- 1. Eclipse の起動
- (1) デスクトップの eclipse をダブルクリックします.



(2) 「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)」をチェック し,「起動」をクリックします.

● Eclipse ランチャー	$\times$
ワークスペースとしてのディレクトリー選択	
Eclipse は設定や開発成果物を格納するためにワークスペース・ディレクトリーを使用します。	
ワークスペース( <u>W</u> ):/workspace ~ 参照( <u>B</u> )	
<ul> <li>最近使うたワークスパース(<u>N</u>)</li> </ul>	
起動(上) キャンセル	

(3) しばらくすると、以下のウインドウが表示されます.

Ci¥pleiades¥workspace - Eclipse						- o ×
ファイル(E) 編集(E) ソース(S) リファクタリング(E) ナビゲート(N) ?	検索(A) プロジェクト(P) 実行(B) ウインドウ(W) ヘルブ(	H)				
📑 • 🔛 🐚   🕲 • % • i 🖆 • 😂 • 🖻 • 🚱 • i	e = O = 9  = 9_ = G / G / E / E	I II 🕘 🕶 🕻	) x 🗿 🖉 + i	1 - 10 O -	4	クイック・アクセス 🔡 🖬 C/C++
陷 70917Ի-17370-7- 🕺 😑 😫 🔝 🔍 🗖 🗆						 ₽ 79 22
						Ş1 V
						表示するアウトラインはありません。
						-1°   ÷. L
	説明 ^	リソース	バス	ロケーション	뒻	
					_	
UTF-8 CRLF 0項目が選択されました						

- 2. プログラムの入力
- (1) 最初に C プロジェクトを新規に作成します.「新規」「C プロジェクト」の順にクリ ックします.

C:¥pleiades¥workspace - Eclipse

(2) この例では、プロジェクト名を「hello」と入力し、ツールチェーンは「MinGW GCC」 が選ばれていることを確認し、「完了」をクリックします。

● C プロジェクト	—		$\times$
C プロジェクト 選択したタイプの C プロジェクトを作成します			2
プロジェクト名(II): hello			
☑ デフォルト・ロケーションを使用(型)			
ロケーション( <u>L</u> ): C:¥pleiades¥workspace¥hello		参照( <u>R</u> ).	
ファイル・システムを選択(Y): <b>デフォルト</b> ~			
プロジェクト・タイプ: >  > GNU Autotools >  >  >  >  >  >  >  >  >  >  >  >  >	を表示		
(P) (N) > 完了(E)		キャンセ	Jμ

(3) プロジェクト・エクスプローラに「hello」が表示されます. 今後は、プログラム毎にプロジェクトを新規に作成していきます.

● Ci¥pleiades¥workspace - Eclipse ファイル(E) 編集(E) ソース(S) リファクタリング(D) ナビゲート(N)	検索(A) プロジェクト(2) 実行(8) ウインドウ(30) ヘルプロ	0		- a ×
📑 - 🗟 🐘 😸 - 🐔 - 🖬 📷 - 68 - 67 - 66		- 	(例・例・(の・の・)	クイック・アクセス
16 70927 12 50-7- X E X V - 0				 <b>9=72</b> ⊗ 9 □
> S hella				
				表示するアウトラインはありません。
	😰 開題 😫 🧔 タスク 💷 コンソール 🔲 プロパティー			과 한 것으 다
	0項目			
	說明	リソース パス	ロケーション型	
UTF-8 CRLF 🖾 hello				

(4) 次はソースファイルを作成します.プロジェクト・エクスプローラの「hello」が選 択されている状態で、「新規」「ソース・ファイル」の順にクリックします.

C¥pleiades¥workspace - Eclipse
 ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) ブロジェクト(P) 実行(R) \*
 アイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) ブロジェクト(P) 実行(R) \*
 ア・シーマン(C) マース・クロジェクト
 C++ ブロジェクト
 C/C++ ブロジェクト
 C/C++ ブロジェクトへ変換
 C/C++ ブロジェクトへ変換 (C/C++ ネーチャー追加)
 ジース・フォルダー
 フォルダー
 マース・ファイル
 ヘッダー・ファイル

(5) この例では、ソース・ファイル名は「hello.c」と入力し、「完了」をクリックします.
 C 言語のソースファイルの拡張子は「.c」になります. 拡張子を忘れないようにしてください.

