討する限りでは、先の検討からも提示してきたように、既存の fair use 法理の適用事例と比べて大きく異なっている点も明らかに指摘できるところであって、あくまでも著作物性があることを仮定した事例的判断にとどまるに過ぎず、fair use 法理による結論の不安定さや解決されるべき法的論点が回避されているといった多くの課題も併せて含むものであって、本件事案の連邦最高裁判決の見かけ上の考え方や結論を直ちに日本法の解釈論の発想に取り入れるといった姿勢を採ることには極めて慎重であるべきだろう。

6 展望—情報技術イノベーション促進と著作権エンフォースメントの調和

Google v. Oracle 事件は、Java 言語というプログラミング言語を用いたアプリケーション開発において重要な役割を果たす API を構成している特定部分のプログラム記述表現に係る著作権侵害が争われた事案であったが、その本質からみれば、既存の広く利用されている情報技術をベースとして新たな情報技術イノベーションの促進がなされようとする際に、既存の技術を構成する要素の法的保護について、どの程度まで図られるべきであるのかという問題状況を巡って、著作権のエンフォースメントとの調和のあり方が表面化した一事例として捉えなおすことができるであろう。そして、それに対する一つの回答として、連邦最高裁判決では、アメリカ著作権法における fair use 法理を積極的に活用するという選択を行ったものと位置付けることができる。

もっとも、API の法的保護を巡っては、その著作物性について連邦最高裁判決で明確な判断が提示されることも大きく期待されていたところ、結論として、この点についての判断を提示することを取って回避し

(79) アメリカ法の学説では、この点について言及するものは現時点ではあまりみられないが、Gary Myers, supra note 70 は公益性に係る評価の局面で本件判決が差止めの影響といった法的救済の影響についての評価がなされていないことを指摘している。

(80) 日本法における判例として、標準必須特許における FRAND 宣言に伴う一定の効果として、技術標準に係る特許発明の実施に係る差止請求権行使についての制限を肯定する判断を提示しており（知財高決・平成 26 年 5 月 16 日・判時 2224 号 89 頁、損害賠償請求権について知財高判・平成 26 年 5 月 16 日・判時 2224 号 146 頁）、その理論的根拠として権利濫用論が適用されている。本件事案においても、本文で示したように、このような標準必須特許を巡る権利行使に通底する問題状況がみられることから、権利濫用論を基礎として導出しうるものと考えられる。

たともいえる。このことは、今後の課題として、特定のプログラミング言語におけるAPIの法的保護のあり方をどのように考えるべきかという新たな問題提起を残したともいえる。Java言語のようにプログラム言語として非常に幅広く普及利用されている状況が成立している場合には、Java言語を用いたアプリケーションプログラム開発という技術環境に対するアクセスが社会に対して広く確保・維持されることにもはや相応の公益性が発生しているといえる。他方で、Java言語を用いたプログラム開発を効率的に行うためには、頻繁に用いられる機能等について再度実装する手間を省くことのできるJava APIを活用することが、プログラミングの現状では事実上ほぼ不可欠な状況となっていること、Java APIを構成する各要素自体の表現（とりわけ宣言コード部分）については、中には著作物性を否定しえないものがあるとしても、その技術的性質を踏まえると創作性の高いものと評価しうるものではないこと、等の事由も指摘できることである。

