Eclipse を使って「動的 Web プロジェクト」としての

Tomcat プログラムの作り方

OD 教室の Eclipse はどのパソコンを利用しても、同一の環境が再現できるように、ホームドライブに Eclipse の設定を保存している為、OD 教室では一般の PC 環境と違う仕組みとなります。 Eclipse の「動的 Web プロジェクト」としての Tomcat プログラムの作り方を説明します。

1 動的 Web プロジェクト

- 1.1 Tomcat サーバ用プロジェクトの作成
- 1.2 動的 Web プロジェクトの作成
- 1.3 動的 Web プロジェクトの設定
- 1.4 プログラムの作成
- 1.5 プログラムの実行

1 動的 Web プロジェクト

1.1 Tomcat サーバ用プロジェクトの作成

① Eclipseを起動し、メニューバーの[ファイル]⇒[新規]⇒[その他]とクリックします。



② [サーバー]⇒[サーバー]と選択し、[次へ]をクリックします。



③ [Tomcat v10.0サーバー]を選択し、[追加]をクリックします。

新規サーバーの定義			
作成するサーバーのタイプを追	盟択します		
サーバーのタイプを選択(S):			
フィルター入力			
🗄 Tomcat v7.0 サ	-/(-		^
🗄 Tomcat v8.0 サ	-/(-		
🗄 Tomcat v8.5 サ	-/(-		
B T I I I I I I I I I I I I I I I I I I			
lomcat v10.0	サーバー		
> Ca OW2			~
IZEE Java EE Jakarta EE)	いし ゴロジェクレレモ パ 株式たり カレ	Toment #= 10-1	- // 881
++	web ノロシェクトと リーハー 催成をローカ	pionical 9 // i	こ公開し(実行し
ます。	web フロジェットとリーハー備成をローカ)	v ionicat y 7()	こ公開し(実行し
ます。 サーバーのホスト名(H):	web ノロシェクトと リーハー 構成 をローカバ localhost	violicat 9 7()	2公開し(実行し
ます。 サーバーのホスト名(H): ⁽	Web フロシェクトとサーバー備成をロール)	Pionical 9 77 1	
サーバーのホスト名(H): ⁽ サーバー名(M):	view クロシェアトとサーバー備成をロール) localhost localhostのTomcat10(Java16)	violicat 9 / 1	
オーバーのホスト名(H): サーバー名(M): サーバー・ランタイム環境(R):	Web クロジェクトと 9 一八一番 80 世 一 JJJ ⁹ localhost localhost の Tomcat10 (Java16) Tomcat10 (Java16)		
サーパーのホスト名(H): サーパー名(M): サーパー名(R):	veb クロウエクトビウーバー確成をロールパ ⁹ localhost localhostのTomcat10 (Java16) Tomcat10 (Java16)	-1/4	ご公開して美行し <u> 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 」 </u>
サーバーのホスト名(H): サーバーる(M): サーバー・ランタイム環境(R):	Web クロウエクトビッーパー確認をロールパ ⁹ localhost localhost の Tomcat10 (Java16) Tomcat10 (Java16)	<u>529</u>	ご公開しく美行し <u> 通加</u> <u> 「 通加</u>
サーバーのホスト名(H): サーバー名(M): サーバーそ(M): サーバー・ランタイム環境(R):	veb クロウエクトビウーバー確認をロール/ localhost localhostのTomcat10 (Java16) Tomcat10 (Java16)	<u>529</u>	ご公開しく実けし → <u>追加</u> <u>く</u> <u>追加</u>
サーバーのホスト名(H): サーバー名(M): サーバーそ(M): サーバー・ランタイム環境(R):	veb クロウエクトビウーバー確認をロールパ localhost localhostのTomcat10 (Java16) Tomcat10 (Java16)	<u>578</u>	ご公開しく実行し → 追加 <u>1</u> 」 <u>1</u> 」 <u>1</u> 」 一 一 一 一 一 一 一 一 一 一 一 一 一
ます。 サーバーのホスト名(H): サーバーそ(M): サーバー・ランタイム環境(R):	Veb クロウエクトとサーバー確認をロール/ localhost localhostの Tomcat10 (Java16) Tomcat10 (Java16)	<u>574</u>	C公開しく実付し ○ 追加 ○ 追加