⊜ 新規ソース・ファイ	JL	_		Х
ソース・ファイル				
新しいソース・ファイル	を作成します。		<u></u>	
ソース・フォルダー(D):	hello		参暇(B)	
ソース・ファイル(E)	hello.c		2 Mil( <u>2</u> )	
テンプレート( <u>T</u> ):	デフォルト C ソース・テンプレート	~	構成	
0	7	<u>⊜</u> 7(□)	キャンカル	
•	N	H I (E)	47700	

(6) プロジェクト・エクスプローラには、hello プロジェクトの配下に、「インクルード」
 と「hello.c」が表示されます.エディタには、自動生成されたコードが表示されます.

hello/hello.c - C:¥pleiades¥workspace - Eclipse ファイル(E) 編集(E) ソース(S) リファクタリング(T) 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) / 🚹 プロジェクト・エクスプローラー 🛛 📄 🤤 🚏 🍯 🗖 🚺 🔒 hello.c 💥 1⊝ /\*~ 🗸 📂 hello \* hello.cd > 🔊 インクルード 2 \*\_ 3 > 🖻 hello.c \* Created on: 2017/09/10 4 \* 5 Author: dohi \*/~ 6 7 🔊 8 2 9

#include <stdio.h> int main(void) { printf("Hello World¥n"); return 0; } hello/hello.c - C:¥pleiades¥workspace - Eclipse ファイル(E) 編集(E) ソース(S) リファクタリング(I) 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) 📑 🖌 🔚 🐚 👻 🔻 🌾 🖛 📾 🖛 📮 🔌 😳 📸 ד 😂 ד 🔂 ד 🐼 ד 🔅 🖬 🚸 ד 🚱 ד 🥵 ד 🥵 🖓 🔁 プロジェクト・エクスプローラー 🛛 📄 🔄 🍃 🗢 🗖 🗖 c \*hello.c 🛛 🗸 📂 hello 10/\*~ \* hello.ca 2 > 🗊 インクルード \*\_ З > 💽 hello.c Created on: 2017/09/10 4 \* 5 Author: dohi⊲ \*/~ 6 7 8 #include <stdio.h>⊶ 9 1( ⊜int main(void)↩ 11 12 {≈∣ 13 printf("Hello World\n");~ 14 return 0;⊶ 1 1 11 3. ビルド (1) 「すべてビルド」をクリックします. hello/hello.c - C:¥pleiades¥workspace - Eclipse ファイル(F) 編集(E) ソース(S) リファクタリング(T) 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) 📑 + 🖫 🐚 🛞 + 🔨 + 🔜 🍋 + 🖳 🔌 30 🔞 + 🚳 + 🗗 + 🚱 + 🐼 + 🕗 - 🌾 + 🔿 + 🧏 + 🥵 -陷 วือรัรวระบราย 🗈 \*hello.c 🛛 すべてビルド (Ctrl+B) 🗸 📂 hello 19/\*~ \* hello.cd > 👔 インクルード > 💼 hello.c 2 \*,... 3 \* Created on: 2017/09/10 4 \* 5 Author: dohi⊲ 6 \*/~ 7 🎝 8 🔊

## (2) ビルドの結果がコンソールに表示されます.

hello/hello.c - G¥pleiades-4.7.0¥workspace - Eclipse			– ø ×
ファイル(E) 編集(E) ソース(S) リファクタリング(I) ナビゲート(N)	検索(品) プロジェクト(ピ) 実行(出) ウィンドウ(W) ヘルプ(出)		
📑 • 🗟 🐚   10 • % • 🛍 🙆 • 1 🖻 🗙 🞯 1	<mark>3 • 63 • 6<sup>°</sup> • 6<sup>°</sup> • <sup>†</sup> ‡</mark> • 0 • 9₁ • 9₄ • <i>i 🖉 🕭 G</i> 🛷 • i A 🕫 🖩 🗊 🖗	- §i - ♥ ♥ - ⇒ -	クイック・アクセス 😰 🖬 C/C++
🔁 אין דער	le hellouc 38		🞥 79 🕴 🗐 97 🖲 9 🗖 🗖
<ul> <li>✓ Sindle</li> <li>&gt; ⊗in f(f))=</li> <li>&gt; ⊗in f(f))=</li> <li>&gt; ⊗in f(f))=</li> <li>&gt; ≥ Debug</li> <li>&gt; ≥ Debug</li> <li>&gt; ≥ hello.c</li> </ul>	10 <sup>1</sup> / <sub>2</sub> <sup>1</sup> / <sub>2</sub> 2 * hello.c <sup>⊥</sup> 3 * <sup>2</sup> / <sub>2</sub> 4 * Created on: 2017/09/10 <sup>⊥</sup> 5 * Author: dohl <sup>⊥</sup> 6 */ <sub>2</sub> 7 <sup>⊥</sup> / <sub>2</sub> 8 <sup>⊥</sup> / <sub>2</sub> 9 #include <stdio.h><sup>⊥</sup>/<sub>2</sub> 10 <sup>⊥</sup>/<sub>2</sub> 10 <sup>⊥</sup>/<sub>2</sub> 11 firt main(void)<sup>⊥</sup> 12 <sup>⊥</sup>/<sub>2</sub> 13 <sup>⊥</sup>/<sub>2</sub> 14 <sup>⊥</sup>/<sub>2</sub> 15 <sup>⊥</sup>/<sub>2</sub> - return 0<sup>1</sup>/<sub>2</sub><sup>⊥</sup> 17 <sup>⊥</sup>/<sub>2</sub></stdio.h>		E I <sup>4</sup> 2 & x <sup>2</sup> • ₩   > ~ ~ stdic.h main(void) : int
		>	1
	国語 2 タスク ロンソール 88 □ プロパティー	ê ê 😵 🗔	🔉 = 🔍 🔍 🛫 ピ • ピ • 🙂 🗆
	COT Build Console[held] 2125514 ***********************************		×
			UTF-8 CRLF

