Java の環境構築

Pleiades All In One Eclipse のインストール

Pleiades All in One は統合開発環境である Eclipse 本体と、日本語化を行うための Pleiades プラグインおよびプログラミン グ言語別に便利なプラグインを加え、さらに Web アプリケーションサーバ(Apache Tomcat)をセットした Windows、Mac 向けパッケージである。

そのため、このソフトウエアをインストールするだけで、開発環境が準備できる

1.1 Pleiades All In One Eclipse のダウンロード

下記 URL ヘアクセスする

https://willbrains.jp/

MercaDoc Project									₽	GitHub	Spons	ors で支援	5 <mark>6</mark> 00	itHub Qi	iita	X Follow	@willbrains 11,556,6(BI 7ックマーク 04 downloads
▲ Pleiades 日本語化プラグイン	Pleiades All in One	ダウンロー	۲															
Smart Boarding ①×	Pleiades All in One は統合開発現 Full Edition の場合は JDK のイン 済みの Eclipse に適用したい場合	境である Eclipse ストールや <u>環境変</u> や、Linux で使用 ¹	本体と、日2 <u>数の設定が3</u> する場合は、	語化を行う 要で Eclip All in One	うための I se の各種 ではなく	Pleiades 口 設定も自 Pleiades	^プ ラグイ 動で行わ <u>プラグ-</u>	ンおよび れるたる (ン本体	バブログ め、ダウ のみを	ラミング リンロー ダウンロ	7言語別 ドして起 ードして	に便利なブ 動すれば、 こください。	ラグインをま すぐに日本調 IEJパージョン	とめた Wind 暦化された Ec 法示	dows, clipse	Mac 向け を利用でき	パッケージ !ます。既に	です。また、 Cインストール
ダウンロードは こちら	Eclipse 2024 最新版	Eclipse 2024/0	e 2023		E 2	Eclipse 2 023/02/1	022 2											
ダウンロード >		Windows				1												
問題・不具合などの報告	Eclipse バージョン	32bit 64bit	Intel A	AM 1	.4 1.	Java Fui 5 6	7	8	11	-232 17	21	25						
	2025 (予定)																	
Eclipse コード生成 AI プラグイ	2024	0	(0	0	0	\otimes							
	2023	0	0	3				0	0	0	0							
	2022	0	0	3		0	0	0	0	\otimes								

上記のような画面となるので、Eclipse2023を選択する

※Eclipse2023の方が安定しているため

Standard Edition	表示	Platform	Ultimate	Java	C/C++	РНР	Python
Windows x64 32bit は 2018-09 で終了	Full Edition	Download	Download	Download	Download	Download	Download
Mac Mac版について (Qiita)	Full Edition	Download	Download	Download	Download	Download	Download
		-				~~~~~	-

いくつかの開発言語に対応したパッケージがあるが、ここでは

Java の Full Edition を選択する

自動でダウンロードが始まる。



1.2 Pleiades のインストール

① ダウンロードした exe ファイルを実行する

± 9020-F	× +					-	0
$\leftarrow \rightarrow \uparrow$	C 7970-F			×	ダウンロードの検索		
④ 新規作成 - X	0 0 0 0 0 1 2482 -	≡ #≅ · ···					
<u>↑</u> #-4	6.6	更新日時	维防	サイズ			
E 507U-	pleiades-2023-12-java-win-64bit-jre_20240218.exe	2024/09/10 13:25	アプリケーション	1,199,258	2		
> 7291-97	○ 野日 N SCodeUserSetup-x64-1.93.0.exe 为 Z 1.1.	2024/09/10 13:23 4 G8 3	アプリケーション	98,877 K	8		
) 📑 F#1%7F	x86_64-14.2.0-release-win32-seth-msvcrt-rt_v12-rev0 〜 今年に入って(今月は会めず)	2024/09/09 14:35	ファイル フォルダー				
