

# Oracle America, Inc. v. Google Inc

SOFTIC判例ゼミ2021

第3回2021年9月17日

伊藤・ガニング

# もくじ

1. はじめに
2. フェアユース
3. APIとは
4. 本事件の経緯
5. 最高裁審理

# はじめに

- 概要
- 審理経緯概略

## 事件概要

Googleが構築したアンドロイドプラットフォームには、「JAVA」の一部のコードが複製して利用されている。「JAVA」の権利者であるOracleの提訴により、Googleによる著作権侵害が争われた。地裁、CAFCにおいて著作物性の有無、及び、著作物性が肯定された場合のフェアユースの該否が判断されたのち、最高裁において、該当コードの著作物性を仮定した上で、フェアユースが適用され、著作権侵害が否定された。

当事者 : Oracle America Inc. v. Google Inc.

著作物 : Java API の Declaring Code

争点 : APIの著作物性、フェアユース

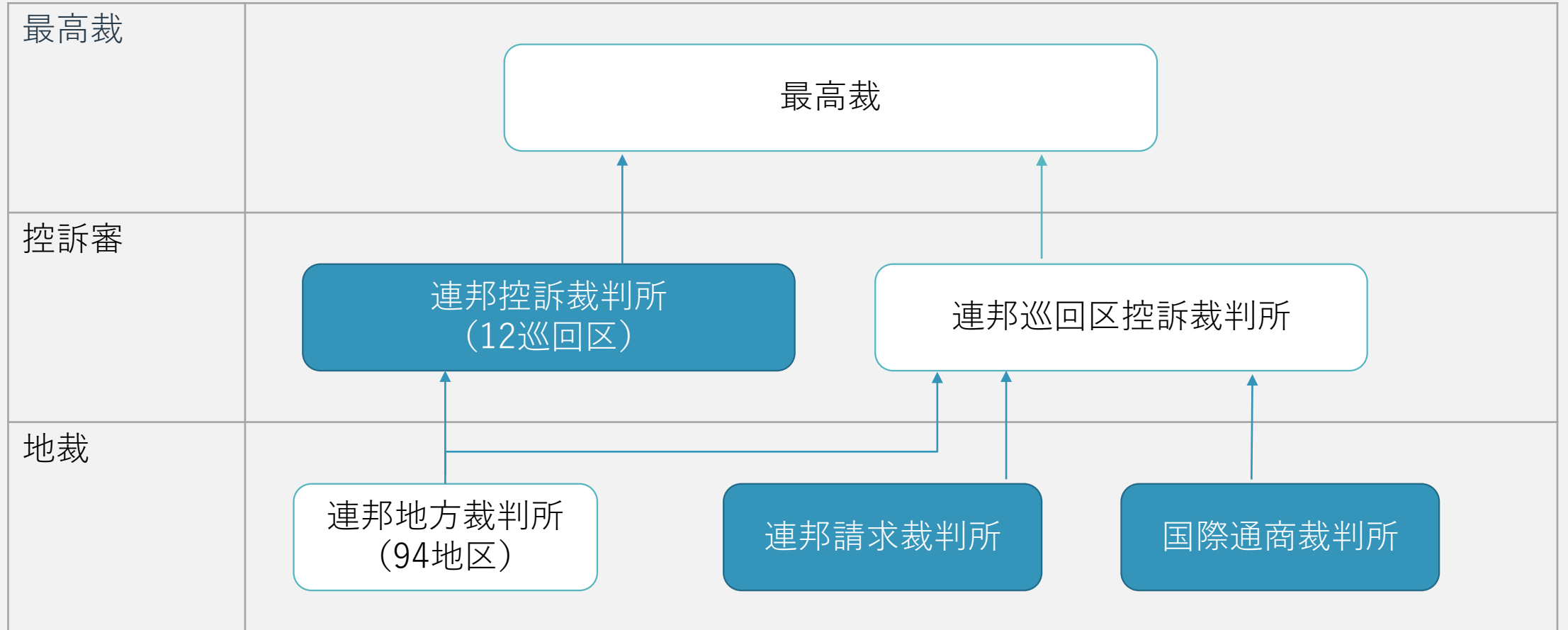
最高裁判決 : 著作権侵害が否定されGoogle勝訴

# 審理のおおまかな経緯

地裁：カルフォルニア連邦地方裁判所、 CAFC:連邦巡回区控訴裁判所

2010	提訴	特許権(2件)及び著作権の侵害	原告:Oracle 被告:Google
2012	地裁1	特許侵害否定、 著作物性否定により著作権非侵害	Oracle 控訴 Google 控訴
2014	CAFC1	著作物性肯定	Google 上告
2014	最高裁1	著作物性肯定 フェアユース適用に関し差戻し	裁量上訴不受理 <b>著作物性肯定</b>
2015	地裁2	フェアユース肯定	Oracle 控訴
2018	CAFC2	フェアユース否定	Google 上告
2021	最高裁2	フェアユース肯定	裁量上訴受理 <b>フェアユース肯定、CAFCに一部差戻し (フェアユースについて)</b>

# (参考) 合衆国連邦裁判所制度



## 2. フェアユース

- 条文
- 評価要素
- 主な判例法理

# 条文

## 第107条 排他的権利の制限:フェア・ユース

第106条および第106A条の規定にかかわらず、批評、解説、ニュース報道、教授(教室における使用のために複数のコピーを作成する行為を含む)、研究または調査等を目的とする著作権のある著作物のフェア・ユース(コピーまたはレコードへの複製その他第106条に定める手段による使用を含む)は、著作権の侵害とならない。著作物の使用がフェア・ユースとなるか否かを判断する場合に考慮すべき要素は、以下のものを含む。

- (1)使用の目的および性質(使用が商業性を有するかまたは非営利的教育目的かを含む)。
- (2)著作権のある著作物の性質。
- (3)著作権のある著作物全体との関連における使用された部分の量および実質性、および
- (4)著作権のある著作物の潜在的市場または価値に対する使用の影響。

上記のすべての要素を考慮してフェア・ユースが認定された場合、著作物が未発行であるという事実自体は、かかる認定を妨げない。



# 評価要素

フェアユース成立の可否は、主に、下記の4要素に照らして総合考量される。

第1要素	使用の目的と性格	(1) 商業的目的	非営利目的/商業的目的*1
		(2) 変容的使用	損害不存在の推定*2
		(3) 悪意	
第2要素	著作物の性質	芸術的/事実的/機能的	
第3要素	使用の量と実質性	(1) 実質性	表現の核心的部分/周縁的部分
		(2) 全体/一部	目的に照らして合理的範囲内
第4要素	使用による 市場への影響	(1) 市場の範囲	既存市場および潜在的市場
		(2) 影響の程度	現実的損害<損害の有意的可能性<損害のおそれ