### 「保管」を忘れて「すべてビルド」を行うと、コンソールにエラーが表示されます.

hello/hello.c - Ci¥pleiades¥workspace - Eclipse			– a ×
ファイル(E) 編集(E) ソース(S) リファクタリング(E) ナビゲート(N)	検索(A) プロジェクト(P) 実行(B) ウィンドウ(W) ヘルプ(H)		
📑 = 🗟 🐚   🕲 = 🌜 = 🛍 🖨 = 💷 🔌 🧿 👔	] • 68 • 61 • 67 • 1 ‡ • 0 • 9₁ • 9₄ • 1 ∰ ∰ ∰ ∰ √ • 1 ≠ 1 ≠ 10 • 11 € 1 • 11 •		クイック・アクセス 🖻 🖬 C/C++
🔁 プロジェクト・エクスプローラー 🕴 😑 😫 😓 🗢 🗖 🗖	C *hellouc 23	- 0	🚼 79 🖄 🗐 97 🛞 9 🖳 🗖
<ul> <li>✓ Enhelic</li> <li>&gt; Ø 7429μ−F</li> <li>&gt; Debug</li> <li>&gt; Enhelice</li> </ul>	10 / /		<ul> <li>P □ I<sup>2</sup><sub>Z</sub> X x<sup>6</sup> • # <sup>∞</sup></li> <li>stdic.h</li> <li>main(vord):int</li> </ul>
( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( )	¢	>	
		4 6 😵 🗔	<u> </u>
	COT Build Console(hello) gcc -00 -g3 -Wali -c -fmessage-length=0 -o hello.o "\hello.c" gcc - hello.cex hello.o C:/pitades/cclipsc/mingu/bin//lib/gcc/x86_64-w64-mingw32/7.1.0///.x86_64- collect2.eex: error: 1d returned 1 exit status 16:08:57 Build Finished (took 801es)	w64-mingw32/lib//lib/	A libmingw32.a(lib64_libmingw32_a-cr
	(		,
			LITE-8 CRLE

### 4. 実行

(1) 「実行」をクリックします.

#### hello/hello.c - C:¥pleiades¥workspace - Eclipse

ファイル(F) 編集(E) ソース(S)	リファクタリング(T)	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)
📑 🗕 🔚 👘 🛛 👟 🔦	- 🗟 🕒 - 📃	X 🖸 🛙	🕯 🔻 🚳 ·	• 🖻 • 🞯 •	- * -	0 9 -	<b>9</b> - 10
陷 プロジェクト・エクスプローラー	ಷ 🖻 😫 🛸	~	.c *hell	o.c 🛛		実行	
<ul> <li>✓ ご hello</li> <li>&gt; 剤 インクルード</li> <li>&gt; ご Debug</li> <li>&gt; ご hello.c</li> </ul>				« hello.c- « Created on: « Author: « Author: «/- include <std nt main(yoid printf("Hel - return 0;-</std 	: 2017/09/0 : dohi⊐ io.h>⊐ io.h>⊐ Io \vorld¥0 Io \vorld¥0	04≓ n″);≓	

(2) 「C/C++アプリケーション」をダブルクリックします.