そこで、Java言語を用いたアプリケーションプログラム開発という技術環境の確保・維持及びそのような技術環境へのアクセス保護のために、Java APIについては、著作物としての保護自体を典型的に否定することによって著作権のエンフォースメントの懸念から画一的に解放するという考え方も成り立つのかもしれない。このような発想を前提とすると、例えば、APIであることだけを根拠としてアприオリに著作物性を否定するという考え方や、さらにそれを一歩進めて、制度設計として、APIのような類型を予め定義して典型的に著作物性を否定する立法論などが具体的な方策として提示できるであろう。しかしながら、このような方策の下では、APIのうち相応のものは著作物性の一般的解釈の帰結としても同様に著作物性が否定されるものの、一般論としては著作物性を否定しえないAPIについてまでも法政策的に著作物性を否定することになってしまい、そこに理論的な不連続性が発生することが大きく懸念されることである。また、立法論として著作物性がアприオリに否定されるプログラムとして、例えばAPIのような類型を予め定義するにしても、著作物性がアприオリに否定されるAPIなるものの定義概念をどのように規定するのか、あるいは個別具体的なものの定義概念への該当性の解釈を巡って、新たな情報技術が生成・発展するに伴って新たな混乱、紛争へと場が移るだけで、根本的な問題の解決とならなくなる懸念される。

そこで、アイデアと表現の二分論の下でアイデアそのものの著作物性を否定するのみならず、APIについては、それによって実現される機能というアイデアへのアクセス手段としての表現として捉えなおすことで、仮にその表現自体の著作物性が一般的解釈の帰結として直ちに否定されないものであるとしても、その幅広い使用を前提として社会的に広く活用される状況に至っている等の一定の要件を充足する場合には標準必須特許における特許権行使の制限と同様の形で差止請求権行使を制限するといった考え方をとることで、情報技術イノベーション促進にとって予見可能性が高まることが期待できるようにも考えられる。そして、その一定の条件としては、例えば、本件事案のように、当該プログラミング言語自体が社会において非常に幅広く普及利用されている状況が当該プログラミング言語開発者の主導的・積極的役割によって成立していること、当該プログラミング言語におけるAPI利用が用いたアプリケーションプログラム開発という技術環境の確保・維持にとって不可欠と評価しうる程度に至っていること、当該プログラミング言語に係るAPI利用が当該APIに係る著作権者によって積極的に促進された事情のあること、といったものを差し当たり列挙できるものとする。

本件連邦最高裁判決でも、APIの著作物性についての扱いに何ら手を加えることなく、著作権のエンフォースメント段階で何らかの調整を行うという発想として、fair use法理による調整を行ったものと解することができるものの、事後的かつ事案依存的な解決であることから、このような手法では予見可能性は低いものとならざるを得ないことが懸念される。とりわけ、本件判決におけるfair use法理の当てはめを検討する限り、既に検討したように個々のファクターの当てはめの結論は予見可能性に乏しいものとなるものが少なくないであろう。イノベーションに資する可能性のある技術開発における著作権行使の可能性という観点については予見可能性が確保されることが極めて大きな意味を有するものと考えられるため、事案依存性が高い（とりわけ陪審によるトライアルの結果も大きく左右しうる）fair use法理だけで対応することは、不安

定かつリスクであることが懸念されるところである。

実際、若干異なる文脈⁽⁸¹⁾であるが、ソフトウェアエンジニアの開発環境をサポートするプラットフォームである GitHub が、あいまいな形での著作権侵害主張が FUD (fear, uncertainty, and doubt) を生み出し、ソフトウェアにおけるイノベーションを阻害するとする見解⁽⁸²⁾を示しており、情報技術イノベーションにとっては、関連する知的財産権エンフォースメントの予見可能性が高まるのが極めて大きな意義を有していることを示唆するものといえる。本件事案のように 10 年にも及ぶ訴訟の結果として fair use 法理の適用によって非侵害との結論を得たとしても、まさに FUD の長期にわたる呪縛からは逃れられないのであって、進化スピードの速い情報技術分野におけるイノベーションにとって保護とエンフォースメントのバランスを時機に合った形で調整できるのかという点こそが大きな課題として認識されるところである。このため、情報技術分野における API を活用した更なるイノベーション創出機会の確保という観点から著作権エンフォースメントを調整させる道具立てとして捉える限り、fair use 法理はあくまで補助的な道具立てといえるもので、「最終手段」として位置付けられる方が望ましいものと考えられる⁽⁸³⁾。

