Ｊａｖａ

Ｊａｖａ（ジャバ、ジャヴァ）は、汎用プログラミング言語とソフトウェアプラットフォームの双方を指している総称ブランドである。オラクルおよびその関連会社の登録商標である。１９９６年にサン・マイクロシステムズによって市場リリースされ、２０１０年に同社がオラクルに吸収合併された事によりＪａｖａの版権もそちらに移行した。

プログラミング言語Ｊａｖａは、Ｃ＋＋に類似の構文、クラスベースのオブジェクト指向、マルチスレッド、ガーベジコレクション、コンポーネントベース、分散コンピューティングといった特徴を持ち、平易性重視のプログラム書式による堅牢性と、仮想マシン上での実行によるセキュリティ性およびプラットフォーム非依存性が理念とされている。Ｊａｖａプラットフォームは、Ｊａｖａプログラムの実行環境または、実行環境と開発環境の双方を統合したソフトウェアであり、ビジネスサーバ、モバイル機器、組み込みシステム、スマートカードといった様々なハードウェア環境に対応したソフトウェア形態で提供されている。その中枢技術であるＪａｖａ仮想マシンは各プラットフォーム環境間の違いを吸収しながら、Ｊａｖａプログラムの適切な共通動作を実現する機能を備えている。このテクノロジは「ｗｒｉｔｅ　ｏｎｃｅ，　ｒｕｎ　ａｎｙｗｈｅｒｅ」と標榜されていた。

２０１９年の時点でＧｉｔＨｕｂによると、Ｊａｖａは特にクライアント／サーバモデルのＷｅｂアプリケーションで使用されている最も人気の高いプログラミング言語の１つであり、全世界でおよそ９００万人の開発者がいるとレポートされている。最新バージョンは、２０２２年９月にリリースされたＪａｖａ　１９と、２０１８年９月にリリースされたＪａｖａ　１１の長期サポート（ＬＴＳ）版、２０１４年３月にリリースされたＪａｖａ　８のＬＴＳ版である。オラクルは未解決のセキュリティ問題によるリスクを回避するために、旧バージョンのアンインストールと新バージョンへの移行を強く推奨している。

Ｊａｖａの特徴

現在の正規ベンダーであるオラクルの公式アピールは、以下の通りである。特に業務用システムの構築に最適であるとしている。

Ｊａｖａ　ｒｅｄｕｃｅｓ　ｃｏｓｔｓ，　ｓｈｏｒｔｅｎｓ　ｄｅｖｅｌｏｐｅｒ　ｔ

ｉｍｅｆｒａｍｅｓ，　ｄｒｉｖｅｓ　ｉｎｎｏｖａｔｉｏｎ，　ａｎｄ　ｉｍｐｒｏｖｅｓ　ａｐｐｌｉｃａｔｉｏｎ　ｓｅｒｖｉｃｅｓ　ａｓ　ｔｈｅ　ｐｒｏｇｒａｍｍｉｎｇ　ｌａｎｇｕａｇｅ　ｏｆ　ｃｈｏｉｃｅ　ｆｏｒ　ｅｎｔｅｒｐｒｉｓｅ　ａｒｃｈｉｔｅｃｔｕｒｅ，　ｆｉｎａｎｃｅ，　ａｎｄ　ＨＲ．　Ｊａｖａ　ｉｓ　ｕｓｅｄ　ｉｎ　ｍａｎｙ　ｉｎｄｕｓｔｒｉｅｓ　ｉｎｃｌｕｄｉｎｇ　ｍａｎｕｆａｃｔｕｒｉｎｇ，　ａｕｔｏｍｏｔｉｖｅ，　ｉｎｓｕｒａｎｃｅ，　ａｎｄ　ｐｕｂｌｉｃ　ｓｅｃｔｏｒ．