④ [名前]と[Tomcatインストールディレクトリー]を、以下の値を設定し、[完了]をクリックします。 名前

: 任意のランタイム名 (後手順で使用します)

Tomcatインストール・ディレクトリー	:	C:¥App¥apache-tomcat
----------------------	---	----------------------

mcat サー/「- ソストール・ディレクトリーを指定してください 部(加): ypache Tomcat v10.0 mcat インストール・ディレクトリー(D): ※App¥apache-tomcat ※App¥apache-tomcat apache-tomcat-10.0.6 ダウンロードしてインストール IE(D): Dークペンチ・デフォルト JRE ノンストール(済みの JRE(D)	🎒 新規サーバー・ラン	タイム環境					×
ンストール・ディレクトリーを指定してください 前(<u>M</u>): ypache Tomcat v10.0 mmcat インストール・ディレクトリー(<u>D</u>): ※App¥apache-tomcat ※App¥apache-tomcat apache-tomcat-100.6 ダウンロードしてインストール E(<u>D</u>): D-クベンチ・デフォルト JRE インストール済みの JRE(<u>D</u>)	Tomcat サーバー						=
i前(<u>M</u>): Apache Tomcat v10.0 xmcat インストール・デイレクトリー(<u>D</u>): WApp¥apache-tomcat apache-tomcat-100.6 ダウンロードしてインストール E(<u>J</u>): 7-クベンチ・デフォルト JRE ✓ インストール済みの JRE(<u>J</u>)	インストール・ディレクト	リーを指定してください	3				
Apache Tomcat v10.0 omcat インストール・ディレクトリー(D): 2¥App¥apache-tomcat apache-tomcat-10.0.6 ダウンロードしてインストール IE(D): - クベンチ・デフォルト JRE	名前(<u>M</u>):						
omcat インストール・ディレクトリー(D): E¥App¥apache-tomcat apache-tomcat-10.0.6 ダウンロードしてインストール IE(j): フークベンチ・デフォルト JRE インストール済みの JRE(j)	Apache Tomcat v10	0.0					
2¥App¥apache-tomcat 参照(B) apache-tomcat-10.0.6 ダウンロードしてインストール IE[]): - クベンチ・デフォルト JRE インストール 済みの JRE(D)	Tomcat インストール・ラ	ディレクトリー(<u>D</u>):					
apache-tomcat-10.0.6 ダウンロードしてインストール YE(j): コークベンチ・デフォルト JRE インストール済みの JRE(j)	C:¥App¥apache-to	mcat			参	照(<u>R</u>)	
KE[]): 7ークベンチ・デフォルト JRE インストール済みの JRE[])			apache	tomcat-10.0.6	ダウンロードし	レインスト	-Jl
7ークベンチ・デフォルト JRE	JRE(J):						
	ワークベンチ・デフォルト	JRE		~	インストール	済みの JR	E(])
	JRE(j): ワークベンチ・デフォルト	JRE		~	インストール	済みの JR	E(