\*1 Sony Corp. v. Universal City Studios, 464 U.S. 417 (1984) ベータマックス事件

\*2 Campbell v. Acuff-Rose Music, Inc., 510 U.S. 569 (1994) プリティ・ウーマン事件

# 主な最高裁判例

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

— プリティ・ウーマン事件 — (変容的利用 transformative)

パロディに関する教示的判例。「Oh! Pretty Woman」の著作権者である原告が、当該楽曲のパロディである曲を製作し発表したラップグループを提訴した。フェアユースの判断において、著作物の利用がtransformative(変容的)であるかが重要。商業目的が前提のパロディについてもフェアユースが適用され得ると判断し、フェアユース成立の判断において、第1要素「使用の目的と性格」が優先的に考慮される例となった。

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

- 有名楽曲のパロディーに関しフェアユースの適用を肯定した事例
- [事実概要]

Acuff-Rose Music, Inc. は、ラップ音楽のグループである 2 Live Crewのメンバーとそのレコード会社たるに対し、2 Live Crew の曲「Pretty Woman」が、Roy Orbisonのロック・バラード「Oh, Pretty Woman」に関する Acuff-Rose の著作権を侵害したと主張して、訴訟を提起。地方裁判所は、2 Live Crew の曲が原曲のフェアユースを構成するパロディーであると判示した。控訴裁判所は、このパロディーの商業的性質は第107条で問題とされる四つの要素の第1の要素に基づきアンフェアと推認される、原作の「核心」を取り出して新作品の「核心」とすることによって 2 Live Crewは第107条の第3の要素に基づき質的に過剰に奪取した、商業的利用に付随する推認によって第107条の第4の要素が目的とする市場の害が立証された、と判示して、地裁判決を破棄、差し戻した
- [結論]

著作権法 107 条(i)～(iv)の各ファクターは、ケースバイケースで各状況に応じて当てはめがなされなければならない。4 要素の当てはめを行い(何れの要素についても、どちらに有利かの明示的な言及なし)、フェアユースに該当し得るとした上で、控訴審判決を破棄、差戻しとした。

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

		判断	評価
第1要素	営利目的の利用	<ul style="list-style-type: none"> <li>作品の商業的又は非営利の教育的な性格は「決定的なものではなく」むしろ「フェアユースの判定において他のものと共に考慮される」。</li> <li>作品の商業的な性質は、このファクターにおける問題の1つの要素にすぎない。</li> </ul>	パロディのような変形的な使用がフェアであるかどうかを判定するに際しては、第1の要素である利用の性格と目的を問題にするにも、第4の要素である市場の被害を問題とするにも、こうした証拠上の推定を利用することはできない。
	変容的使用	あらゆる著作権を有する作品の商業的使用は、不公正であると推定される(Sony Case (データマックス Case. 464 US 417))とするのは、パロディーの商業的な性格に実質的に決定的なウエイトを置いたもので、誤り。	
	悪意の利用	—	
第2要素	著作物の性質	パロディーに関しては、公衆に知られた作品を複製することがほとんど不可避。パロディーに関するケースを解決するにあたり、「利用された著作物の性質」はさほど重要ではない。	
第3要素	使用の量と実質性	<ul style="list-style-type: none"> <li>法律問題として 2Live Crew が過度に原曲から複製を行っている、とするのは誤り。</li> <li>例え、原曲の歌詞の最初の一節と、特徴的なオープニングのベースのリフが原曲の「中心」に通じるかも知れないとしても、それはパロディーのためにその原曲を思い出させるためのものであり、それこそがパロディーの重要な点である。</li> <li>それより後の部分では、2Live Crew は、原曲の歌詞からは明らかに離れ、別の独特な音楽を作り出している。歌詞に関しては、その歌のパロディーとしての性格からすれば、過度とはいえない。音楽の面では、本裁判所は、ベースのリフの繰り返しが過度の複製に当たるかについて意見を表明しないが、本件音楽のパロディーとしての目的・性格、変化的要素、市場における代替可能性の検討等に照らして複製部分の評価をするために、差し戻す。</li> </ul>	
第4要素	市場への影響	<ul style="list-style-type: none"> <li>Sony Case に基づき2Live Crewの使用が商業的であることから「重大な市場における損害の可能性」を推定するのは誤り。Sony Case における「市場における損害の推定」は、単なる複製にあたらぬ案件では、適用されない。このファクターで認識されるべき損害は、市場における代替によるものであって、批評から生ずる損害ではない。パロディーに関しては、原作の代替としての役目を果たすことは考えがたい。</li> </ul>	
総合評価			フェアユースに該当し得る

# (参考) Sony Corp. of America. v. Universal City Studios, Inc., 464 U.S. 417 (1984)

## — ベータマックス事件 — (非商業的・非営利的使用)

➤ 有名楽曲のパロディーに関しフェアユースの適用 を肯定した事例

### ➤ [事実概要]

テレビ番組について著作権を有するUniversal City Studios, Inc (ユニバーサル) が、Sony Corp. of America (ソニー) に対し、ソニー製のVTRを使用してテレビ番組の録画をしている消費者の行為は著作権侵害であり、また、当該VTRを製造して一般に販売している点でソニーも著作権侵害の責めを負うものであるとして、差し止め、損害賠償などを求めた。家庭内の録画はフェアユースに該当し、著作権侵害にならないとした第一審の地方裁判所の判決は、第二審である控訴裁判所により逆転されたが、最高裁判所は控訴裁判所の判決をさらに逆転し、5対4の僅差でソニーの侵害を認めなかった。

### ➤ [結論]

連邦最高裁判所は、利用者の多くが家庭における後日の視聴 (time-shifting) を目的としていることから、フェアユースを構成すると判断。

## 3. API

- APIとは
- Java APIとAndroidプラットフォーム

# APIとは

API=Application Programming Interface

“APIとは、あるコンピュータプログラム(ソフトウェア)の機能や管理するデータなどを、**外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約**のこと。”

