

# Tomcatを用いたWebアプリケーション開発環境についての考察

## Consideration of the Web application development environment using Tomcat

児玉 満

キーワード：プログラミング、システム開発、Webアプリケーション

### 1. はじめに

1990年にWorld Wide Webが登場し、飛躍的にインターネットの利用が増加したが、当初は静的サイトであり、Webページはビルボード的な役割のものであった。その後CGIの登場で、ユーザーからの入力に応じたりソースを動的に生成することができるようになり、さらにデータベースなどとの連携が可能になったことやセキュリティ面の強化も相まって、昨今言われているWebアプリケーションを構築できるようになった。

WebアプリケーションとはWebの仕組み・機能を使ったインターネット、またはイントラネット上で提供されるアプリケーションのことであり、Webサイトに高度な機能を持たせ、Webを介して利用するサービスを提供するものである。例えば、ブログ、電子掲示板、インターネットバンキング、オンライントレード、ネットショップなどが挙げられる。ただし、Webアプリケーションの明確な定義はなく、動的ページとの差異などは不明瞭であるが機能の多様さという点で単純な動的ページではないと言える。

Webアプリケーションを動作させる方法としては、古くから使われているCGIがあるが、これはWebサーバーからプログラムを実行するためのインターフェイスで、静的なWebサイトから動的なWebアプリケーションを呼び出すことができる。しかしながらこの方法は、プロセスを毎回起動しなければならないため、動作に時間がかかってしまうことが欠点である。そこで、近年ではWebサーバー内のプロセスでWebアプリケーションを実行する方法が用いられるようになり、比較的高速な実行が可能で、サーバーへの負荷も低くなっている。

現在では、スマートフォンアプリの分野において「ネイティブアプリと同等な体験の提供」を目的としたプログレッシブWebアプリと呼ばれる標語に基づいた技術群が精力的に開発されており、クラウドコンピューティングの発展に伴って、自前のWebサーバーではなくフルマネージドクラウドサービスをバックエンドに利用したWebアプリの開発が一部の分野では可能となっている。

Webアプリケーションを開発するためにはHTMLおよびCSSといった静的サイトを構築する言語だけでなく、JavaScriptやPHP、Ruby、Python、Javaなどのプログラミング言語が多く利用されており、それぞれに関連したLaravel、Rails、Django、Springなどといったフレームワークが数多く存在する。

現在人気のプログラミング言語の1つにPythonがあり、これは人間の思考に非常に近く見えるということから、複雑な構造でも簡単にコーディングすることができるため、開発時間を大きく短縮することができる利点があるので、ここで利用する言語として用いても良いと考えられるが、今回は他言語に比べてセキュリティや機能性に優れたJavaを用いたWebアプリについて検討をする。

JavaにはWebアプリケーション開発のためのフレームワークとして、Spring、JSF、GWT、Grailsなどがあり、特にSpringは複雑なWebアプリケーションを高いパフォーマンスで構築できることから、最も強力なフレームワークと言われる。

また、Javaは元々Webサーバー上で動作することも考慮して作られた言語であるため、初めからJava Servletという機能を持っている。これを動作させるためにサーブレットコンテナが必要となるため、その機能を持つTomcatを用いることとする。この基本システムを用いることでどの程度のWebアプリケーションを開発することができるのか、また開発環境としてはどういったものが適切かを考察する。

## 2. 統合開発環境について

統合開発環境（IDE：Integrated Development Environment）とは、プログラミングにおいて、ソースコードを記述するための「エディタ」、ソースコードから実行プログラムを生成するための「コンパイラ」、コードの間違いなどを発見・修正するための「デバッガ」など、様々なプログラム開発ツールをひとまとめにした環境のことを指す。

数多くの種類があるが、どのIDEも現在ではGUI（Graphics User Interface）を備えており、対話的な操作でアプリケーションを開発できる。

また、近年のIDEは、プログラミングの半自動化、グラフィカルな画面の設計、モジュール開発など多くの機能を備えており、アプリケーション開発の高速化に大きく貢献している。