⑤ [完了]をクリックします。

					Ц	×
新規サーバーの定義					4.0	
作成するサーバーのタイプを注	選択します					
サーバーのタイプを選択(S):						
フィルター入力						
🗄 Tomcat v9.0 번	-/(-					^
Tomcat v10.0	サーバー					
> 🗁 IBM						
> Cracle						
> 🗁 Red Hat						
> 🗁 Red Hat JBoss Mi	ddleware					
	dureware					Y
J2EE、Java EE、Jakarta EE ます。	Web プロジェクト	とサーバー構成を	ローカル Tomcat	サーバーに	公開して	実行し
J2EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H):	Web プロジェクト	とサーバー構成を	ローカル Tomcat	サーバーに	公開して	実行し
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバー名(M):	Web プロジェクト localhost localhost の Te	とサーバー構成を omcat10 (Java	ローカル Tomcat 16)	サーバーに	公開して! 	 実行し
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバー名(M): サーバー・ランタイム環境(R):	Web プロジェクト localhost localhostのTa	とサーバー構成を omcat10 (Java at v10,0	ローカル Tomcat 16)	サーバーに		実行し <
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバー名(M): サーバー・ランタイム環境(R):	Web プロジェクト localhost localhostのTa Apache Tomc	とサーバー構成を omcat10 (Java at v10.0	ローカル Tomcat	サーバーに <u>ランタイ</u>	公開して	 実行し 追加. 構成.
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバーそ(M): サーバー・ランタイム環境(R):	Web プロジェクト localhost localhostのTr Apache Tomc	とサーバー構成を omcat10 (Java at v10.0	ローカル Tomcat 16)	ש-א(-וכ <u>דעקר</u>	公開して 、 、 、 、	ま行し 追加. 構成.
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバーそ(M): サーバー・ランタイム環境(R):	Web プロジェクト localhost localhost の Tr Apache Tomc	とサーバー構成を omcat10 (Java at v10.0	ローカル Tomcat (6)	9-1(-10 <u>32/94</u>	公開して	ま行し 追加. 構成.
12EE、Java EE、Jakarta EE ます。 サーバーのホスト名(H): サーバー名(M): サーバー・ランタイム環境(R):	Web プロジェクト localhost localhost の Tr Apache Tomc	とサーバー構成を omcat10 (Java at v10.0	ローカル Tomcat (6)	サーバーに ランタイ	公開して 	実行し 追加. 構成.

1.2 動的 We プロジェクトの作成

① Eclipseのメニューバーの[ファイル]⇒[新規]⇒[その他]とクリックします。



② [web]⇒[動的Webプロジェクト]を選択し、[次へ]をクリックします。

1917-トを選択	
フィザードを選択 動的 Web プロジェクトを作成します	Ď
ウィザード(W):	
フィルター入力	
 >	
 浴 封約 Web プロジェクト ご 動的 Web プロジェクト 	

③ [プロジェクト名]に、任意のプロジェクト名を設定します。

また、[ターゲット・ランタイム]は、前手順で作成したTomcatの設定のランタイム名を指定します。 ※特に変更せず進めている場合は、[Apache Tomcat v10.0]となるはずです。 その後、[次へ]をクリックします。

fox-bl dan		
フロシェクトの場所 J デフォルト・ロケーションを使用(D)		
ロケーション(L): U:¥Workspace¥example		参照(S)
ターゲット・ランタイム(U)		
Apache Tomcat v10.0	× #	i規ランタイム(R).
動的 web モジュール パージョン(V)		
5.0		
構成(C)		
Apache Tomcat v10.0 デフォルト構成	~	変更(I)
Apache Tomcat v10.0 ランタイムを操作するための良い出 追加するために追加ファセットをインストールすることができま	発点です。後で新しい機能 す。	能をプロジェクトに
EAR メンバーシップ		
EAR にプロジェクトを追加(A)		
EAR プロジェクト名(E): EAR	~ 新	見プロジェクト(P).
ワーキング・セット		
□ ワーキング・セットにプロジェクトを追加(T)		新規(W)
ワーキング・セット(0):	~	選択(E)

④ [次へ]をクリックします。



⑤ [web.xmlデプロイメント記述しの生成]のチェックがONになっていることを確認し、[完了]をクリックしま

9	ο	

新規動的 Web プロジ:	፤//				×
Web モジュール Web モジュール設定を構成	にます。			T	9
コンテキスト・ルート(R):	example				
コンテンツ・ディレクトリー(D):	src/main/web	арр			
☑ web.xml デプロイメント	記述子の生成(G))			
?	< 戻る(B)	次へ(N) >	完了(F)	キャンセ	2JL