- ▶ 機能: OSやミドルウェアなどの形でまとめて提供されている**汎用的な機能**を呼び出す。プログラミング言語の提供する機能や言語処理系に付属する標準ライブラリの持つ機能を呼び出すための規約などを含む場合もある (Java APIなど)。
- ▶ 利点:
  - 個々のソフトウェアの開発者が毎回すべての機能をゼロから開発する無駄を省く。(個々の開発者はAPIに従って**機能**を呼び出す**短いプログラム**を記述するだけ)で、その機能を利用したソフトウェアを開発できる。
  - 同じAPIが実装されていれば別のソフトウェア上でそのまま動作させることができるため、APIを引き継ぐことで、旧バージョン向けに開発されたソフトウェアの動作を維持しつつ、新しい機能を拡張しやすい。



# Java Platform SE で提供されるAPI

パッケージ  
クラスを機能  
ごとにまとめ  
たもの

クラス  
機能別プロブラ  
ムを呼び出すカ  
テゴリ

Java(tm) Platform Standard Edition 8

概要 パッケージ クラス 使用 階層ツリー 非推奨 索引 ヘルプ

前のクラス 次のクラス フレーム フレームなし

サマリー: ネスト | フィールド | コンストラクタ | メソッド 詳細: フィールド | コンストラクタ | メソッド

compact1、compact2、compact3

java.lang

## クラスMath

java.lang.Object  
java.lang.Math

```
public final class Math
extends Object
```

Mathクラスは、指数関数、対数関数、平方根、および三角関数といった基本的な数値処理を実行するためのメソッドを含んでいます。

StrictMathクラスの一部の数値メソッドとは異なり、Mathクラスの等価関数のすべての実装は、ビット対ビットの同じ結果を返すようには定義されていません。このゆるやかさによって、厳密な再現性が要求されない実装においてパフォーマンスの向上が可能になります。

デフォルトでは、Mathの多くのメソッドは単に、StrictMath内の対応するメソッドを実装として呼び出します。コード・ジェネレータは、プラットフォーム固有のネイティブ・ライブラリまたはマイクロプロセッサの命令がある場合は積極的に使用し、高性能なMathメソッドの実装を提供します。そのような高性能な実装でもMathの仕様に準拠する必要があります。

実装仕様の品質は、2つのプロパティ、返される結果の正確さ、およびメソッドの単調性と関係があります。浮動小数点Mathメソッドの正確さは、ulp (最新位置の単位)で計測されます。浮動小数点形式が指定された場合、特定の实数値の... (最新位置の単位)で計測されます。浮動小数点形式が指定された場合、特定の实数値の...

メソッド  
機能別プロ  
グラム



# Java API のコードの例

共通部分

```
java.lang.Math.max()
```

package      class      method

Methodで呼び出すコードは独自に開発可能

**An API Provides Access to Source Code**

API	Source Code Implementation
choose greater of two numbers <code>max ()</code>	tells the computer what to do <pre>public static float max(float a, float b) {     if (a != a) return a; // a is NaN     if ((a == 0.0f) &amp;&amp; (b == 0.0f)         &amp;&amp; (Float.floatToIntBits(a) ==             negativeZeroFloatBits)) {         return b;     }     return (a &gt;= b) ? a : b; }</pre> <p><b>OpenJDK 6</b></p>

**An API Provides Access to Source Code**

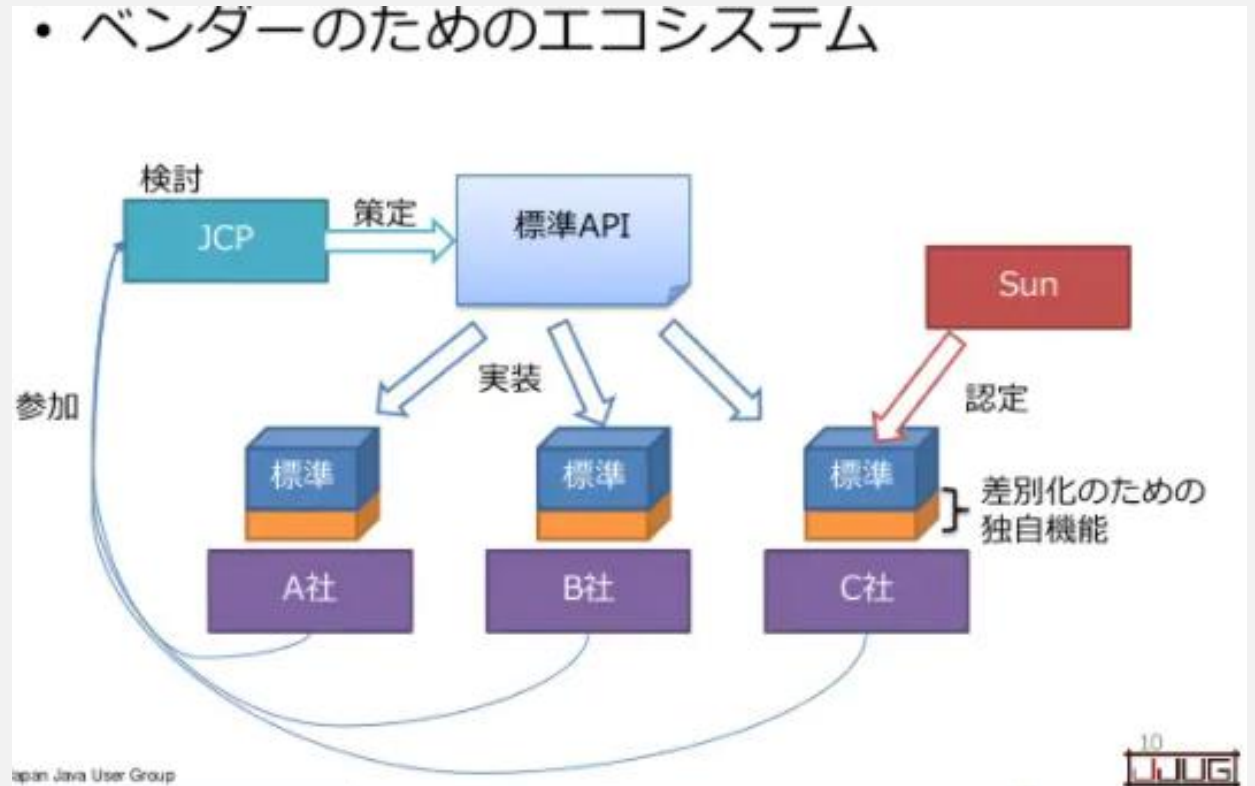
API	Source Code Implementation
choose greater of two numbers <code>max ()</code>	tells the computer what to do <pre>public static float max(float f1, float f2) {     if (f1 &gt; f2) {         return f1;     }     if (f1 &lt; f2) {         return f2;     }     /* if either arg is NaN, return NaN */     if (f1 != f2) {         return Float.NaN;     }     /* max(+0.0,-0.0) == +0.0 */     /* Float.floatToRawIntBits(0.0f) == 0 */     if (Float.floatToRawIntBits(f1) != 0) {         return f2;     }     return 0.0f; }</pre> <p><b>Android</b></p>