実際に使われているIDEではMicrosoft社のVisual Studio Code、Visual Studioの利用者が多く、次いでオープンソースで開発されているNotepad++や、JetBrains社が開発したIntelliJ IDEA、個人が開発をしているVimなどだが、Visual Studio Code、Notepad++、Vimはどちらかというコードエディタやテキストエディタと言った使われ方が多く、統合開発環境とするためには、様々なユーティリティを手動で統合し、開発者独自で構築する必要があるため、敷居が高いと考えられる。

## 3. 統合開発環境の検討

今回の検証では、Javaを用いることにしているため、Javaが扱えるIDEであることを考慮し候補を挙げ、その中でよく利用されるものを検討対象とする。

候補となるIDEは次の5つで、オープンソースのEclipseプロジェクトを運営する非営利団体であるEclipse Foundationが開発する「Eclipse」、JetBrains社が開発する「IntelliJ IDEA」、Apacheソフトウェア財団が開発する「NetBeans」、Google社が開発する「Android Studio」、Microsoft社が開発する「Visual Studio Code」といったものがある。

それぞれの機能や特徴を次に示す。

### (1) Eclipse

Javaの開発環境が初めから含まれているIDEであり、機能を拡張するためのプラグインも豊富に用意されている。

その中でもPleiades All in OneというプラグインはIDEを日本語化するだけでなく、あらかじめ必要となるプラグインを1つにまとめたものでもあるため、日本語を前提とするならPleiades All in Oneをダウンロードして利用するのがよい。

対応する言語は、Javaだけでなく、C言語、C++、PHP、Pythonも扱える。

画面構成は、コード入力画面を中央に配置し、左右や下部にパッケージ・エクスプローラーやアウトライン、コンソールといった領域がある。(fig3-1参照) また、先に述べたようにJavaの開発環境がすでにEclipseに内蔵されているため、環境構築のためにシステムを操作する必要がない。

また、このIDEは無料で利用できる。



fig.3-1 Eclipseの画面

### (2) IntelliJ IDEA

このIDEは有償版と無料版があり、無料版でもそれなりに開発は行えるが、フロントエンド、SQL、Webフレームワークのサポートが無いため本格的に開発をするには有償版を利用する方が良い。

対応する言語は、Javaはもちろんのこと、Scala、Groovy、Clojure、Kotlinといった比較的最近登場したプログラミング言語が扱える。

画面構成は、初期設定では左右に分かれており、右側でコード入力、左側でプロジェクト管理となっており、実行することで結果を表示するためのコンソールが開くようになっている。(fig 3-2参照)

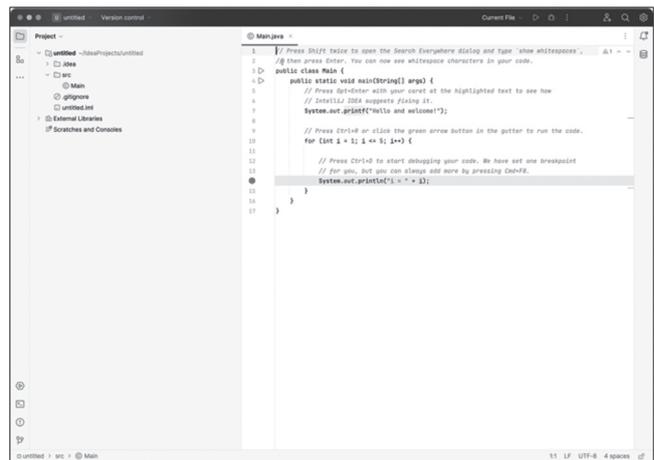


fig.3-2 IntelliJ IDEAの画面

### (3) NetBeans

現在のJavaの開発を手掛けているOracleを中心としたコミュニティにより開発されているため、最新版のJavaにいち早く対応できる。Eclipseと比較されることが多いが、シェア、多機能性、プラグインボリュームなどにおいてEclipseに一日の長がある。

対応言語は、Java、JavaScript、PHP、C言語、C++、Ruby、Python、Groovyと比較的多くに対応している。

画面構成は、初期設定で左右に分かれており、右側にコードを入力する画面、左側でプロジェクトなどの管理が行えるシンプルな構造である。(fig3-3参照)