> 🚞 #ft7rt&	₩ サンプルデー9.zip	2024/05/16 14:46	正編 (zip 形式) 7#	5 K	в		
-	sakura-tag-v2.4.2-build4203-a3e639150-Win32-Release-Int.	2024/05/16 10:36 2024/05/16 10:28	注題(210 形式) フオー アプリケーション	5.425 K	8) 8)		
T70F97 #	🗃 M5Teams-x64.msix	2024/05/16 9:57	MSEX 774 lb	169,920 K	8		
± 9720-F ≠	🖏 OxomeSetup.exe	2024/05/16 9:50	アプリケーション	1,345 K	8		
F#1X2F #	<u>⊨</u> サンプルナータ	2024/05/16 14:46	3812 3889-				
 ビクチャ # ジョク # 	늘 sakura-tag-v2.4.2-builid4203-a3e63915b-Win32-Release-Inst.	2024/05/16 1041	32436-38369-				
0 個の項目							1

② インストール先を指定し「解凍」ボタンをクリックする



③ フォルダが作成されることを確認する

L	OS (C:)	× +					-	×
\leftarrow	\rightarrow \uparrow (C □ → PC → OS (C:) →				OS (C:)の検索		۹
+	新規作成 🗸 🛛 👗	0 10 40 10	↑↓ 並べ替え ~ 🛛 🗮	表示 ~ •••				詳細
~	🛀 OS (C:)	へ 名前	更新日時	種類	サイズ			
>	Apps	E Apps	2024/04/05 16:55	ファイル フォルダー				
>	📒 backup	📒 backup	2024/04/05 16:55	ファイル フォルダー				
>	🚞 dell	all dell	2024/04/05 16:55	ファイル フォルダー				
>	Drivers	Drivers	2024/04/05 16:55	ファイル フォルダー				
	e-logo	e-logo	2024/04/05 16:55	ファイル フォルダー				
		Iangpacks	2024/04/05 16:55	ファイル フォルダー				
		PerfLogs	2022/05/07 14:24	ファイル フォルダー				
	PerfLogs	🔤 pleiades	2024/09/10 13:54	ファイル フォルダー				
>	pleiades	Program Files 作成日時: 2024/09/10 13:	:54 2024/05/21 9:32	ファイル フォルダー				
>	🚞 Program File:	Program Files (x サイズ: 1.68 GB フォルダー: 2023-12	2024/09/09 14:35	ファイル フォルダー				
>	🚞 Program File:	training	2024/09/09 16:14	ファイル フォルダー				
>	늘 training	Windows	2024/09/09 11:43	ファイル フォルダー				
>	Windows	늘 ユーザー	2024/04/08 9:16	ファイル フォルダー				
13 個	の項目							

②で指定した場所に pleiades フォルダが作成されていることを確認する

1.3 Eclipse の起動

	eclipse	× +				-	- 0	×
÷	\rightarrow \uparrow	C □ PC → OS (C:) →	pleiades > 2023-12	2 > eclipse >		eclipseの検索		۹
+	新規作成 🗸 🛛 👗	0 🗋 🖉 🖄	↑↓ 並べ替え ~ 🛛 🗮	表示 ~ •••				詳細
~	🗐 OS (C:)	へ 名前	更新日時	種類	サイズ			
>	🚞 Apps	configuration	2024/02/18 11:21	ファイル フォルダー				
>	🚞 backup	aropins 🔁	2024/02/18 11:21	ファイル フォルダー				
>	📒 dell	features	2024/02/18 11:21	ファイル フォルダー				
>	Drivers	p 2	2024/02/18 11:21	ファイル フォルダー				
	e-logo	plugins	2024/02/18 11:21	ファイル フォルダー				
		readme	2024/02/18 11:21	ファイル フォルダー				
		.eclipseproduct	2023/12/01 6:19	ECLIPSEPRODUCT	1 KB			
	PerfLogs	C artifacts.xml	2023/12/01 21:22	Microsoft Edge HT	455 KB			
>	pleiades	🗢 eclipse.exe	2023/12/01 21:25	アプリケーション	521 KB			
>	Program File:	eclipse.exe -clean.cmd	2017/11/03 23:50	Windows コマンド ス	1 KB			