**An API Provides Access to Source Code**

API	Source Code Implementation
choose greater of two numbers <code>max ()</code>	tells the computer what to do <pre>public static float max(float a, float b) {     // this check for NaN, from JLS 15.21.1, saves a method call     if (a != a)         return a;     // no need to check if b is NaN; &gt; will work correctly     // recall that -0.0 == 0.0, but [+]-0.0 - [+]-0.0 behaves special     if (a == 0 &amp;&amp; b == 0)         return a - -b;     return (a &gt; b) ? a : b; }</pre> <p><b>GNU Classpath 0.9.8</b></p>

## (参考) Java API (2007以前)

- Java Community Process (JCP)が仕様を確定。  
利用者は、技術互換Kit (TCK)により、Sunより認定を受ける。
- 但し、当時のライセンスは、改変部のコードの開示が要求されていたため、Googleは認定を受けずに、独自の開発を行った。



# (参考) Java API のライセンス(2019年4月以前)

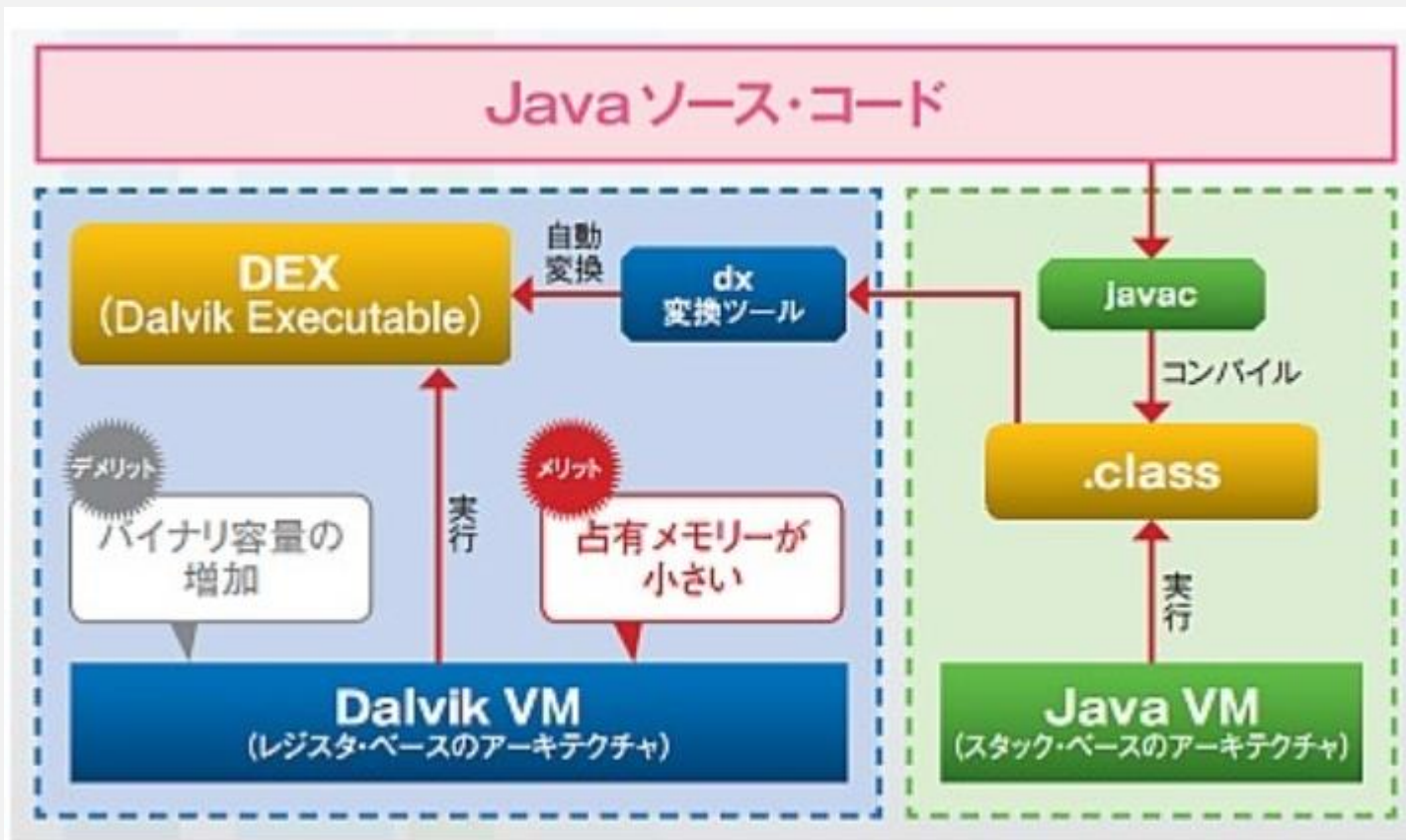
Java SEプラットフォーム製品及びJavaFXに関するオラクル・バイナリ・コード・ライセンス契約 【日本語参考訳】

## 補足ライセンス条項 D.再頒布可能ファイルの頒布許可

- 本契約の条件、並びにREADMEファイルに規定されている制限事項及び除外事項(本補足条項の再頒布に関するJavaテクノロジーに関する制限事項を含むが、これに限定されません)に従って、オラクルはお客様に対し、以下を条件として、READMEファイルで再頒布可能ファイルとして明確に特定されているファイル(「再頒布可能ファイル」)を複製及び頒布するための非独占的、譲渡不能の限定的なライセンスを無償で許諾します。:
  - (i) 再頒布可能ファイルを、完全かつ改変されないまま、プログラムの一部としてバンドルされた状態でのみ、頒布すること、
  - (ii) プログラムが重要かつ主要な機能を再頒布可能ファイルに与えること、
  - (iii) 再頒布可能ファイルのコンポーネントを取り替える目的で追加的なソフトウェアを頒布しないこと(適用されるREADMEファイルに別段に明記されている場合を除く)、
  - (iv) 再頒布可能ファイルに記載されているいかなる所有権表示や通知も除去又は変更しないこと、
  - (v) (a) 本契約の完全かつ変更されていない複製である、又は(b) 本契約に含まれる条項と一致するオラクルの利益を保護し、セクションHに記載されている通知を含む、ライセンス契約に従ってのみ再頒布可能ファイルを頒布すること、
  - (vi) すべてのプログラム及び/又はソフトウェアの使用又は頒布に起因する第三者による請求、訴訟又は法的措置に関連して生じるあらゆる損害、コスト、損害賠償額、和解金及び/又は費用(弁護士費用を含む)について、オラクルとそのライセンサーを防御し、それらに補償することに同意すること。
- 本セクションDに記載されているライセンスは、セクションGに特定されているソフトウェアには適用されません。