⑥ プロジェクトが作成されます。

U:¥Workspace - Eclipse IDE		– 🗆 X
ファイル(F) 編集(E) リファクタリング ナビゲート	-(N) 検索(A) プロジェクト(P) Tomcat(T) 実行((R) ウィンドウ(W) ヘルプ(H)
 	V Tomcat9_Java16 V	🔅 📑 🕶 🔛 💿 🖙 🐵 🐙 🍓
* • O • Q • I • • • • • •	• @ @ @ & • & @ @ &	
		Q : R B lava R C/C++ S PHP P Python S Per & Ruby
は パッケー (ト ブロジェ S - 日	P 701-547 23 59 8 7 0	
> ₩ example > ₩ Servers	アウトラインを提供するアクティブなエディターがあり ません。	
	271µ9-λ⊅	
	★ たテゴリーなし ▲ 「Tomcat10_Java16 「Tomcat5_Java6 「Tomcat7_Java7 「Tomcat7_Java7	
🕄 ע-ח- 🛙	7 8 8 - 0	
エラー: 2、警告: 0、その他: 0		
設明 ^ > ❷ XML 問題 (2項目)	ע ארעע ארא	
2 avample	77764 / 190664	
go example	/2/M / 1896M	101-8 LF

1.3 動的 We プロジェクトの設定

① 作成されたプロジェクトを右クリックし、[プロパティー]を選択します。

1523	DCEX(A)	>				_	
💭 U:¥Workspi	ローカル・ヒストリーからの復元(Y)				-		×
ファイル(F) 編集	W 1 70770 5	T) 実	行(R) ウィンドウ(W) ヘルプ(H)				
	Windows エクスノローラー		× 🏟 i 📑 • 🖩 🖷 🖻 🙆 🛸 i	o 🗶 😹 📓			
	3774-2027	ni _	Sieman Augusta				
***	検索(A)	Ctrl+H					
	プロファイリング・ツール	>	Q : 😰 🛛 数 Java 暄 C	/C++ 🐻 PHP 💣 Pytho	n 🦙 Pe	erl 🧇	Ruby
1 パッケー 👔 🚿	Perl ネーチャーの追加	2 0][1	
Q	カバレッジ(V)	> - <i>i</i> /a9					
> 🛃 example 🛈	実行(R)	>					
> 🤂 Servers 🛛 🎋	デバッグ(D)	>					
	プロファイJル(P)	>					
	Quick JUnit(Q)	>					
	Checkstyle	>					
	PyDev	, P 🗆					
	ステップカウンタ	> 7 8					
	Java EE ツール	>					
	<i>∓−</i> /₄(E)	> ^					
	構成	>					
	ソース(S)	>					
🕄 र-ग्र- 🖾 🥖	Thymeleaf	> = =					
エラー: 2、警告:(🔽	検証						
説明	プロパティー(R)	Alt+Enter					
→ ⑧ XML 間起	NEL .	10					
<		>					
避 example		542M / 1889M		UTF-8			LF
2							
and the second						_	

 を側のエリアで[Javaのビルド・パス]を選択し、[ライブラリー]タブを開きます。

その後、[モジュールパス]を選択した状態で、[外部JARの追加]ボタンをクリックします。

🔘 example のプロパティー		— 🗆 X
フィルター入力	Java のビルド・パス	< ⇒ <
 > リソース へ AnyEdit ツール Checkstyle Javadoc ロケーション > Java エディター > Java エディター > Java エディター > Java コード・スタイル > Lava アレバズミュー Java のビルド・パス - SpotBugs Spring Tomcat Web Content の設定値 Web パージ・エディター WikiText > XDoclet カパレッジ > クライアントサイド JavaScrip サーバー サービス・ポリシー 	 (● ソース(S) (● ブロジェクト(f) ■ ライブラリー(L) ◇_① 順序およびエクスポート ◇ ドンストレイス > ■ FE・シストレイス > ■ FE・シストレイス > ■ FE・シストレイス > ■ EE マスプラリー > ■ Server Runtime [Apache Tomcat v10.0] > ■ Web App ライブラリー 	 ・(0) ・ ・(0) ・ ・ ・
AF6(C:100-2)		適用(A)
?		適用して閉じる キャンセル