>	🚞 Program File:	eclipse.ir サイズ: 520 KB	2024/02/18 11:22	構成設定	1 KB			
>	training	eclipsec.exe	2023/12/01 21:25	アプリケーション	233 KB			
>	Windows	🗋 lombok.jar	2024/02/18 11:16	JAR ファイル	1,965 KB			
13 個	の項目							

① C:\pleiades\2023-12\eclipse\フォルダの eclipse.exe を実行する

作成したソースコードなどのファイルを保存する場所を設定する画面が表示されるので、「起動」ボタンを押下 する

■ Eclipse IDE ランチャー	_		×
ワークスペースとしてのディレクトリー選択 Eclipse IDE は設定や開発成果物を保管するためにワークスペース・ディレク <mark>ト</mark> リーを	使用します。		
/workspace		~ 参照	₹(<u>B</u>)
□ この選択をデフォルトとして使用し、今後この質問を表示しない(<u>U</u>) ▶ 最近使ったワークスペース(<u>R</u>) 起動(L	J	キャンセノ	Ŀ

- ② C:\pleiades\2023-12\workspace フォルダが作成される
- ③ Eclipse の画面が表示されればインストール完了

1.4 パッチ適用

Eclipse の WTP プラグインの不具合を修正する必要がある。

下記 URL を参考にしパッチを適用する

<u>Eclipse プラグインの修正(Windows)</u>

Java アプリケーションの開発方法

Eclipse をインストール後、コマンドラインに「Hello World」を表示するまでの流れを説明する

1.1 プロジェクト作成

プロジェクトとは、プログラム一式をまとめて管理するもので、プロジェクトの中にソースファイルやライブラリなどを格納していく

① 「ファイル」⇒「新規」⇒「Java プロジェクト」の順にクリックする



② プロジェクトの名前を入力する

← 新規 Java プロジェクト					D	×
Java プロジェクトの作成						1
プロジェクト名を入力してください。						
加ジェクト名(P): Sample						
✓ デフォルト・ロケーションを使用(D)						
					参照(<u>R</u>)	
_ JRE						
● 実行環境 JRE の使用(⊻): JavaSE-2						~
● プロジェクト固有の JRE を使用(<u>S</u>): Java21						~
● デフォルトの JRE 'Java21' およびワークスペース・:	ンパイラー設定を使用する(<u>A</u>)				<u>JRE を構成</u>	
L _「 プロジェクト・レイアウトーーーーーーーーーーーー						
│ │ ● プロジェクト・フォルダーをソースおよびクラス・ファイ	レのルートとして使用(<u>U</u>)					
● ソースおよびクラス・ファイルのフォルダーを個別に1	₣成(<u>C</u>)				フォルトを構。	<u>戎</u>
│ │					新規(<u>W</u>)	
ワーキング・セット(<u>O</u>):				~	選択(<u>E</u>)	
 _ Eジュール						
module-info.java ファイルの作成(<u>M</u>)						
tジュール谷(M): ■ コメントの生成(G)						
0	< 戻る(<u>P</u>	<u>3)</u> 次へ(<u>N</u>) >	完了(E)		キャンセノ	

プロジェクト名に「Sample」と入力し「次へ」ボタンを押下する

듣 新規 Java プロジェクト	プロジェクトのフォルダ構			- 0	×
Java 設定	成が表示される				
Java ビルド設定を定義します。					
◎ Y-ス(<u>S</u>) きプロジェクト(<u>P</u>)	イブラリー(<u>L)</u> 🔸 順序およびエクスポー	-ト(<u>O)</u> 🔍 モジュール依存関係(<u>N</u>	<u>1</u>)		
₩ ₩ 1 T T 3 ×				Fa #	
> 📇 src					
 ▼詳細					
■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	プロジェクトに新規ソース・フォルダーを追	れする場合は、これを使用します			
≌ ≌ <u>追加ソースのリンク</u> : ファイル・シ	マステムに、追加ソース・フォルダーとしてイ	使用しなければならないフォルダ−1	がある場合、このリンクを使	用します。	
<i>尋 プロジェクト 'Sample' のビルド</i> ・	・ <u>パスへの追加</u> :プロジェクトがパッケーシ	がおよびソース・ファイルのルートであ	る場合、プロジェクトをビルド	・パスに追加し	します。ビ
ルト・ハスのエントリーは、コンハ	イフ−には リ 視 ぐ、ビルトのために 使用	anto.			