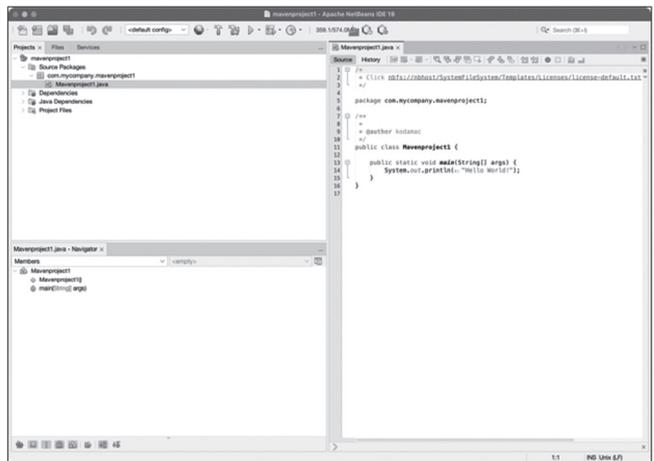


fig.3-3 NetBeansの画面

### (4) Android Studio

Google社が開発し、配布している、Androidアプリ開発用の統合環境である。Androidの開発には現在JavaとKotlinが対応しているため、これら言語を扱うのが主であるが、Kotlinを利用する方が生産性の向上につながると言われている。そのほかの言語としては、C#もあるが、基本的にAndroidアプリ開発がメインであるため、用途は限定される。

Android StudioはIntelliJ IDEAをベースとしたIDEのためそちらに画面構成に近い。(fig3-4参照)

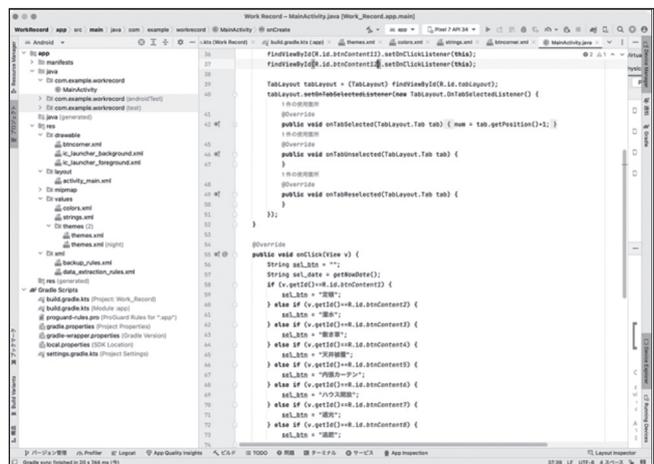


fig.3-4 Android Studioの画面

### (5) Visual Studio Code

Microsoftが開発したオープンソースのコードエディタで、プログラム開発、Web制作、テキスト原稿の執筆など、IDEとしてよりも様々な用途に利用できるエディタである。そういった仕様であるため、初期状態ではテキストエディタ程度にしか利用できないが、手動でプラグインを導入することで、多くのプログラミング言語に対応したIDEにもなり得る。

対応言語は、C#、C/C++、TypeScript、JavaScript、Visual Basic、Go、Java、PHPなど多く、プラグインがあればどのような言語でも対応が可能である。

画面構成はシンプルで、初期状態では右側がコードを入力する画面、左側がファイル进行操作する画面になっている。(fig3-5参照)

ここに挙げたIDE以外のものも多く存在するが、今回は導入が容易かつ多くのプロジェクトで利用されていること、さまざまなプラグインの提供、単体テストや構文解析などが簡単に行えるEclipse (Pleiades All in One) を採用することとした。