③ ファイル選択画面で、以下のフォルダーの中の、

C:¥App¥apache-tomcat¥lib

以下のファイルを選択し、[開く]をクリックします。

JAR の選択					×
← → ~ ↑ <mark> </mark> « Ap	p > apache-tomcat > lib	ٽ ×	,⊂ lib⊘	検索	
整理 ▼ 新しいフォルダー				EE 👻 🛄	?
^	名前	更新日	時	種類	^
20 + Tai-AL	🔯 jakartaee-migration-1.0.0-shaded.jar	2021/0	7/30 10:51	JAR ファイル	
30 A 7 919 P	🔯 jasper.jar	2021/0	7/30 10:51	JAR ファイル	
💼 U:¥	🔯 jasper-el.jar	2021/0	7/30 10:51	JAR ファイル	
◆ ダウンロード	😰 jaspic-api.jar	2021/0	7/30 10:51	JAR ファイル	
三二 デスクトップ	🔯 jsp-api.jar	2021/0	7/30 10:51	JAR ファイル	
📰 ピクチャ	mvsql-connector-java-8.0.26.jar	2021/0	6/08 16:39	JAR ファイル	_
📔 ビデオ	🖾 servlet-api.jar	2021/0	7/30 10:51	JAR ファイル	
♪ ミュージック	tomcat-api.jar	2021/0	7/30 10:51	JAR ファイル	
ローカルディスク (C)	🔯 tomcat-coyote.jar	2021/0	7/30 10:51	JAR ファイル	
	🔯 tomcat-dbcp.jar	2021/0	7/30 10:51	JAR ファイル	
	🔛 tomcat-i18n-cs.jar	2021/0	7/30 10:51	JAR ファイル	
	🔛 tomcat-i18n-de.jar	2021/0	7/30 10:51	JAR ファイル	~
n 提出 (¥¥ODSVCF ✓	<				>
ファイル	名(N): servlet-api.jar		*.jar;*.zip		~

④ 前手順で選択したファイルが追加されていることを確認し、[適用して閉じる]をクリックします。

example のプロパティー		– 🗆 X
イルター入力	Java のビルド・パス	< ⇒ < <
ソリース ヘ AnyEdit ツール へ Checkstyle Javadoc ロケーション Java ゴディター Java コド・スタイル Java コド・スタイル Java コンパイラー Java ロンパイラー Java ロンパイラー Java ロンパイラー Java のビルド・パス JSP フラグメント 名前空間 SpotBugs Spring Tomcat Web Content の設定値 Web プロジェクトの設定 WikiText > XDoclet カパ(レッジ クライアントサイド JavaScript サーバー	Java のビルド・パス ● ソース(S) (ご プロジェクト(P) ■ ライブラリー(L) % 順序およびエクスポート(O) ④ ビルド・パス上の JAR およびクラス・フォルダー(T): ● servlet-apijar - C!¥App¥apache-tomcat¥lib > 書 RE ライブラリー > ■ Server Runtime [Apache Tomcat v10.0] > ■ Web App ライブラリー	◆ ・ マ・ ・ 8 ● モジュール依存関係(M) JAR の追加(X) 英数の追加(X) ライブラリーの追加(X) ライブラリーの追加(X) クラス・フォルダーの追加(C) 外部クラス・フォルダーの追加(C) 操部クラス・フォルダーの追加(C) 操部クラス・フォルダーの追加(C) 福葉(E) 隆去(R) JAR ファイルのマイグレーション(M)
サービス・ポリシー ターゲット・ランタイム >		適用(A)
9-595-59976 ×		適用し