■ ソース・フォルダーごとに出力フォルタ	ダ−の指定を可能にする(<u>C</u>)				
デフォルト出力フォルダー(<u>I</u>):					
Sample/bin				参照(<u>M</u>	<u>/</u>)
<u>a</u>				+	
				-+77	

「完了」ボタンを押下する

③ プロジェクトが作成され、プロジェクトエクスプローラーに追加される



1.2 パッケージ作成

ソースファイルなどを格納するためのパッケージを作成する

① 「ファイル」⇒「新規」⇒「パッケージ」の順に選択する

(= c	:\pleiades\2023-12\workspace	- Eclipse IDE						
ファイ	イル(F) 編集(E) ナビゲート(N)	検索(A)	プロジェクト(P)	実行(R)	ウィンドウ(W)	ヘルプ(H)		
•	新規(N) ノアイルを開く(.) ファイル・システムからプロジェクトを 最近使ったファイル(F)	AI 開<	t+Shift+N >	Grace Graee	lle プロジェクト en プロジェクト ng スターター・プロジ ng 入門コンテンツ	ジェクト (Spring Initializ のインポート	zr)	← → → →
	エディターを閉じる(C) すべてのエディターを閉じる(L)			 動的 Java ブロジ 	Web プロジェクト プロジェクト シェクト(R)			
() () ()	保存(S) 別名保存(A) すべて保存(E) 前回保存した状態に戻す(T)			 パック グラフ グラフ グラフ グラフ グラフ 	ージ -フエース フェース			
11 m 11 m 12 m	移動(V) 名前変更(M) リフレッシュ(F) 選択リソースのカウント タブ <-> スペースの変換 行区切り文字の変換(V)		F2 F5	 ○ フィッチ ◎ レコー ○ 注釈 ● ソーフ ▲ ソーフ ▲ Java ▲ フォ川 首 ファイ 	± ド ワーキング・セット ダー ル			
الله الله الله	印刷(P) インポート(I) エクスポート(O)			 書表題 ● 表題 ● HTN ● CSS ● Java 	なしのテキスト・ファ IL ファイル ファイル Script ファイル	イル		
_	プロパティー(R) ワークスペースの切り参え(MA)		Alt+Enter	📑 サン	別レ(X)			
			,	<u></u> 701	也(O)		Ctrl+N	

② パッケージ名を入力する

듣 新規 Java パッケージ				×
Java パッケージ				
新規 Java パッケージを作成しま	き。		_	
パッケージに対応するフォルダーを作	乍成します。			
ソース・フォルダー(<u>D</u>): Sample	e/src		参照(<u>C</u>)
名前(<u>M</u>): sar	mple			
■ package-info.java を作成す ■ コメントの生成(<u>G</u>) (テンプレ	る(<u>C</u>) トの構成お			
Q		完了(E)	キャンセル	

名前に「sample」と入力し「完了」ボタンを押下する

③ プロジェクトの src 直下にパッケージが作成される



1.3 クラス作成

1.2 で作成したパッケージの下にクラスを作成する

① パッケージを選択した状態で、「ファイル」⇒「新規」⇒「クラス」の順に選択する



② クラス名などの情報を入力する

듣 新規 Java クラス		_	۵	x c
Java クラス				2
新規 Java クラスを作成	します。			
ソース・フォルダー(<u>D</u>):	Sample/src		*	[:] 照(<u>O</u>)
パッケージ(<u>K</u>):	sample		参	·照(<u>W</u>)
■ エンクロージング型(Y):				照(<u>W</u>)
名前(<u>M</u>): 修飾子:	Sample ・ public(P) ・ パッケージ(C) ・ private(V) ・ protected(I) abstract(I) final(L) static(C) ・ なし(E) ・ sealed(S) ・ non-sealed(N) ・ final(L)			