学説においても、fair use 法理の適用による著作権侵害を否定するだけでは、予見可能性が低いことに加えて、fair use の成否判断の前提として陪審による事実判断を介することの不安定性、本件事案の Oracle と Google の場合と異なり、直接的な競業者間での争いの場合にうまく活用できるか未知であることを指摘して⁽⁸⁴⁾、本件判決の「射程」は狭いと認識しているのがみられる。また、仮に fair use 法理の適用が認められたとしても、あくまでも当事者限りの効果しか持たない。このため、例えば、本件事案で著作権侵害が否定されても Google 以外の事業者が Java API を利用して別のプラットフォームで用いる場合には結論は変わる可能性があることが学説では指摘されており、fair use 法理の結論について対世効を認める形へ改めることが提案⁽⁸⁵⁾されている。

以上、Google v. Oracle 事件連邦最高裁判決の検討を契機として、情報技術分野におけるイノベーションの促進にとって著作権のエンフォースメントの調整のあり方がどのようにあるべきであるのか、その方向性を模索してきた。情報技術イノベーションにとって重要な要素（その典型例が API といえる）の著作権による保護とエンフォースメントの調整としては、端的には、イノベーションにとって重要な技術要素であって著作権法による保護対象への該当性が議論になりうるものについては、ある意味で一律で法的な保護対象の俎上から除外しようとする「入口規制」的な考え方と逆に、著作物性段階での調整は原則行うことなく、権利行使段階において、例えば fair use 法理を用いて、事案によって調整を考慮する「出口規制」的な考え方、いずれの方向性を原則とすることを目指すのが望ましいのか、という課題として捉えることができるであろう。

そして、情報技術イノベーションの発展スピードを考慮した場合、少なくとも無方式主義という権利発生構造を有する著作権については、そのエンフォースメントの予見可能性を高めることの意義は極めて大き

(81) ソフトウェアの機能的側面の著作物性について争われ、目下その帰趨が注目されている事案 (CAFC に係属中) である *SAS Institute, Inc. v. World Programming Limited*, (Appeal #21-1542) について提出されている amicus に表明されている諸見解という文脈である。

(82) Amicus of GitHub (<https://fedcircuitblog.com/wp-content/uploads/2021/05/Github.pdf>)

(83) もちろん、このことは fair use 法理自体の意義を否定するものではなく、学説では、むしろ特許法においても fair use 法理に相当する権利調整法理 (patent fair use) を取り入れることを提案する見解 (例えば, Strandburg, Katherine J., Patent Fair Use 2.0 (May 7, 2011). UC Irvine Law Review, 2011, NYU Law and Economics Research Paper No. 11-18, Available at SSRN: <https://ssrn.com/abstract=1835007>) もみられるところである。そして、特許法における権利調整法理としての fair use 法理の役割機能は、おそらく著作権法の場合と比べるとかなり異なる可能性があって、固有の役割意義が見いだせるのかもしれない。

(84) Lemley, Mark A. and Samuelson, supra note 66

(85) Bell, Abraham and Parchomovsky, Gideon, Propertizing Fair Use (2021). Virginia Law Review, Vol. 107, p. 1255, 2021, U of Penn, Inst for Law & Econ Research Paper No. 21-07, Bar Ilan University Faculty of Law Research Paper No. 21-06, San Diego Legal Studies Paper No. 21-001, Available at SSRN: <https://ssrn.com/abstract=3782545>

なものであって、基本的には「入口規制」を中心に構成することが望ましいと考えられるところである。あるいは、「出口規制」の形をとるとしても、いかなる要件を充足した場合に著作権のエンフォースメントが制約されるのか、理論的な枠組みとして一定程度明確にされていることが望ましく、fair use 法理のように個別事案への当てはめに大きく依存する型の「出口規制」的な考え方を中心に用いることには限界があるものとする。もちろん、具体的にはいかなる条件の下であれば著作物性を否定することの合理性が認められるのか、エンフォースメントの制限についてどのような要件を定立しておくのかという次なる大きな課題について対峙することは避けられないであろう。