# Androidの開発環境

Android Runtime: アプリケーションの実行環境



- Androidのソースコードは基本的にJavaで記述する。
- アプリケーション開発では、Javaのクラス・ファイル(.class)を作成し、Androidが提供している開発ツール「dx変換ツール」で、Javaのクラス・ファイルからDEXに自動で変換する。
- アプリケーション開発者はDEXの作成を意識しないでよく、Java言語での開発と同じ感覚でコア・ライブラリを扱える。

# (参考) Android プラットフォームのライセンス

## ➤ Androidのライセンス形態

AndroidはApache License 2.0により公開されているため、開発したアプリケーションはApache License 2.0を適用でき、ソースコードを公開する必要はない。

Android上でアプリケーションを開発するには、アプリケーションの有償・無償にかかわらず、登録する必要もライセンス費用を支払う必要もありませんが、開発したアプリケーションをGoogle Playで公開するには登録が必要です(登録費用25ドル)。

Androidのソースコードは、Apache License 2.0とBSD Licenseに準じ、改変可能である。

## ➤ GPLライセンス

AndroidのLinuxカーネル部分とそのライブラリに関してはGNU GPLで保護されており、改変して商用としてリリースする場合は、ソースコードを公開しなければならない。

Gmail、Google Talk、Google Maps等のGoogleアプリケーションの著作権はGoogle社に帰属するため、使用・改変・複製にはGoogle社の許可が必要となる。。

## 4. 本事件の経緯

- 提訴までの経緯
- 審理の経緯
- 訴えの提起
- 本件のAPIの位置づけ
- 第1段階: 著作性の有無
- 第2段階: フェアユース



## 提訴までの経緯

Java		Android
Sun / Oracle		Google
<b>Sun:</b> Java開発開始	1995	
	2003	(Android. Incが設立される)
<b>Sun:</b> GoogleとJavaプラットフォームのライセンス交渉を開始(契約には至らず)	2005	Android. Incを買収 SunとJavaプラットフォームのライセンス交渉を開始(契約には至らず)
<b>Sun:</b> 11月 J2SE(現OpenJDK)、J2MEをOSS化:(GPLv2:コード開示義務あり)	2006	(独自にプラットフォームを開発)
<b>Sun:</b> 5月 OpenJDK (Java APIのOSS)を発表	2007	11月 携帯電話向けのAndroidを発表し、AndroidプラットフォームをOSS化(Apache2.0)
	2008	Androidを搭載した携帯電話の発売開始
<b>Oracle:</b> 1月 Sunを買収 10月 カルフォルニア地裁に特許および著作権侵害の訴えを提起	2010	

## 審理の経緯

2010年	提訴	Oracle → Google GoogleのAndroidシステムによる著作権及び特許侵害
2012年	地裁	特許非侵害、著作物性否定による著作権非侵害 判決
2014年	CAFC	著作物性肯定 37のAPIパッケージと9条の実装コードの著作権侵害。但しフェアユースについては結論を出さず、差し戻した。
2015年	最高裁	裁量上訴不受理
2016年	地裁	フェアユース肯定
2018年	CAFC	フェアユース否定
2021年	最高裁	著作権侵害を認めず、CAFCの判決を破棄、同裁判所に差し戻し 宣言コードの著作物性に関する判断は回避し、 著作物性があると仮定した上で、フェアユース肯定

2014年までAPIの著作物性の有無(第1段階)と2015年からフェアユースの適否(第2段階)が審理された。



## 提訴

➤原告: Oracle America, Inc. 被告: Google, Inc

➤主訴:

2005年Google社が、スマートフォン向けのOS「Android」を開発するにあたり、Oracle社の有するAPIである「Java SE」から、コード約11,500行を複製した。Androidは特許権を侵害し、また、複製行為は著作権を侵害している。

# 第1段階: 著作物性の有無 地裁の判断

## ➤ 主訴

GoogleのAndroid APIが、Oracleが所有するJava APIの特許権及び著作権を侵害している。

## ➤ 地裁の判断

### ➤ 一部を除き、著作権侵害は認められない

- 侵害 : rangeCheck機能(9行のコード)および逆コンパイルされたセキュリティファイル
- 非侵害: 37件のJavaのAPIは、著作権の保護対象にならない。

### ➤ 特許権侵害は認められない…… 確定 (Oracle による控訴なし)

## ➤ 著作権侵害について

- 著作権保護されるのはプログラムコードであり、機能自体は著作権保護の対象とならない。
- 複製されたAPIコードのうち37件のAPI パッケージは機能を示すものである。
- ただし、あくまでこの訴訟で問題となった37件のAPIに限定されたもので、APIというものの全般が著作権保護の対象外というものではない。

# Androidのプラットフォーム



全168パッケージ

## コアライブラリ

複製した37のAPI Javaパッケージ  
論理構造 (SSO、Structure・Selection・Organization) の位置づけ

APIによって規定された規則群の「実際の実装」。

# 判断の対象となったAPIの対象範囲

- 宣言コード Declaring code : 処理を記述するコードを呼び出すコード
- 実装コード Implementing code : 実際の処理方法内容を記述するコード

```
package java.lang;  
public class Math {  
    public static int max (int x, int y) {  
        if (x > y) return x ;  
        else return y ;  
    }  
}
```

```
// Declares package java.lang  
// Declares class Math  
// Declares method max  
// Implementation, returns x or  
// Implementation, returns y  
// Closes method  
// Closes class
```

