2016/03/07

NetBeansではじめるJava

第一回 はじめてのソフトウェア開発

ArkOak 代表 加納 徹

Java講習会のねらい



・ プログラムが動くのを楽しむ

● ソフトウェア開発を身近に感じる

作る楽しさを知れば、学ぶのが楽しくなる!

Java 講習 会の 流れ



1. 開発環境の構築



- 1. JDK (Java SE Development Kit)
 - Javaのプログラミングに必要な基本道具セット

2. NetBeans

- Java公式の統合開発環境(IDE)
- 優れたGUIエディタを搭載
- 3. Javaドキュメント
 - Javaの便利な取扱説明書

英語版のJavaドキュメントは最初から入っているみたい



JDK/NetBeansのインストール①

- 1. <u>http://www.oracle.com/technetwork/java</u> /javase/downloads/index.html にアクセス
- 2. 「NetBeans with JDK 8」をクリック

Overview Downloads	Documentation	Community	Technologies	Training
Java SE Downloa	ads			
Java Platform	Java ~ NLOAD ± (JDK) 8u73 / 8u74		DownLo NetBeans w	Beans DAD ± vith JDK 8
	Java Platfor	m, Standard	Edition	
Java SE 8u73 / 8u74 Java SE 8u73 includes i users upgrade to this rel additional features (deso Learn more ►	1 mportant security fi ease. Java SE 8u7 cribed in the releas	xes. Oracle stro 4 is a patch-set e notes).	ngly recommends update, including	that all Java SE 8 all of 8u73 plus

JDK/NetBeansのインストール②

- 3. 「Accept License Agreement」をチェック
- 4. 各自の環境に合ったものをダウンロード

verview Downloads [Documentation C	community Technologies Training
JDK 8u73 with Net This distribution of the JDK in integrated development env	tBeans 8.1 ncludes the Java SE ironment for developi	bundle of NetBeans IDE, which is a powerful bing applications on the Java platform. Learn more
You must accept the J download this softwar	IDK 8u73 and NetBeare. nse Agreement	ans 8.1 Cobundle License Agreement to Decline License Agreement
Java SE ar	d NetBeans Cob	oundle (JDK 8u73 and NB 8.1)
Product / File Descripti	ion File Size	Download
Linux x86	281.84 MB	tidk-8u73-nb-8 1-linux-i586.sh
Linux x64	277.82 MB	
Mac OS X	340.95 MB	jdk-8u73-nb-8_1-macosx-x64.dmg
Windows x86	298.3 MB	idk-8u73-nb-8 1-windows-i586 exe
Windows x64	305.27 MB	jdk-8u73-nb-8_1-windows-x64.exe

JDK/NetBeansのインストール③

1. 「次へ」を押す

🙆 Java SE Development Kitと	NetBeans IDEインストーラ	-		×
	JDK 8 Update 73とNetBeans IDE 8.1インフ	ストーラへ。	kえてそ	
	このインストーラは、JDK 8 Update 73とNetBeans IDEを化	ンストールしま	す。	
(Java [®]				•
ORACLE	1	んストール・	サイズ: 778	13 MB
	[次(N) >	Į R	消