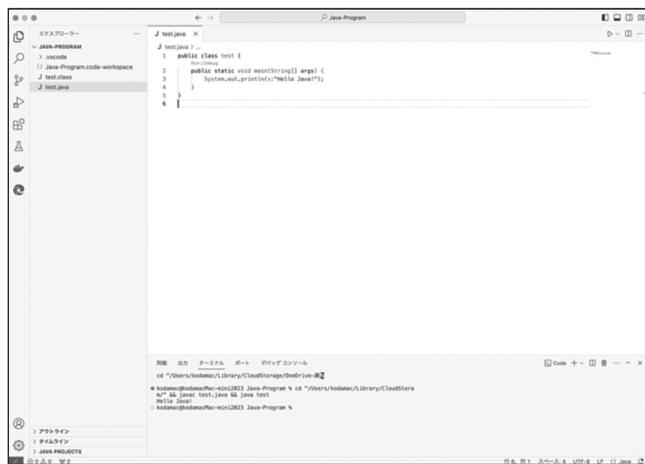


fig.3-5 Visual Studio Codeの画面

## 4. Eclipse (Pleiades All in One) とは

### 4-1. Eclipseについて

Eclipseは1998年にIBM社によって開発された統合開発環境 (IDE) で、主にJavaアプリケーションの開発に使用されており、Eclipse自身もJavaで作られている。2001年にオープンソース化されたが、IBMが制御権を握っているという認識からか、他団体から新たなツールが提供されないという問題を抱えることになり、2004年に開発は非営利団体であるEclipse Foundationに移管することで、その問題が解決され、現在では115以上のメンバー企業、50以上のサブプロジェクトを抱えるオープンソース組織に成長している。

Eclipseには主に次の7つの機能がある。

#### (1) プラグイン

様々な機能をプラグインという形で組み込むことができ、拡張性は非常に高く、Javaの開発環境自体もプラグインとして実装されている。同様の方法で、C++、PHP、Perl、C#、Python、Ruby、JavaScriptなど多様な言語への対応も可能である。

#### (2) バージョン管理システム連携

バージョン管理システムのCSVやGitなどを使ってソースコードの管理が行えるというもの。

#### (3) JUnit連携

JavaのソースコードからJUnitテストコードの自動生成およびテスト実行を行うことができるというもの。

#### (4) Ant連携

ビルドシステムAntと連携ができる。AntはUnix系コマンドのmakeを置き換えるもので、Makefileに相当する各ソースコードの依存関係をXMLで記述できるというもの。

#### (5) リファクタリング

クラス名、メソッド名、変数名等の変更がそれを参照している部分も含め自動的に書き換わるというもの。

#### (6) コード編集支援

クラス名、メソッド名、変数名の補完や自動整形、import文の自動生成、必要なメソッドの自動生成など様々な編集支援機能がある。

#### (7) Eclipse Compiler for Java

Eclipse独自のJavaコンパイラのこと、これがあるためEclipseは別途JDKをインストールする必要なくJavaファイルのコンパイルが可能。

### 4-2. Pleiades All in Oneについて

一般的PleiadesはEclipseを日本語化するためのプラグインと言われているが、Pleiades All in Oneは開発者がOSやプログラミング言語ごとにパッケージを分け、その言語に必要なと思われるプラグインをまとめてくれているものである。

つまり、日本語環境でEclipseを使用したい場合は、PleiadesのサイトからPleiades All in Oneをダウンロードすることで、必要な環境がEclipse込みで得られるというものである。そのため、統合開発環境の構築を考えてもほかのIDEのように様々な設定やプラグインの手動インストールをする必要がなく、Pleiades All in Oneをインストールするだけで構築が完了するという非常に有用なものである。

### 4-3. Tomcatプラグインについて

Pleiades All in OneにはTomcatプラグインが同梱されている。

Tomcatとは、Javaで作成したWebアプリケーションを実行するために利用するためのものであり、無償で利用できるサーブレットコンテナと呼ばれるものである。

サーブレットとは、JavaでWebアプリケーションを構築する際に作成するクラスで、HttpServletを継承したクラスを作成し、Webブラウザからのアクセスに応じた処理を実装していくもので、それをサーバー上で実行する役割を持つものがサーブレットコンテナである。

つまり、Javaでサーブレットを作成し、サーブレットコンテナで実行すると、Webブラウザからのアクセスに応じることができるということである。

そういった機能を持つのがTomcatであるから、それをプラグインとしてEclipseに実装すれば、Eclipse上でWebアプリケーションを作成して実行できるようになる。

以上のことから、統合開発環境にはEclipseを利用するが、本家のeclipse.orgからではなく、実際にはPleiades All in Oneをダウンロードして利用することで開発環境の構築を容易にすることができると考えられる。