スーパークラス(<u>S</u>):	java.lang.Object		*	▶照(<u>E</u>)
インターフェース(<u>I</u>):			<u>ن</u>	1加(<u>A</u>)
				余去(<u>R</u>)
どのメソッド・スタブを作成	」 し.主すか?		J	
	 ✓ public static void main(String[] args)(フーパークラスからのコンストラクター(U) ✓ 縦承された抽象メソッド(H) 			
コメントを追加しますか? (:	テンフレートの構成およびデフォルト値については <u>ここ</u> を参照) ■ コメントの生成(<u>G</u>)			
0		5	+	ャンセル

クラス名を Sample と入力する

public static void main(Staing[] args)にチェックを入れると Java プログラム作成時に、必要な main メソッドが自動で生成される

継承された抽象メソッドはデフォルトでチェックが入るので、そのままチェックを付けておく

一通り入力後「完了」ボタンを押下する

③ Sample.java ファイルが生成される





1.4 プログラムの作成

Sample.java にソースを追加する



1.5 実行

実行する

「実行」⇒「実行」の順にクリックする



② コンソールに Hello World と表示されることを確認する



Java サーブレットでの Web アプリケーション開発方法

Eclipse を起動し、Web プロジェクトを作成し起動するまでの手順を説明する

1.1 動的 Web プロジェクトの作成

※動的 Web プロジェクトとは、Web アプリケーションのためのプロジェクトのこと

① 「ファイル」⇒「新規」⇒「動的 Web プロジェクト」を選択

	C:\pleiades\2023-12\workspace - Ecl	ipse IDE			
ידר	<mark>イル(F)</mark> 編集(E) ナビゲート(N) 相	検索(A) プロジェクト(P)	実	行(R) ウィンドウ(W) ヘルプ(H)	
	新規(N) ファイルを開く(.)	Alt+Shift+N >	*	Gradle プロジェクト ・ Maven プロジェクト	
2	ファイル・システムからプロジェクトを開く 最近使ったファイル(F)	>	(전) (전) (전) (전) (전) (전) (전) (전) (전) (전)	Spring スターター・プロジェクト (Spring Initializr) Spring 入門コンテンツのインポート	
		Ctrl+W Ctrl+Shift+W	2 2 2 2 2	動的 web フロジェクト Java	
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	保存(S) 別名保存(A) すべて保存(E) 前回保存した状態に戻す(T)		11 6 12	パッケージ クラス インターフェース	
1	移動(V) 名前変更(M)	F2	6 6 0 0	列挙型 レコード 注釈	
ان پیر مرب	リフレッシュ(F) 選択リソースのカウント タブ <-> スペースの変換	F5	81 14 11	ソース・フォルダー Java ワーキング・セット フォルダー	
.	17区切り又子の変換(V) 印刷(P)	Ctrl+P	 € 	ファイル 表題なしのテキスト・ファイル HTML ファイル	
د د ا	インホート(I) エクスポート(O)	Alt - Entor	S 1	CSS ファイル JavaScript ファイル	
	フークスペースの切り替え(W) 再開	Ait+Einter		サンプル(X) その他(O) Ctrl+N	

② プロジェクト名、ターゲット・ランタイムを選択

プロジェクト名:WebSample

ターゲット・タイムライン: Tomcat10(Java21)を選択

← 新規動的 Web プロジェクト	—		×
動的 Web プロジェクト			
スタンドアロンの Java ベースの Web アプリケーションを作成するか、新規または既存のエンタープライズ・アプリケーショ	シに追	加します。	