⑤ 次に、作成したプロジェクトを右クリックし、[新規]⇒[サーブレット]とクリックします。

● ● 東行(N) ※ 目 Tomcat9 Java16		c風
		(1992)
	Q : 암 郞 Java ቬ C/C++ 돐 PHF	🤉 🥔 Python 🐪 Perl 🥔 🕻
אר- 🔁 דייי צע 🗧 🗖 🗄 אייי		-
「「「「」」」 「 「」 「」 「」 「」 「」 「」 「」 「」 「」 「 「」 「」 「」 「」 「」 「」 「」 「」 「」 「 「」 「」 「」 「 「」 「」 「」 「 「」 「 「 「」 「 「	ブなエディターがあり	
zexam 新規(N)) プロジェクト(R)	
中にジャンフ(1)	SQL Jr/h	
表示(W) Alt+Shift	:+W> 📑 ファイル	
III 32-	ゴオルダー	
脂肪 貼 り 付 け	Fortran Project	
※ 削除(D) 終動(\))	エンタープライズ・アプリケーション・プロジェクト	
名前変更(M)	びる 動的 Web フロジェクト	
シ コンテキストから除去 Ctrl+Alt+Shift+De	own におう 同期プロジェクト	
インポート(1)	、 GS ファイル 、 UTAN フー (川)	
-カ- Σ エクスポート(X)	→ ☆ HIML J71ル → ISP ファイル	
2、譬在 SpotBugs	> gara Script ファイル	
 ミリフレッシュ(F) XMLI プロジェクトを用いる(S) 	F5 受 サーブレット	
クラス/ 関連のないプロジェクトを閉じる(U)	19 J1119-	
比較(A)	[∞] ³ ³ ³ ³ ³ ³	
ローカル・ヒストリーからの復元(Y)		
Windows T777D-5-	📑 その他(O) Ctrl+N	

⑥ [クラス名]に、任意のクラス名を入力し、[完了]をクリックします。

💭 サーブレット作成		o x
サーブレット作成 クラス・ファイルの宛务	をを指定します。	S
プロジェクト(<u>P</u>):	example ~	
ソース・フォルダー(<u>D</u>):	¥example¥src¥main¥java	参照(<u>O</u>)
Java パッケージ(<u>K</u>):		参照(<u>W</u>)
クラス名(<u>M</u>):	HelloWorld	
スーパークラス(<u>S</u>):	javax.servlet.http.HttpServlet	参照(<u>E</u>)
□ 既存サーブレット・	クラスまたは JSP を使用(<u>U</u>)	
クラス名(<u>M</u>):	HelloWorld	参照(W)
?	< 戻る(B) 次へ(N) > 完了(F) :	キャンセル

1.4 プログラムの作成

① 作成されたHelloWorld.javaファイルへ、コードを記述します。

※以下はサンプルです。実際に作成する場合は、各自でコードを記述してください。

	HelloWorld.java ※サンプル
1	import java.io.IOException;
2	import java.io.PrintWriter;
3	import jakarta.servlet.ServletException;
4	import jakarta.servlet.http.HttpServlet;
5	import jakarta.servlet.http.HttpServletRequest;
6	import jakarta.servlet.http.HttpServletResponse;
7	public class HelloWorld extends HttpServlet {
8	public HelloWorld() {}
10	protected void doGet(HttpServletRequest request,
11	HttpServletResponse response)
12	throws ServletException, IOException {
13	PrintWriter out = response.getWriter();
14	out.println(
15	" <html><body>Hello World!</body></html> ");
16	}
17	}

② 必要に応じて、web.xmlファイルも更新します。

※以下はサンプルです。実際に作成する場合は、各自でコードを記述してください。

	HelloWorld.java ※サンプル
1 2 3 4	xml version="1.0" encoding="UTF-8"? <web-app <br="" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemal ocation="http://xmlns icp.org/xml/ns/javaee</web-app>
5	http://xmlns.jcp.org/xml/ns/javaee/web-app 4 0.xsd" id="WebApp ID"
6	version="4.0">
7	<display-name>Example02</display-name>
8	<welcome-file-list></welcome-file-list>
10	<welcome-file>index.html</welcome-file>
11	<welcome-file>index.htm</welcome-file>
12	<welcome-file>index.jsp</welcome-file>
13	<welcome-file>default.html</welcome-file>
14	<welcome-file>default.htm</welcome-file>
15	<welcome-file>default.jsp</welcome-file>
16	
17	<servlet></servlet>
18	<servlet-name>Hello</servlet-name>
19	<servlet-class>HelloWorld</servlet-class>
20	
21	<servlet-mapping></servlet-mapping>
22	<servlet-name>Hello</servlet-name>
23	<url-pattern>/</url-pattern>
24	
25	