2. 「次へ」を押す



5. しばらく待つ



3. 「次へ」を押す

	🂰 Java SE	Development KitとNetBeans IDEインストーラ	-	
NetBensi DE0/1/2/~1/8(0. CVFroeran FilesWetBears 81 NetBensi DEF用のJDK(mX(J) CVFroeran FilesWetBears 81 ・「更更新の確認」のチェッグ ・「更新の確認」のチェッグ ・「Jose St Development Kit2/NetBears DE1/22/~3 ・」 Jose St Development Kit2/NetBears DE1/22/~3 ・ 「 アントー ・ 「 ノンス ー // ショルタッ・ ・ 「 ノンス ー // ソノン // ショー ・ 「 ノンス ー // ソノン // ショー ・ 「 ノンス ー // ソノン // ジョー ・ 「 アンス ー // アノン // ジョー ・ // アンス // ジョー ・ // アンス /	Java [®]	NetBeans IDE 8.1のインストール インストール・フォルダとJDK(tm)を選択します。		ORACI
Cv¥Program Files¥MetBare 81 使気(4) (m(3)	NetBeans	IDEのインストール先(I):		
NetBeare IDE時0,JDK(m)(J) ©¥Program Files¥Java¥jsk180,73 ● ● (次の) (次の) (1) (夏&(0) (次の) (1) (夏&(0) (次の) (1) (1) (1) (1) (1) (1) (1) (1)	C:¥Progra	am Files¥NetBeans 8.1		参照(R).
文学のcran Filest Javairjs 118.73 (東広田) 次加2 (東広田) 次加2 (東広田) 次加2 (東京田) (東西) (東西) (東西) (東西) (東西) (東西) () (letBeans	IDE用のJDK(tm)(J):		
(夏60) 次(1)2 10 (夏60) 次(1)2 「 更新の確認」のチェッグ トして「インストール」を拥 Java SE Development Kitと NetBeans IDE (1/2)トーラ - ジャック 「 ないって、 「 ないって、 「 して、 」 した。 「 した。 」 した。 「 した。 」 した。 した。 」 した。 」 した。 」 した。 した。 」 した。 」 した。 した。 した。 」 した。 」 した。 」 した。 した				
・「更新の確認」のチェック トして「インストール」を押 Java SE Development Kitz Netileans IDE1/2k-9 - □ Java SE Development Kitz Netileans IDE1/2k-9 - □ Java SE Development Kitz Netileans IDE1/2k-9 - □ Java SE Development Kitz Netileans IDE1/2k-9 Java SE Development IDE			< 戻る(B) 次(N) > 取消
Loc 「インストール」を Ava SE Development Kitと NetBeans IDE (ソストーラ				
 Java B Development Kitz NetBeans IDE (ジストーラ) – □ プマノー プマノー プマノー プマノー プマノー プマノー JDK 8 Update 73 (シストール・フォルダー CAProgram Files/Vayavárdí 180,73 NetBeans IDE (* ノストール・フォルダー CAProgram Files/Vayavárdí 180,73 Net Teams B1 LE EDGAUET オ, インターネット機械を使用して、インストール・決力プラグインの更新がないかを インストール・サイズ会社・ アムストール・サイズ会社・ アメント アメント ノンス トール・サイズ 会社・ アメント アメント		「更新の確認」	- のチェ	ック
JDK 8 Update 79インストール・フォリダ CVProgram FileskJavvýják 13.0_78 NetBaens IDEインストール・フォリルS CCProgram FileskWetBaens 8.1 正変更的の確認ない またシークーネット接続を使用して、インストール済のプラヴインの更新がないかを 含体的にまたりできます。 インストール・サイズを含ま 78.3 MB	'トし	「更新の確認」 って「インス」	のチェ \ール」 [;]	ック を押
C4Program File¥/Javvájdk 18.0,73 NetBana IDEインストール・フォリルボ C3Program File¥/Helsant 8.1 「更新の確認とい 自動ロジェオックで表す。 インストール・サイズ会計: 778.3 MB	∙ ↓ Java SE	「更新の確認」 て「インス」 Development Kitz NetBears IDE1/3ストーラ 1977- 17-3ストールは2019のにインストールを開始にます。	のチェ \-ル」 [:]	ック を押
NetBaers DEインストール・フリルテ CS4Program FUEWAREBane 8.1 ・ 更新の確認(い) も、クレクーネット機能を使用して、インストール液のプラグインの更新がないかを 自動的に支まっかできす。 インストール・サイズ合計: 778.3 MB	I Java SE	「更新の確認」 て「インス」 Development Kitz NetBeans IDE インストーラ 1722トールはをりゅうしてインストールを開始します。	のチェ \-ル」 [;]	ック を押 。
□ 更新の確認い しま、 インターネット接続を使用して、 インストール 泳のプラヴインの更新がないかを 自動的に手っかできます。 インストール サイズ合計: 778.3 MB	Java SE	「更新の確認。 ててインスー Provent Kitz NetBeans IDE 1/2ストーラ 1727 - 「クストールはをりゅうしてインストールを開始します。 dete 734/2ストール-7241/5* eram Fiber Haverbick 182,73	のチェ \-ル」 [;]	ック を押 。
インストール・サイズ合計: 778.3 MB	JDK 8 Up C×¥Pro NetBeans C×¥Pro	「更新の確認。 「更新の確認。 して「インスー」 Development Kit NetBeans IDEインストーラ サマリー 「クンストール」を知らしてインストールを開始します。 ロロインストールーフォルダ・ eram Fileskit Javavijski 180,78 IDEインストールーフォルダ・ eram Fileskit Javavijski 180,78 IDEインストールーフォルダ・ eram Fileskit Javavijski 180,78	のチェ \-ル」 [:]	ック を押 。
	Java SE Java SE Java JDK 8 Up C+¥Pro NetBeans C+¥Pro	「更新の確認。」 したいとののののである したいとのののののののののののののののののののののののののののののののののののの	しのチェ 〜ール」。 ー - い泳のフラクイレの運転がないかを	ック を押 。
	・ よ し よ し よ し よ し よ し ま し こ 半 た で し 、 半 わ で し こ 半 た の こ 、 半 た ち し こ 半 た の こ 半 た の こ 半 た の こ 半 た の こ 半 た の こ 半 た の こ 半 た の こ ま あ ち い こ 半 た の こ ま あ ち い こ よ し こ ま あ ち こ よ ち っ こ よ ち っ こ よ ち っ し こ ま あ ち こ よ ち っ し こ ま ち こ 、 ま し 、 ち し こ ち こ 、 ち し 、 ち し 、 ち し こ ち こ 、 ち し 、 ち し 、 ち し 、 ち し 、 ち し 、 ち 一 、 ち し 、 ち 一 こ ち 一 、 ち 一 こ ー ち 一 、 し 、 ち 一 、 し 、 し 、 し 一 、 し 一 、 し 一 う し 、 し 一 、 し 一 、 し 一 、 し 一 、 し 一 、 し 一 し 一 し 一 し 一 、 一 、 一 し 一 、 一 、 し 、 、 し 、 し 、 し 、 し 、 し 、 し 、 つ 、 し 、 し 、 、 、 し 、 し 、 、 、 し 、 、 、 、 、 、 、 、 、 、 、 、	「更新の確認。 「 「 「	のチェ 〜ール」; ー -ル:家のブラグインの変新がないかを	ック を押