複製した  
コードの  
対象範囲

# 第1段階: 著作物性の有無 CAFCの判断

## ➤ 主訴

Oracle: APIパッケージの宣言コード(7000行)および37 Java APIパッケージのSSO についての著作権侵害不成立の判断について控訴

Google: rangeCheck機能およびセキュリティファイルに関する著作権侵害成立の判断について控訴

## ➤ CAFCの判断

- 37のJava APIの宣言コードおよびSSOの著作物性を認める。
- rangeCheck機能およびセキュリティファイルに関する著作権侵害成立。

## ➤ 著作物性について

- 著作権法102条に宣言コードおよびSSOを当てはめて判断
- Googleの上告が棄却され、著作物性が是認。
- Googleによる利用がフェアユースか否かの判断をするために、地裁に差戻された。

## 第2段階:フェアユース 双方の主張

		Oracle	Google
第1要素	営利目的の利用	<ul style="list-style-type: none"> <li>AndroidにJavaを利用する事で利益を得た。圧倒的に営利的。</li> </ul>	<ul style="list-style-type: none"> <li>AndroidはOSSの条件に基づき提供されている</li> <li>Googleの収入源にはなっていない</li> <li>よって、営利目的ではない</li> </ul>
	変容的使用	<ul style="list-style-type: none"> <li>「新しい表現、意味またはメッセージ」に代替していない。</li> <li>Java APIを同じ目的で利用している。</li> </ul>	<ul style="list-style-type: none"> <li>AndroidとJavaの目的は違う目的</li> <li>「Androidのポイントは、スマートフォン向けの革新的なプラットフォームを創造すること」</li> </ul>
	悪意の利用	<ul style="list-style-type: none"> <li>Googleは、Androidの開発促進にはJava APIを用いることが必要であり、またJava APIを使用するにはライセンスが必要と認識していた。</li> </ul>	<ul style="list-style-type: none"> <li>宣言コードとSSOは開発者にとって、また、Java言語の独立した実装可用性から、無償で使用および実装できる</li> </ul>
第2要素	著作物の性質	<ul style="list-style-type: none"> <li>宣言コードおよび37のパッケージのAPIの設計は高度に創作的な作業</li> </ul>	<ul style="list-style-type: none"> <li>宣言コードおよびSSOは、「最小のレベルの創造性」</li> </ul>
第3要素	使用の量と実質性	<ul style="list-style-type: none"> <li>170行でよいところを、11,500行複製</li> </ul>	<ul style="list-style-type: none"> <li>全286万行のうちの11,500行複製</li> <li>Java APIを利用することは、Java利用者のためにもなる</li> </ul>
第4要素	市場への影響	<ul style="list-style-type: none"> <li>Java SEにとって、Googleの複製によって生じる現実および潜在的な損害は「圧倒的」</li> </ul>	<ul style="list-style-type: none"> <li>①Oracleは装置メーカーではないこと、②Oracleは独自のスマートフォン向けプラットフォームを構築できていないことから、Java SEとAndroidは同じ市場において競合しない。</li> </ul>

## 第2段階:フェアユース 地裁の判断

		争点	評価
第1要素	営利目的の利用	<ul style="list-style-type: none"> <li>・営利目的</li> <li>・OSSとしての性質は営利目的を和らげる</li> <li>・完全に営利目的の利用であったとしても、フェアユース可能性あり。</li> </ul>	該当:営利目的の利用ではあるが、変容的かつ善意での利用である。
	変容的使用	<ul style="list-style-type: none"> <li>・複製された宣言コードは、同じ機能、目的をもつ。しかし、</li> <li>・選択的利用、スマホの動作環境実現、新たな機能との組み合わせ</li> </ul>	該当
	悪意の利用	<ul style="list-style-type: none"> <li>・Googleは宣言コードおよびSSOは無償利用可、実装コードは使用許諾が必要と認識</li> </ul>	非該当
第2要素	著作物の性質	<ul style="list-style-type: none"> <li>・APIの設計は、高度に創作的な作業であるが、宣言コードは高度に創作的ではない。</li> <li>・宣言コードとSSOは、著作権保護に値するほどに十分に創作的ではあるが、機能的な考慮がその設計を占める。</li> </ul>	有利:複製されたコードは機能的であり、高度に創作的ではない。
第3要素	使用の量と実質性	<ul style="list-style-type: none"> <li>・必要最小限である37のAPIパッケージのみ複製</li> <li>・創作性の高い実装コードとSSOは未複製</li> <li>・複製は、もとの著作物全体の1%</li> <li>・複製の目的はJava開発者の混乱を避けるため</li> </ul>	有利:フェアユースに有利:変容的な複製において必要な範囲内
第4要素	市場への影響	<ul style="list-style-type: none"> <li>・Javaの市場であるPC市場に影響を与えていない。</li> <li>・K」という名称の無償オープンソースとして提供されていた。Androidに利用された37のパッケージおよびSSOとほぼ同じものが、GPL ver.2の条件下において、無償で提供されていた。Androidによる市場への影響は、OpenJDKの提供によって見込まれていたものと同じ</li> </ul>	該当:AndroidによってJavaの市場に影響は与えられていない。
総合評価			Googleによる利用はフェアユースである

## 第2段階:フェアユース CAFCの判断

		争点	評価
第1要素	営利目的の利用	<ul style="list-style-type: none"> <li>・ナップスター事件「『通常は購入させるであろうものを無償で』提供することは、営利利用を構成」</li> <li>・A&amp;Mレコード事件において「直接的な経済的利益は、営利利用の立証に要求されない。」</li> </ul>	該当
	変容的使用	<ul style="list-style-type: none"> <li>・107条柱書に列挙されている利用ではない。(「批評、解説、ニュース報道、教授、研究または調査等」の分類に該当しない。)</li> <li>・目的は同じ</li> <li>・選択的利用のみではフェアユースとは言えない</li> <li>・複製された以外の部分は評価に関係なし</li> <li>・スマホは新しいコンテキストではない</li> </ul>	非該当 営利目的の利用であり、変容的ではない性質である時点で、第1要素は <b>フェアユースに対して不利</b> であるとした。
	悪意の利用	<ul style="list-style-type: none"> <li>・善意であったとしても有利には働かない</li> <li>・Googleが悪意である説得力のある証拠がなかっただけ</li> </ul>	(不明)「悪意」とも判断してはいない。
第2要素	著作物の性質	<ul style="list-style-type: none"> <li>・機能的な考慮は実質的かつ重要</li> <li>・地財のは『類型としてフェアユース全体の判断にそれほど大きな影響を与えない』</li> </ul>	有利 ただし、全体の判断に影響なし
第3要素	使用の量と実質性	<ul style="list-style-type: none"> <li>・Java言語によるソフトウェア開発者に受け入れられるために、開発者に親しまれている著作物の外形と同じように設計することはフェアユースではない。</li> <li>・複製した部分がJava全体の小部分だったとしても、質的には僅かであるとはいえない。</li> </ul>	よくて中立的、むしろ <b>不利に傾く</b>
第4要素	市場への影響	<ul style="list-style-type: none"> <li>・Java SEはスマホにも利用されていた</li> <li>・OracleがJava SEをスマートフォンに対しライセンスしようとしていることに疑いはない。</li> <li>・法律は、著作権者の「潜在的な市場」に参入する権利を保護しているため、この事実をもって市場への影響を立証するに十分</li> </ul>	該当 現実の市場および潜在的市場への影響がある
総合評価		<ul style="list-style-type: none"> <li>・Gは自社開発またライセンスも可能だったが複製を選んだ。</li> <li>・Androidの公開によって、Oracleの著作物を提供するJava SEは事実上置き換えられ、Oracleが開発途上の市場(=スマートフォン市場)に参加することを妨げられた。</li> </ul>	<b>Googleによる利用はフェアユースではない。</b>