5. 開発環境について

5-1. ダウンロードについて

今回は、Pleiades All in OneからTomcatプラグインが同梱されたものを利用するため、それがダウンロードできるMergeDoc ProjectサイトからPleiades All in Oneダウンロードページへ遷移してTomcatプラグインを同梱するUltimateもしくはJavaを選択する。(fig.5-1参照)

ただし、Ultimateはすべてのプラグインが同梱されたものになるため、Javaのみでの開発であれば不要なものも多く、プラグインが多ければその動動作も重くなることから、Javaを選択しておく方が無難である。

Platform	Ultimate	Java	C/C++	PHP	Python
Windows x64 32bit 2018-09 2017	Download	Download	Download	Download	Download
Mac Mac 版について (Qita)	Download	Download	Download	Download	Download
Eclipse 実行用 JDK 17	●	●	●	●	●
IDE Adoptium Bu392, 11.0.21, 17.0.9, 21.0.1	●	●	●	●	●
Lombok 1.18.30	●	●	●	●	●
Lombok 8.5.95, 9.0.82, 10.1.15	●	●	●	●	●
(Winのみ) MinGW 64bit 8.1.0	●	●	●	●	●
(Winのみ) Python 2.7.18, 3.12.0	●	●	●	●	●
(Winのみ) XAMPP x64 (PHP) 8.2.4	●	●	●	●	●
Pleiades 2023.10.02	●	●	●	●	●
Autodetect Encoding 1.8.5	●	●	●	●	●
Platform 4.29.0	●	●	●	●	●
Buildship 3.1.7	●	●	●	●	●
CDT 11.3.0	●	●	●	●	●
EGSE 6.7.0	●	●	●	●	●
GEF 3.17.0	●	●	●	●	●
JDT 3.19	●	●	●	●	●
PDT (PHP) 8.1.0	●	●	●	●	●
Ruby 5.11.0	●	●	●	●	●
Subversive 4.8.0	●	●	●	●	●
WTP SDK - JST	●	●	●	●	●
WTP SDK - WST (改: アイコン改善)	●	●	●	●	●
Wild Web Developer	●	●	●	●	●
Amateras Modeler 2.1.0 改	●	●	●	●	●
AmaterasStepCounter 3.0.4	●	●	●	●	●
AnyEdit Tools 2.7.2	●	●	●	●	●
CheckStyle 10.9.3	●	●	●	●	●
Eclipse Zip Editor 1.1.9	●	●	●	●	●
SootBugs 4.7.3	●	●	●	●	●
Glance 1.2.8	●	●	●	●	●
Enhanced Class Decompiler 3.3.0	●	●	●	●	●
Limy Prop Editor 1.2.5	●	●	●	●	●
NTail 1.3.0	●	●	●	●	●
Open External 0.7.0 改	●	●	●	●	●
PyDev 10.2.1	●	●	●	●	●
QuickUnit 0.7.0	●	●	●	●	●
QuickRex 3.5.0	●	●	●	●	●
Spring Tool Suite (STS) 4.20.1	●	●	●	●	●
SQL Editor 1.4.0	●	●	●	●	●
Thymeleaf 3.0.1	●	●	●	●	●

fig.5-1 ダウンロード画面

5-2. 環境構築について

MergeDoc Projectサイトからダウンロードしたファイルは、それぞれのプラットフォームで実行することでインストールが可能である。ただし、ここではインストール方法については一般的な方法で可能であるため割愛する。

インストールが完了し、Eclipseを起動するとワークスペースをどこに保存するか指示する必要があるが、基本的に初期の操作はそれだけで環境の構築が完了しており、トラブルがなければそのまま起動画面が表示される。(fig.5-2参照)