ブロジェクトタ(M)・WebSample			
- プロジェクトの場所			
ロケーション(L): C:¥pleiades¥2023-12¥workspace¥WebSample		参照	(<u>S</u>)
Tomcat10 (Java21)	~	新規ランタ	イム(<u>R</u>)
6.0			~
Tomcat10 (Java21) デフォルト構成	~	変更	
Tomcat10 (Java21) ランタイムを操作するための良い出発点です。後で新しい機能をプロジェクトに追加するため ルすることができます。	こ追加フ	ファセットを	インストー
EAR にプロジェクトを追加(<u>A</u>)			
EAR プロジェクト名(E): EAR	~ (亲	新規プロジ:	ェクト(<u>P</u>)…
■ ワーキング・セットにプロジェクトを追加(1)		新規(<u>)</u>	<u>M</u>)
ワーキング・セット(<u>O</u>):		選択(<u>E</u>)
? < 戻る(B) 次へ(N) > 完了(F)		 	ッセル

③ 完了後、プロジェクトが作成できていることをプロジェクトエクスプローラーで確認する



④ 動的 Web プロジェクトをサーバーへ追加

「サ-	-バー	リビコ	」ーを選択⇒	[Tomcat10	Java21	を選択=	⇒右クリッ	ク⇒「追	追加および	「除去」	を選択
				_							





│	-	
追加および除去		
サーバー上に構成済みのリソースを変更しま	ŧ	
サーバー上で構成するリソースを右側に移動し	んてください	
使用可能(<u>A</u>):	構成済み(C):	_
	🖡 WebSample	
	追加(<u>D)</u> >	
	< 除去(<u>R</u>)	
	すべて追加(<u>L)</u> >>	構成済み」へ移動
	<<すべて除夫(M)	できたことを確認
		後、「完了」ボタン
	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	を押下する
? < 戻る(<u>B</u>)	次へ(N) > 完了(E)	キャンセル

⑤ 「サーバー」ビューを選択⇒「Tomcat10_Java21」を選択⇒右クリック⇒「開始」を選択





1.2 サーブレットクラスの作成

※サーブレットは Java を使いサーバサイドプログラムを作成する技術のこと

① サーブレットクラスを作成する動的 Web プロジェクトを選択し、右クリック⇒「新規」⇒「その他」を選択

e C:\pleiades\2	023-12\workspace - Eclipse IDE		
ファイル(F) 編集	〔(E) ナビゲート(N) 検索(A)	プロジェクト(P) 実行(R) ウィン	ドウ(W) ヘルプ(H)
📑 – 🖪 🖻 I 🕻	→ 新規(N)	· · · ·	📑 プロジェクト(R)
■ パッケージ・エク	中にジャンプ(I)		📋 วิราไม
> 📂 Sarnple	表示(W)	Alt+Shift+W >	🞽 フォルダー
> 📂 Servers	🗈 בצ-	Ctrl+C	📓 CSS ファイル
🔹 🛩 WebSam	na 貼り付け		💁 HTML ファイル
> - setun	¥ 削除(D)	Delete	😭 JavaScript ファイル
v 🗁 src		Ctr <mark>+</mark> Alt+Shift+Down	📑 JSP ファイル
∽ ⊳ ma	移動(V)	5	iii サンプル(X)
-	石削変史(Ⅲ)	F2	き その他(O) Ctrl+N
× 💆	🖮 インポート(I)		

② 「ウィザードを選択」画面において、「Web」⇒「サーブレット」を選択し「次へ」ボタンを押下する

━ ウィザードを選択					×
ウィザードを選択)
ウィザード(<u>W</u>):					
フィルター入力					
 > SVN Web G CSS ファイル M HTML ファイル JSP タグ JSP ファイル Web フラガメント・プロジェク ボ サーブレット ネ フィルター 塗 リスナー 静的 Web プロジェクト 	71-				I
0	< 戻る(<u>B</u>)	次へ(<u>N</u>) >	完了(E)	キャンセル	