## 5. 最高裁審理

- 背景の確認
- 審理の目的
- 論点と判示
- 各論
- 総合考慮
- フェアユースに関する審理のまとめ

## 背景の確認

- ▶ Oracle America社は、良く使われるJavaコンピュータプログラミング言語を利用するコンピュータプラットフォームであるJavaSEの著作権者である。
- ▶ 2005年に、Googleはアンドロイドを取得し、モバイル機器のための新たなソフトウェアプラットフォームを構築しようとした。
- ▶ Googleはおよそ11500行のコードをJavaSEプラットフォームからコピーした。コピーされた行はアプリケーション・プログラミング・インターフェース(API)と呼ばれるツールの部分である。
- ▶ APIはプログラマーにその自身のプログラムにおける利用のために前もって書かれている計算タスクを呼び出す事を可能とするものである。
- ▶ コピーの目的は、Javaプログラミング言語に馴れ親しんだ数百万のプログラマーが新しいアンドロイドプラットフォームで働ける様にするためであった。

# 審理の目的

## ▶ 下級審の検討

- (1)JavaSEの権利者はAPIのコピーされた行について著作権を持ち得るか、
- (2)もしそうであるなら、Googleの複製行為は「フェアユース(公正利用)」となり、Googleを著作権侵害責任から免除するか。

## ▶ 下級審の判断

- 連邦控訴裁判所は、コピーされた行は著作権保護の対象であると判示した。
- 陪審員がGoogleについてフェアユースとした。
- 連邦控訴裁判所はそれを覆し、法律事項として、Googleのコピー行為はフェアユースではないと結論づけた。

## ▶ 最高裁による受理

API(宣言コード)の著作物性とフェアユース成否の両方に関する判断について審理する。

# 論点と判示

## ➤ 論点:

- a. 著作物性の検討
- b. コンピュータプログラムに対する「フェアユース」の法理
- c. フェアユースとした陪審の判断を裁判官が覆すことができるのか
- d. 当てはめによフェアユースの該非

## ➤ 判示:

プログラマーが自身で獲得した才能により新しく変形的なプログラムについて、そのプログラムが正常にはたらく事を可能とするのに必要なものであるコード行のみを含む、GoogleのJavaSE APIのコピー行為は、法律事項として、そのコードのフェアユースである。

## (a) 著作物性の判断と審理対象

- ▶ 著作権はある期間著作物を利用する排他的権利を著作者に付与する事によりその創造を促すものである。この様な排他性はネガティブな結果を引き起こし得るものであるから、議会と裁判所は、著作権者の独占が公益を害さない事を保証するために著作権保護の範囲を限定して来た。
- ▶ 本ケースにおいては、現行の著作権法における2つの限定が関係している。
  - 著作権法第102条(b): 著作権の保護は「いかなるアイデア、手続き、プロセス、システム、実行方法、概念、原理、又は発見・・・」には及ばない。
  - 著作権法第107条: 著作権者は他人が著作物について「フェアユース」をする事を止められない。
- ▶ Googleの訴えは、これらの規定を問題の複製行為に適用する事を裁判所に求めるものである。本ケースを解決するのに必要以上の事を判断しないよう、コピーされた行は著作保護の対象であり得ると仮定し、Googleのこれらの行の利用がフェアユースであるかに集中する。

## (b) コンピュータプログラムに対する「フェアユース」の法理

- ▶ 「フェアユース」の法理は柔軟に技術の変化を考慮に入れる。
- ▶ コンピュータプログラムは常に機能的な目的に使われるものである。従って、他の多くの著作物とはある程度異なる。
- ▶ したがって、フェアユースは、コンピュータプログラムに与えられる独占をその適法な範囲に留める、文脈に基づくチェックを提供する事により、コンピュータプログラムに対して重要な役割を果たす。

## (c) フェアユースは事実問題か、法律問題か

- フェアユースの問題は事実と法律の混合問題である。
- 再審査法廷は、陪審員による前提となる事実認定に適切に従うべきであるが、その事実からフェアユースとなるかの最終的な問題は法的な問題であり、判事が再審理することができる。
- ここでの最終的な問題は法律の問題であり、事実の問題ではないから、このアプローチは、修正第7条の陪審員による判断された事実の法廷による再審査の禁止に反するものではない。
- 「陪審員による審理の権利」は、フェアユースの抗弁を陪審員が解決する権利を含むものではない。

## (d) 当てはめによるフェアユースの成否

- ▶ Googleの限定的なAPIのコピー行為がフェアユースとなるかについて判断するため、著作権法のフェアユース規定に定められた4つの主要要素が検討する。
- ▶ 裁判所は、Campbell事件判決に基づき、ある文脈において他より重要な要素があると考えられる事を認めて来た。
- ▶ 本件において、他より重要と判断されたのは、第2要素(Campbell事件とは異なる)。



## あてはめ：第2の要素

### ➤ 根拠

著作権の種類によっては、著作権による保護の核心に遠近がある。事実的または機能的な著作物では、保護を受けない事実やアイデアの要素が多く、フェアユースの成立に有利にはたらく。

### ➤ 事実

- APIの全体的な構成は著作権保護対象を有しないアイデアである。
- Googleによって独立に書かれたコードのみが、新たな創作的表現である。
- コピーされた行は、著作権保護対象ではないアイデア(一般的なタスク)と、新たな創作的表現(Androidの実装コード)を本質的に結びつけているインターフェース部分である。
- コピーされたコードは、「ユーザインターフェース」の部分のみである。
- コピーされた行の価値はAPIシステムを学んだ利用者(コンピュータプログラマー)の投資に由来するものである。
- ここでのフェアユースの適用は一般的な著作権保護を危うくするものとはならない