次(N) > 取消

Javaの特徴

● 世界で最も使われているプログラミング言語

- C言語から派生(文法が似てる)
- GUIソフトウェアが作りやすい

CUI Character User Interface

% cd /bin				
¥ la				
Î*	dd*	kill*	n#d*	supc*
hash*	df*	ln*	pen*	tcsh*
cat*	domainname*	1.5*	ra*	test*
chmod*	echo*	mkdin*	radir*	zsh*
cn*	ed*	mu*	sh*	zsh=4.1.1*
csh*	expn*	nav*	sleen*	2011 11111
date*	hostname*	ns*	sttu*	
% nud	nootnamo	P 0	0119	
Zhin				
% od /ten				
¥ ev foo eb				
"foo eb" [Nem Fi	[ما ا			
Entening Ex mode	Tupe "uieual'	to ao to Nonwal	ande.	
in a second second	s. igpo visuai	to go to normal	liode.	
ogt * tyt ! eoni	-pp luipg -c l	bead		
cut stext soli		lieuu		
:				
"foo sh" [New] :	2 30C moitten			
1001011 [1100] 1	LE, oso an recom			
X ls - l foo.sh				
-nw-nn 1 u	nid wheel 39.20	Sen 02.50 foo	ab.	
* chaod +v IS		, sep ez:se iee.		
8				

GUI Graphical User Interface



Javaの特徴

- 世界で最も使われているプログラミング言語
- C言語から派生(文法が似てる)
- GUIソフトウェアが作りやすい
- マルチプラットフォーム(どんなPCでも動く)
- ライブラリが豊富(便利な道具がいっぱい)
- オブジェクト指向(効率的で安全)
- だいたい何でもできる



● 電気製品(携帯電話、電子レンジ)

● 業務/研究アプリケーション

● Webアプリケーション

• スーパーコンピュータ

Java無しではもう生きていけない





● 変数

- 数値や文字などを入れておく入れ物
- メソッド
 - 一連の処理のまとまり (C言語の関数に相当)
 - プログラムはmainメソッドから始まる (C言語のmain関数に相当)



- 変数やメソッドのまとまり
- プログラムには必ずmainクラスが一つ存在する
- プログラムはmainクラスのmainメソッドから始まる

全ての変数とメソッドは、必ずクラスに属している

NetBeansの起動

1. アイコンをクリック



3. データ提供は「いいえ」を選択











Javaドキュメントのダウンロード

- 1. <u>http://www.oracle.com/technetwork/jp/java/java-sun-</u> <u>1440465-ja.html</u> にアクセス
- 2. 最新バージョンのファイルをダウンロード
- 3. そのまま「C://Program Files/Java/」等の場所に入れる