③ プログラムのコード完成後、メニューバーの[ファイル]⇒[すべて保存]で、保存します。



1.5 プログラムの実行

- ① 作成したプロジェクトを選択した状態で、メニューバーの[実行]⇒[実行]⇒[サーバーで実行]とクリックしま
 - す。



② [localhostのTomcat v10.0サーバー]を選択し、[完了]をクリックします。

使用するサーバーの選択 サーバーの選択方法 ④ 既存のサーバーを選択(X) ○ 手動で新規サーバーを選択(X) ○ 手動で新規サーバーを定義(M) 使用するサーバーの選択: フィルター入力 サーバー ▼ Cackboxt ↓ Icoskboxt ↓ Icoskboxt	状態 間。停止 間。停止	
サーバーの選択方法 ● 既存のサーバーを選択(X) ○ 手動で新規サーバーを定義(M) 使用するサーバーの選択: フィルター入力 サーバー マールター入力 10calhost の Tomcat v10.0 サーバー interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16 interatio_Java16	状態 ◎ 停止 ◎ 停止	
 ● 既存のサーバーを選択(X) ○ 手動で新規サーバーの選択: フイルター入力 サーバー マイルター入力 サーバー * Coalhost の Tomcat v10.0 サーバー Tomcat U_Java to Tomcat U_Java to<td>状態 四一停止 四一停止</td><td></td>	状態 四一停止 四一停止	
 ○ 手動で新規サーバーを定義(M) 使用するサーバーの選択: フイルター入力 サーバー マーレーク ・ Concention ・ Concentio ・ Concention ・ Concention ・ Concention ・ Concention ・ Co	状態 品 停止 品 停止	
使用するサーバーの選択: フィルター入力 サーバー ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	状態 品 停止 品 停止	
フィルクー入力 サーパー ● localhost () Tomcat v10.0 サーパー ■ Tomcat () Java 6 ■ Tomcat () Java 6 ■ Tomcat () Java 6 ■ Tomcat () Java 6	状態 日 停止 日 停止	Í
#−/(- V localhost iocalhost iomcat 0,3va6 Tomcat 0,3va6 Tomcat 2,3va6	状態 日 停止 日 停止	
localhost localhost () Tomcat v10.0 サー/(- Tomcat U_Java to Tomcat U_Java to Tomcat Java 5 Tomcat Java 5	品 停止 品 停止	
localhost () Tomcat v10.0 +/(- iomcatio_avaite Tomcatio_avaite	☐ 停止 □ 停止	
i iomcatio_Java16 Tomcat6_Java6	☐ 停止	
Tomcat6_Java6		
Tomcat7 Java7	福 停止	
loncat_Java/	福 停止	
Iomcat8_Java8	□□ 伊止	- 1
Iomcat9_Java11 Tempat0_lava16	日信に	3
Apache Tomcat v 10.0 は、J2EE 1.2, 1.3, 1.4, および Java EE 5, 6, 7, 8 および ジュールをサポートします。	Jakarta EE 9 Web ₹	列(C).
」このプロジェクトを実行するときは常にこのサーバーを使用(P)		

③ コンソール部を確認し、Tomcatサーバが起動していることを確認します。



④ しばらく待機していると、ブラウザが起動し、Webページが表示されます。
 記述したコードのとおりに表示されていることを確認します。



⑤ 本手順で起動したTomcatサーバを停止する場合は、コンソール部にある ■ ボタンをクリックします。



⑥ Eclipseの終了は、右上の「×」ボタンで終了します。

以上