⊜ 実行構成		×
構成の作成、管理、および実行		
C () () () () () () () () () () () () ()	このダイアログから起動設定を構成します: □ - 選択したタイプの構成を作成するには、「新規」ボタンを押します。 ■ - 選択した構成を口ピーするには、「複製」ボタンを押します。 × - 選択した構成を除去するには、「削除」ボタンを押します。 → - フィルター・オブションを構成するには、「フィルター」ボタンを押します。 - 既存構成を選択して、それを編集または表示します。 <u>パースペクティブ</u> 設定ページから、起動パースペクティブの設定を構成します。	
?		実行( <u>R</u> ) 閉じる

(3) 「実行」をクリックします.

⊜ 実行構成		×
構成の作成、管理、および実行		
<ul> <li>○ ● ● ● ● ● ● ●</li> <li>○ 1/1/ター入力</li> <li>&gt; ● C (/C++ アブリケーション</li> <li>○ hello Debug</li> <li>○ C /(C++ リモート・アブリケーシ</li> <li>● 起動グループ</li> <li>● 起動グループ (非推奨)</li> </ul>	名前(№: hello Debug メイン (№ 引数) ■ 環境) □ 共通(①) ブロジェクト(P): hello C/C++ アブリケーション: Debug¥hello.exe 変数(♡ ブロジェクトの検索(∐) 起動前に (必要に応じて) ビルド ビルド構成: Select Automatically 〇 自動ビルドを使用可能にする 〇 自動ビルドを使用不可にします ⑨ ワークスペース設定の使用 ワークスペース設定の構成	参照(B) 参照(R)
< > フィルターー致: 5 / 5 項目	前回保管した状態に戻す(⊻)	適用( <u>Y</u> )
?	実行(图)	閉じる

# (3) 実行結果がコンソールに表示されます.

C hello/helloc - CYpleiades/Workspace - Eclipse - ヴ ファイル(F) 標準(F)、ソース(S)、リファクタリッグ(D、サンゲート(N)、株の(A)、プロジェクト(P)、単行(R)、ウィンドウ(N)、ヘルブ(H)						
[] + [] () + ¶ () + ¶ () + () + () + () + () +	クイック・アクセス 🔹 🖬 C/C++					
26 70 9/27 1×127 70 − 5+ ∞ 😑 🧐 🐷 🗢 🗖 🗋 Mellocc ∞	□ 🞥 79 28 🗐 97 ® 9 " □					
• (a) 197/197/197/197/197/197/197/197/197/197/	Image: Plan: R     Image: Plan:					
Hello Norld	^					
c	~ ~ ~					
UTF-8 CRLF C /hello/hello.c						

- 5. プログラムを書き換えた後のビルド
- (1) 誤り等を修正した場合や、ソースファイルのプログラムを書き換えた場合は、「保管」 「すべてビルド」の手順に従います.しかし、操作が煩雑になるので、「実行」をク リックすると良いでしょう.

hello/hello.c - C:¥pleiades¥workspace - Eclipse

ファイル(F) 編集(E) ソース(S) リファクタリング(T) 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) 📑 + 🔚 🐚 🛞 + 🗞 + 📾 🕘 + 📃 🔌 😳 💣 + 🚳 + 💽 + 🎯 + 🏘 + 🚺 🤂 🖓 + 🤬 + 🖉 実行 陷 プロジェクト・エクスプローラー 🛛 📄 🔄 🌍 🗢 🗖 🗖 💼 \*hello.c 🔀 1⊝/\*.. 🗸 📂 hello \* hello.c≓ 23 > 🗊 インクルード - 8-1 \* Created on: 2017/09/04-4 > 📂 Debug 5 Author: dohi₊ ж > 💼 hello.c \*/... 6 8 **‡include** ≺stdio.h>₊ - 9 10 11⊖ int main(yoid){→ printf("Hello World¥n");4 return 0;₊ } 15

(2) 「常に起動前にリソースを保管する」をチェックし、「OK」をクリックします.

⊜ 保管して起動			$\times$
保管するリソースを選択して	てください:		
hello.c			
すべて選択(	<u>S</u> ) 選択を	すべて解除	余( <u>D</u> )
☑││に起動前にリソースを	保管する( <u>R</u> )		
?	ок	キャンセ	μ

(3) エラーが無ければ「保管」「すべてビルド」「実行」が自動的に行われます.エラー がある場合は、「保管」「すべてビルド」まで行われます.エラーを修正し、「実行」 を行います.今後は、エラーの原因を早く見つけられるようになることが重要です.