Ｊａｖａは、コストを削減し、開発者の時間枠を短縮し、イノベーションを促進し、エンタープライズアーキテクチャ、財務、およびＨＲに最適なプログラミング言語としてアプリケーションサービスを改善します。　Ｊａｖａは、製造・自動車・保険・公共部門などの多くの業界で使用されています。

オラクルによると、全世界の３億のコンピュータデバイスでＪａｖａ実行環境が動作しており、全世界の２００万の人員がＪａｖａ開発環境を使用しており、全世界で２５０億枚のＪａｖａ　Ｃａｒｄが発行されている、と統計されている。

Ｊａｖａの構文

Ｊａｖａプログラム構文は、Ｃ＋＋によく似たものである。オブジェクト指向言語の一面が強調されがちだが、Ｃ言語のような手続き型言語としてもプログラミングできる。Ｊａｖａはオブジェクト指向パラダイムをそれほど強制しない。

Ｊａｖａは、同時にマルチパラダイム言語でもある。ＪＤＫ　１．１でＪａｖａＢｅａｎｓ／ＪａｖａＲＭＩ／ＣＯＲＢＡによるコンポーネントプログラミングと、リフレクションＡＰＩによるメタプログラミングが備えられた。Ｊ２ＳＥ　５．０でジェネリクス構文／ＡＰＩによるジェネリックプログラミングが追加された。Ｊａｖａ　ＳＥ　７で並行ＡＰＩによる並行プログラミングが追加された。Ｊａｖａ　ＳＥ　８ではラムダ式／関数型インターフェース／ストリームＡＰＩなどによる関数型プログラミングが追加された。２０１４年（Ｊａｖａ　８）以降の関数型とジェネリクスを多用しているＪａｖａプログラムは、それ以前のＪａｖａプログラムから大きく様変わりしている。

オブジェクト指向

Ｊａｖａは、クラスベースのオブジェクト指向である。クラス、インターフェース、インスタンスといった概念を中心にしたものである。クラスのメンバ要素は、フィールド、メ

ソッド、静的フィールド、静的メソッド、定数、内部クラス、コンストラクタ、ファイナライザである。インターフェースは抽象メソッドと定数で構成される純粋抽象クラスである。クラスはインスタンスのひな型であり、インスタンスはクラスを実体化したものである。Ｊａｖａプログラムは、１個以上のクラス定義文から形成される。Ｊａｖａのクラスはカプセル化、継承、多態性をサポートしている。

カプセル化は、クラスメンバの可視性　（ｐｒｉｖａｔｅ，　ｐａｃｋａｇｅ，　ｐｒｏｔｅｃｔｅｄ，　ｐｕｂｌｉｃ）　でサポートされている。可視性とはメンバのアクセス許可範囲を定めるものであり、ｐｒｉｖａｔｅは同クラス内限定、ｐａｃｋａｇｅは同クラス内と同パッケージ内限定、ｐｒｏｔｅｃｔｅｄは同クラス内と同パッケージ内と派生クラス内限定、ｐｕｂｌｉｃは制限なしを意味する。パッケージはプログラム全体を任意に分割したソースファイルの１個以上のまとまりである。Ｊａｖａのデフォルト可視性は、ファイル単位のｐａｃｋａｇｅなので隠蔽性よりも利便性が重視されている。

継承は、スーパークラスが一つに限られる単一継承をサポートしている。多重継承は不可である。既存クラスに任意メンバを追加した新規クラスを作成できる。Ｊａｖａの全クラスはＯｂｊｅｃｔクラスをルートクラスとしてデフォルト継承する。Ｏｂｊｅｃｔクラスにはロック機能が備えられており、これは並行プログラミングを前提にした仕様である。