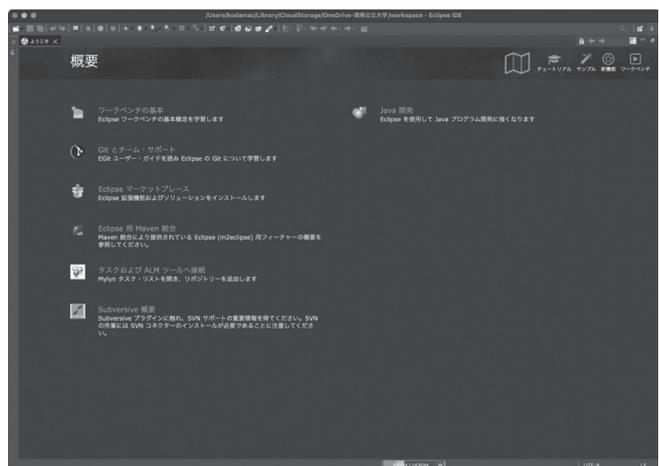


fig.5-2 Eclipseの起動画面

5-3. Webアプリケーションの準備

Eclipseは用途に合わせたプロジェクトを作成し、プログラミングを行う方法であるため、通常のJavaアプリケーションであれば、「Javaプロジェクト」を作成する。今回の目的はWebアプリケーションであるため、「動的Webプロジェクト」を作成する。以降はどのようなプロジェクトであってもクラスを作成し、プログラミングを行うため、作業自体に違いはない。(fig.5-3参照)

また、実行に関しても通常のJavaアプリケーションと同様で「実行」コマンドを用いてプログラムを実行するが、Webアプリケーションの場合は、実行前にあらかじめWebサーバーを起動しておく必要がある。

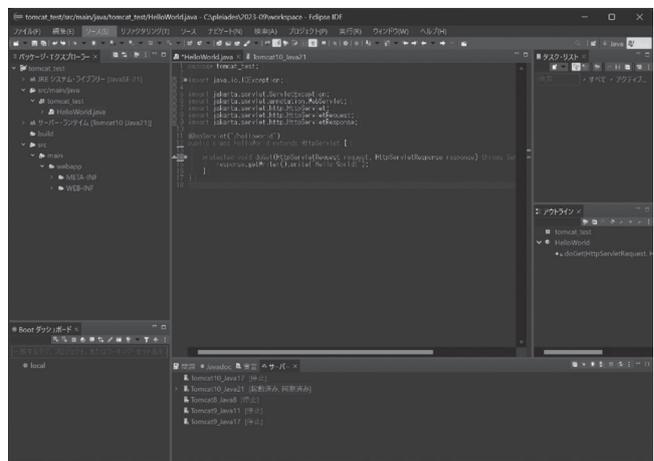


fig.5-3 クラスの作成

ここでは、Tomcatが元々持っているWebサーバーの機能を利用してEclipseでサーバーを起動させておくだけで済むが、本格的にWebアプリケーションサーバーを運用する場合は、サーバー機にApache等のWebサーバーを起動した上でTomcatのサーブレットコンテナを実装するといった作業が必要となるため、サーバー運用の知識が必須である。

今回はあくまで検証であるため、簡易的なEclipseによるサーバーを利用してWebアプリケーションが実際に動作するかを確かめることとする。

#### 5-4. Webアプリケーションの実行

Eclipse上でTomcatによるWebサーバーを起動していれば、Javaによるプログラムを実行するだけで、実行結果がWebブラウザで確認ができる。今回は、「Hello World!」の文字を表示するだけのクラスをJavaで作成し、それを実行した。(fig.5-4参照)

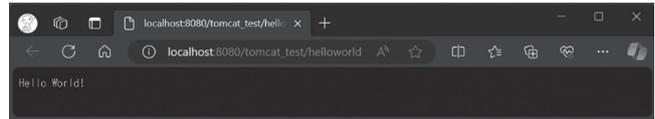


fig.5-4 実行結果

### 6. 考察

Pleiades All in Oneをダウンロードしたものであれば、統合開発環境であるEclipseは、日本語化はもとより、Tomcatプラグインも導入済みであるため、その機能であるサーブレットコンテナが利用できる。そのため、Java Servletの動作確認のためだけにWebサーバーの構築やTomcatを設定してサーブレットコンテナを実装するといった比較的敷居の高い作業をせずに済む点が非常に便利である。また、そういった煩雑な前作業を行わなくて済み、プログラミングに集中できる点も評価できる。