③ 「サーブレット作成」画面において、Java パッケージ名、クラス名を入力し「次へ」ボタンを押下する

🔵 サーブレット作成						
サーブレット作成						R
クラス・ファイルの宛会	先をを指定します。					
ノロシエクト(<u>P</u>):	WebSample					<u>~</u>
ソース・フォルダー(<u>D</u>):	/WebSample/sro	:/main/java				参照(<u>O</u>)
lava パッケージ(<u>K</u>):	controllers					参照(<u>W</u>)
クラス名(<u>M</u>):	SampleServlet					
スーパークラス(<u>S</u>):	jakarta.servlet.ht	tp.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>):	SampleServlet					参照(<u>W</u>)
0		< 戻る(<u>B</u>)	 次へ(<u>N)</u> >	一定	(E)	キャンセル

④ サーブレットの名前や URL パターンなどのサーブレットクラスに関する情報を設定し「次へ」ボタンを押下する

(サーブレット作成			
サーブレット作成			R
サーブレット配備記述子固有の情報	服を入力してください。		
名前(<u>M)</u> : SampleServlet			
説明(<u>S</u>):			
初期化パラメーター(<u>P</u>):			
名前	值	説明	追加(<u>A</u>)
			編集(<u>E</u>)
			除去(<u>R</u>)
URL マッピング(<u>U</u>):			
/SampleServlet			追加(<u>D</u>)
			編集(<u>T</u>)
			除去(<u>V</u>)
■ 非同期サポート(<u>Y</u>)			
0		N) > 完了(F)	キャンヤル

⑤ doGet、または doPost の必要なメソッドにチェックをいれ、「完了」ボタンを押下する

(= サ−ブレット作成				×
サーブレット作成			(3
修飾子、実装するインターフェース、および生成するメ 	ソッド・スタブを指定してください。			
修飾子: v public(<u>P</u>) abstract(<u>T</u>)	final(<u>L</u>)			
インターフェース(<u>)</u>):			追加除去	(A)) \$(R)
 どのメソッド・スタブを作成しますか?				
 □ スーパークラスからのコンストラクター(C) ✓ 継承された抽象メソッド(<u>H</u>) 	_		このチェ	ックははずす
init destroy aetServletInfo service	detServletConfig	l		
✓ doPost(P)] doDelete(<u>D</u>)] doTrace(<u>T</u>)			
	次へ(<u>N</u>) > 完了(<u>P</u>)		キャンセノ	L J



⑥ ソースコードを編集する

「src/main/java/controllers」内にサーブレットクラスが作成され、エディタには作成したサーブレットクラスの内容が 表示される。⑤でチェックをつけたメソッドが自動で作成されている。

ここでは、doGet()関数を以下のように修正する



※この行は削除しない

@WebServlet アノテーションといい、URL パターンを設定している

1.3 サーブレットクラスの実行

① 実行するサーブレットクラスを選択し右クリック⇒「実行」⇒「サーバで実行」を選択する

· 🗸 💈	🖻 main	^	則际(D)	Delete		12	∗ Servlet imp
	🗸 🗁 java			Ctrl+Alt+Shift+Down			_ */J nWobSorvlot("/
	✓ ≥ controllers	ø					public class S
	SampleServl		移動(V)			16 17	private st
`	🗁 webapp		名前変更(M)	F2	Boot ダ 👘 🗖	189) /**⊲ ∞ ♠ !!!