多態性は、仮想関数、抽象クラス、インターフェース、動的ダウンキャストでサポートされている。スーパークラスのｖｉｒｔｕａｌメソッドを、サブクラスの同名メソッドでオーバーライドできる機能を仮想関数と言う。スーパークラス変数にサブクラスインスタンスを代入してその変数からサブクラスのメソッドが呼ばれるようにするのは、サブタイピングになる。インターフェースは抽象メソッドだけの純粋抽象クラスであり、任意の数だけクラスに実装できる。実行時ダウンキャストはｉｎｓｔａｎｃｅｏｆ演算子の実行時型チェックが可能で、ダウンキャスト失敗時は例外発生する。

プラットフォーム非依存

プラットフォーム非依存とは、Ｊａｖａプログラムが特定のハードウェアやオペレーティングシステムに依存せずに、あらゆる環境での共通動作を保証する概念である。”Ｗｒｉｔｅ　ｏｎｃｅ，　ｒｕｎ　ａｎｙｗｈｅｒｅ”（一度プログラムを書いてしまえば、どのコンピューターでも動くよ）とされる。Ｊａｖａのプラットフォーム非依存性は次のようにして実現されている。
（中略）
Ｊａｖａコンパイラは、Ｊａｖａソースコードを、Ｊａｖａバイトコードという中間表現

にコンパイルする。Ｊａｖａバイトコードは、Ｊａｖａ仮想マシン用の実行コードである。Ｊａｖａバイトコードは通常、Ｊａｖａクラスファイルにまとめられる。
Ｊａｖａ仮想マシンは、各プラットフォームの差異を吸収するクッション的なソフトウェアである。Ｊａｖａ仮想マシンは、様々なコンピュータ環境対応バージョンが提供されており、各プラットフォームにＪａｖａ実行環境の中核としてインストールされる。
Ｊａｖａ仮想マシンは、任意のＪａｖａクラスファイルをＪａｖａクラスローダーで読み込み、そのＪａｖａバイトコードを解釈実行する。インタプリタ式の解釈走行と、実行時コンパイラで解釈走行させるものがある。
Ｊａｖａ初期のインタプリタ式で走行されるＪａｖａプログラムの実行速度は遅かったが、実行時コンパイラ技術と動的再コンパイル技術　（ｄｙｎａｍｉｃ　ｒｅｃｏｍｐｉｌａｔｉｏｎ）　の導入によって実行速度問題はほぼ解決した。実行時コンパイラとは、一定のＪａｖａバイトコードをまとめてネイティブコードにコンパイルして継続的に実行させる技術である。Ｊａｖａ仮想マシンはメモリ境界とバッファオーバーフローのチェックを行いながらプログラムを走行させる。また、クラスロード時のバイトコード検証機能によって、あからさまなコード暴走や致命的エラーの頻発を事前抑止している。

ガーベジコレクション

Ｊａｖａプログラムのメモリ管理は、Ｊａｖａ仮想マシンのガーベジコレクションによって行われる。ガーベジコレクションとは、すでにどこからも参照されていないインスタンスを自動的に特定して破棄し、その占有メモリ領域を自動的に解放する機能である。人の手によるオブジェクトの生成と破棄を正確に対応させるメモリ管理作業は煩雑化するのが常であり、メモリリークや不正リリースによるエラーを引き起こしやすく、バグの温床と化すのが通例であった。ガーベジコレクションは、Ｊａｖａプログラマを複雑なメモリ管理作業から解放する。

ガーベジコレクタのプロセスは、システムスレッドに乗って未参照のインスタンスを探し続ける。どこかの末端だけが途切れている参照の連鎖のかたまりも正確に特定して、参照の孤島に例えられたメモリ領域を一気に解放する。Ｊａｖａではガーベジコレクション機能に並々ならぬ力が入れられており、その技術更新は現在も進行中である。ガーベジコレクタレクタ、応答時間短縮化のレイテンシ重視ガーベジコレクタ、休止時間短縮化のスループット重視ガーベジコレクタなどが導入されて更に改訂を重ねており、運用環境別の選択使用も可能にされている。

出典: フリー百科事典『Wikipedia』―Java（https://ja.wikipedia.org/wiki/Java）