しかしながら、Eclipse自体は不安定さがあり、今回の検証ではプラットフォームとしてMacOSを利用していたが、OSのバージョンが変わった時点でEclipseは起動するものの、ある特定の必要不可欠な操作をするだけで強制終了する不具合が発生した。そのほか、一般的に起動の遅さや動作が重いといった欠点はあるが、Eclipseの継続的な更新やPCの性能向上とともにそういった点も克服されつつあるようである。それらを踏まえてもEclipseはWebアプリケーション開発においてプログラミングに集中できることは非常に有用であり、システム開発のために行う環境構築の負担もほぼ無いことは高評価と考えられる。

### 7. まとめ

1990年にWWWが登場してから、飛躍的にインターネット利用者が増加し、Webサイトもそれに合わせて急増した。当初のWebサイトは静的なものであり、ビルボード的な役割であったが、CGIの登場により、ユーザーからの入力に応じた動的なサイトへと変化することとなる。さらに、データベースとの連携やセキュリティの強化から昨今のWebアプリケーションへと至った。

WebアプリケーションはWebの仕組みや機能を利用したインターネットまたはイントラネット上で提供されるアプリケーションのことであり、Webサイトに高度な動的機能を持たせることで、Webを介したサービスを提供することができる。

その動作方法としては、古くからCGIがあるが、これはWebサーバーからプログラムを実行するためのインターフェイスであり、プロセスを毎回起動するため、動作に時間がかかる欠点がある。そこで、Webサーバー内のプロセスでWebアプリケーションを実行する方法が用いられることで、その欠点を補い、サーバーへの負荷軽減となった。

Webアプリケーションの開発には、HTMLおよびCSSといった静的サイトを構築する言語に加え、JavaScript、PHP、Ruby、Python、Javaといったプログラミング言語が使われており、さらにそれぞれに連携するLaravel、Rails、Django、Springといったフレームワークが数多く存在する。

今回の検証ではJava言語を用いてWebアプリの開発を試みることにした。

Javaは元々Webサーバー上で動作することも考慮して作られた言語でもあり、Java Servletというサー

ブレットコンテナの機能を持っている。このサーブレットコンテナの機能はTomcatというプログラムが持っており、これを基本システムとしてWebアプリケーションを開発することがどの程度可能であるのか、また開発環境としてはどういったものが適切かについて考察を行うこととした。

プログラミング言語の開発環境には様々なものがあるが、Javaを扱えるものでは、特にEclipseが一般的である。

また、Eclipseには日本語化するためのプラグインと一般的に言われているPleiadesがあり、その作者が提供するPleiades All in OneにはTomcat等の必要と思われるプラグインを一まとまりにしたものがあるため、それを利用することで、より簡単に開発環境を構築することが可能である。

実際にWebアプリケーションのテストとして簡易的なプログラムを組み、Eclipse上で実行すると、Webブラウザを起動し、localhostとしてWebサーバーが動作することでサーブレットコンテナを実行することができた。

それに際しては、特に複雑な操作は必要なかったため、結果的にEclipse単体で開発環境が整い、システムに余計なものをインストールする手間がかからず、開発自体もプログラミングに集中することが可能であることや、動作速度も実用的であることなどがわかった。

このことから、Webアプリケーションを開発するための環境としてEclipseはかなり優秀であることがわかったため、今後は、データベースの連携や本番仕様のWebサーバーでの検証を踏まえた上で、何かしら有用なWebアプリケーションの開発を行うこととする。

## 8. 参考文献

総務省『令和5年版「情報通信白書」』<https://www.soumu.go.jp/johotsusintokei/whitepaper/r05.html>

総務省『令和4年版「情報通信白書」』<https://www.soumu.go.jp/johotsusintokei/whitepaper/r04.html>

文部科学省『Society 5.0に向けた人材育成の推進』[https://www.mext.go.jp/component/a\\_menu/other/detail/\\_icsFiles/fieldfile/2018/06/06/1405844\\_002.pdf](https://www.mext.go.jp/component/a_menu/other/detail/_icsFiles/fieldfile/2018/06/06/1405844_002.pdf)

児玉満『スクレイピング技術を用いた学務システムへの出欠入力の効率化』周南公立大学総合研究所『紀要』第1号（令和5年3月）

児玉満『webサイト構築の現状と効率化』徳山大学総合研究所『紀要』第42号（令和2年3月）

児玉満『プログラミング教育に関する考察』西日本短期大学総合学術研究論集第8号（平成30年3月）