	> 🗁 META-INF				- 🔄 🕑 🕼 🗖 🏟 🗄	20	* €see H1 */↓
	🗸 🗁 WEB-INF	è	インポート(I)			21€	protected
	📂 lib	4	エクスポート(〇)…			23	System
X .(lasspath	<u>_</u>	וריצאין בא	E5		24	}ਹ ਹ
<u>x</u> .	roject	¥	<i>J/<i>VJ</i>/(r)</i>			269	/**⊴
			ソース	>		27	* €see H1 */↓
			₩龄(∧)			▲29 ⊜ ∲30	protected
				, ,		31	doGet (
			直揆(L)	,		32	}र र
		E	Windows エクスプローラー			34	لم
-カ-	שרענב <u>■</u>	•	コマンド・プロンプト		📃 – 🃑 – 🗖		रू } रू
at10	ava21 [Apache Tomcat]		Checkstyle フィックスの適用) [pid: 10800]	37	_
:]		0	カバレッジ(V)	>	13 14:11:50		
		5	宝行(R)		◎ 1 サーバーで実行	Ŧ	Alt+Shift+X R
: 21		1	デバッグ(ロ)				
				, ,	実行 の構成(N)	
		18≣		,			
			Quick JUnit(Q)	>			
			Checkstyle	>			

② 実行するサーバを選択し完了ボタンを押下する



③ サーバを再起動する必要のあるときは、以下の画面が表示されるので、「サーバ再起動」を選択し OK ボタンを押



④ ブラウザに実行結果が表示される

•	Ô		☐ localhost:8080/WebSample/Samp × +
\leftarrow	С	Ŵ	(i) localhost:8080/WebSample/SampleServlet
Hello	o Java	worl	d

1.4 JSP ファイルの作成

JSP ファイルは、HTML と Java のコードで構成されたファイルのこと

① 動的 Web プロジェクトを選択し、右クリック⇒「新規」⇒「その他」を選択



② 「Web」⇒「JSP ファイル」を選択し「次へ」ボタンを押下する

● ウィザードを選択		×
ウィザードを選択		*
新規 JSP ファイルの作成		
ウィザード(<u>W</u>):		
		_
Y ► Web		
The cost of the c		
		11
◎ JSP ファイル 健 Web フラガイント・プロジェクト		11
? < 戻る(B) 次へ(N) > 完了(F)	キャンセル	

③ ファイル名を入力する。保存場所は「src/main/webapp/」フォルダとし、「完了」ボタンを押下する

듣 新規 JSP ファイル							\times
JSP							
新規 JSP ファイルを作成しま	き。						
ᆆᄀᆇᆘᄸᅳᅕᅐᆂᆂᄮᆣᇔ							
税ノイルターを八刀よたは迭 WebSample/src/main/we	水(<u>E</u>)。 ehann						
	spabb	_	_	_	_	_	_
i fi ← → Sample							
> 🗁 Servers							
🗸 🐸 WebSample							
😂 .settings							
> ≽ build							
v 🗁 src							
✓ Imain > Imain							
Juvu							
ファイル名(<u>M</u>): sample.jsp							
 拡張(<u>A</u>) >>							
	=7/0			676		مار د باد	
	< 戻る(<u>B</u>)	次/	<u>(N)</u> >	_ 完了(E)		キャンセ	

④ ファイルが作成できる



⑤ ソースコードを編集する

SampleServlet.java

サーブレットクラスから作成した sample.jsp ファイルを呼び出すように doGet()関数を修正する

/***
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
 */J
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Servle
 // TODO_Auto-gonorated_method_stub
 RequestDispatcher dispatcher = request.getRequestDispatcher("sample.jsp");J
 dispatcher.forward(request,response); | J
}J

■ sample.jsp

HTML を修正し、画面に「Hello Java World」と表示する



※実行は 1.3 サーブレットの実行を参照