### ➤ 評価

問題となる著作物の性質はフェアユース成立に有利である。

# あてはめ：第1の要素

## ▶ 根拠

利用の「目的及び性質」に関する調査は、そのコピー行為が「変形的transformative」、すなわち、その行為は「さらなる他の目的又は異なる性質とともに新しい何かを追加する」かに帰着する（Campbell事件判決）。

## ▶ 事実

- GoogleのAPIの限定的なコピー行為は変形利用である。
- Googleは、プログラマーが異なるコンピューティング環境において慣れ親しんだプログラミング言語の一部分を捨てずに働く事を可能にするのに必要なだけのをコピーした。
- Googleの目的は異なるコンピューティング環境（スマートフォン）のために異なるタスク関連システムを作り出し、その目標を達成して広める事に役立つプラットフォーム—アンドロイドプラットフォーム—を作り出す事にある。
- インターフェースの再実装によりさらなるコンピュータプログラムの開発が可能となる数多くのやり方を示している。

## ▶ 評価

Googleの目的は、著作権自体の憲法上の基本的な目的である創作的進歩と合致する。

## あてはめ：第3の要素

### ➤ 根拠

「本質性」の要素は、コピー行為の量が妥当なかつ変形の目的に結びついている場合には、一般的にフェアユースに有利に働く。

### ➤ 事実

- Googleは、APIから、仮想的に何百かの異なるタスクを呼び出すのに必要な全ての宣言コードに及ぶ約11500行の宣言コードをコピーした。しかし、これらの11500行は、全体で286万行からなる、問題となるAPI全体の0.4%に過ぎない。
- 「利用された部分の量及び本質性」を考慮するにあたり、11500コード行は遥かに大きな全体の小部分と見られるべきである。
- インターフェース部分として、コピーされたコード行は、プログラマーによってアクセスされる他のコード行と分離不能な形で結びついている。Googleがこれらの行をコピーしたのは、その創造性又は美しさのためではなくそれによりプログラマーがその技能を新しいスマートフォンコンピューティング環境に持ち込む事を可能とするためである。

### ➤ 評価

コピー行為の量が妥当なかつ変形の目的に結びついているので、フェアユースに有利に働く。

## あてはめ：第4の要素

### ➤ 根拠

「著作物の潜在的市場又は価値」におけるコピー行為の「影響」に焦点をあてる

### ➤ 事実

- Googleの新しいスマートフォンプラットフォームはJavaSEの代替市場ではない事を示している。
- JavaSEの著作権者はそのインターフェースの他の市場への再実装から利益を受けるであろう事も示している。
- これらの事実に基づき、著作権を行使する事は公衆に対し創造性に関して害をもたらす恐れがある。

### ➤ 評価

市場への影響の判断は、フェアユースに有利に働く。

# 総合考慮

- ▶ コンピュータプログラムが主として機能的なものであるという事実は伝統的な著作権の概念をこの技術的な世界に適用する事を困難にしている。
- ▶ GoogleのAPIのコピー行為は、法律事項として、そのマテリアルのフェアユースとなる。
  - 当該コピー行為は、ユーザインターフェースを再実装するためのもで、Androidプログラマーが自身で獲得した才能により新しく変形的なプログラムについて働く事を可能とするのに必要なもののみを選択しているからである。
- ▶ この結論に至るにあたり、最高裁のフェアユースに関する先例を覆す、あるいは修正する判断はされていない。

# 反対意見

審理において、フェアユース成立に関し、賛成意見(6人)に対し、下記の様な反対意見(2人)があった。

## 反対意見

- コードの区別は恣意的である。著作権法上に区別はなく、宣言コードにも著作権性あり。
- Googleの行動は「スマートフォンやワイヤレス業界におけるOracleの潜在的な市場に「悲惨な影響を与えた」。(Amazonは、Kindle端末にJavaプラットフォームを組み込むためにOracleにライセンス料を支払っていたが、Android登場後、Oracleは97.5%の割引を要求された)。

# フェアユースに関する審理のまとめ

GによるAPI複製に対する要素		地裁	CAFC	最高裁
第1要素	営利目的の利用	Oに有利	Oに有利	Oに有利
	変容的使用	Gに有利	Oに有利	Gに有利
	悪意の利用	Gに有利	明言せず	
第2要素	著作物の性質	Gに有利	Gに有利	Gに有利
第3要素	使用の量と実質性	Gに有利	Oにやや有利	Gに有利
第4要素	市場への影響	Gに有利	Oに有利	Gに有利
フェアユースの成否		成立	非成立	成立

## (参考)現在のJavaライセンス

### オラクルのJDK商用ディストリビューション:

商用サポートと予測可能性による利点を求める場合は、オラクルの商用ディストリビューションのJava SE [JDK](#)または[JRE](#)を使用できます。オラクルの商用JDKリリースは、ほとんどの部分がオープンソース・コードから構築される予定です。ただし、JDKの一部には制約のあるコードが含まれるため、完全に機能するオープンソースの代替コードで置き換えられるまでの間、オラクルは引き続き商用リリースでこのコードを使用します。開発とデプロイのサポート・オプションについて、詳しくは[Java Support](#)サイトを参照してください。



## 出典・参考文献

- 最高裁判例 [https://www.supremecourt.gov/opinions/20pdf/18-956\\_d18f.pdf](https://www.supremecourt.gov/opinions/20pdf/18-956_d18f.pdf)
- SOFTIC 判例ゼミより
  - [Oracle v. Google\(Fed. Cir., 2018.3.27\)](#)
  - [Oracle v. Google CAFC判決\(2014.5.9\)](#)
- Groklaw Net “Google’s Opening Statement”  
<http://www.groklaw.net/pdf3/OraGoogle-Trial-GoogleOpeningStills.pdf>
- SOFTIC外国判例仮訳集より[Campbell v Acuff](#)
- [「分析・検索サービスがフェアユースとされた外国判例」](#) 2016年10月18日 弁護士 石新智規
- [「フェア・ユースの考え方 第一章フェア・ユースの法理」](#) 2010年 弁護士 山本 隆司
- 「ベータマックス事件の概要」 著作権分科会(第23回)議事録・配付資料 [資料4](#) > [参考資料2](#)

EOD