Javaドキュメントの組み込み

4. [ツール] ⇔ [Javaプラットフォーム] と進む



Javaドキュメントの組み込み

5. [Javadoc] タブを選択し、 [ZIP/フォルダの追加] ボタンを押す

		×
「Javadoc」タブを使用して、JDKのAPIドキュ 「ブラットフォームを追加」をクリックして、他のJ	メントをIDEに登録します。 avaブラットフォーム・バージョンを登録します。	
プラットフォーム(P):		
Java SE	プラットフォーム名:	
	Ci#Program Files#Java#jdk1.8.0_73	
	クラス ソース Javadoc	
	プラットフォーム Javadoc:	
	https://docs.oracle.com/javase/8/docs/api/	ZIP/フォルダの追加…
	https://docs.oracle.com/javase/8/javafx/api/	LIRLの追加
		除去
		1.2.5.605
		上へ移動
		下八移動
		1 (19/22)
プラットフォームの追加 除去		
		閉じる ヘルプ(H)

Javaドキュメントの組み込み

6. [上へ移動] ボタンを押して、 一番上まで移動させる

🗊 Javaプラットフォーム・マネージャ		×
「Javadoc」タブを使用して、JDKのAPIドキュ 「プラットフォームを追加」をクリックして、他のJ	」メントをIDEに登録します。 lavaブラットフォーム・バージョンを登録します。	
「フラットフォームを注加」をジリックして、他のJ プラットフォーム(P): ↓ Java SE ↓ ■ JDK 1.8(デフォルト)	オンラットフォームや: JDK 1.8(デフォルト) プラットフォーム・フォルダ: C*Program Files*Java*jdk1.8.0_73 クラス、ソース Javadoc プラットフォームJavadoc: jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/attach/spec/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javac/tree/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jre/api/javac/tree/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jre/api/plugin/dom/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/joda/jdi/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ jarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javadoc/doclet/ 上へ移動 iarfile:/C:/Program%20Files/Java/8zipl/E54527_01/jdk/api/javaf 下へ移動	連打
プラットフォームの追加 除去		
		-1)

2. 基本的なプログラミング



プロジェクトの作成

1. [新規プロジェクト...] ボタンを押す



3. プロジェクト名(Workshop1)をつけて、 [終了] ボタンを押す

<u>ステップ</u>	名前と場所		
1. フロジェクトを選択 2. 名前と場所	プロジェクト名(N):	Workshop 1	
	プロジェクトの場所(L):	C:¥Users¥Toru-ITX¥Documents¥NetBeansProjects	参照(0)
	プロジェクト・フォルダ(D):	$\verb+ ers \texttt{*} Toru-\Pi X \texttt{*} Documents \texttt{*} Net Beans Projects \texttt{*} Workshop1$	
	□ ライブラリの格納用に	:専用フォルダを使用(U)	
	ライブラリ・フォルダ(1):		参照(W)
		複数のユーザーとブロジェクトが同じコンパイル・ライブラリを共有 できます(詳細はヘルブを参照)。	
確認	2メイン・クラスの作成(C) workshop 1.Workshop 1	

プロジェクトの作成



フォント変更のすすめ

-Д(M)	ッール(T) ウィンドウ(W) ヘルプ(H) 🔍	検
•	差分パッチを適用(Y)	
	差分(F)	
	お気に入りに追加(R)	
	ターミナルで開く	
choos	Javadocを解析(A)	t P
choose	パレットに追加(E)	
ditor.	テストの作成/更新	
	国際化(Z) >	
	Javaプラットフォーム(J)	
	NetBeansプラットフォーム(N)	
	Ant変数(V)	
	ライブラリ(L)	
	サーバー(S)	
	クラウド・プロバイダ(C)	
ine ar	テンプレート(T)	
	DTDおよびXMLスキーマ(D)	
ng[] a logic	/(レット(P) >	
	プラグイン(G)	
	オプション(0)	

1. [ツール] ⇒ [オプション] と進む 2. [フォントと色] タブでフォントを変更する

 オプション デー般(G) エディタ(E) エディタ(E) ステントと色(F) キーマップ(K) Java 	×
プロファイル(P): NetBeans 構文(S) 強調表示(L) 注釈(N) 差分(E) バージョン管理 言語(U): すべての言語 〜	✓ 複製(D) 復元(R)
カテコリ(G): デフォルト URL エラー エンティティ参照 キーワード コメント セパレータ フィールド マークレスコーデ プレビュー:	 フォンド(0): メイリオ 18 前景(E): 単黒 ず景(B): □白 小 効果: なし ◇ 効果:
/** * Comment. */	~
エクスポード(T) インポード(D	OK 適用 取消 ヘルプ(H)

フォント変更のすすめ

3. [その他] タブの [出力] タブでフォントを変更する

🗊 গ্ৰন্ট্য					×
② ② ② ③	 ・ ・ ・	♪観 テ	1 その他(M)		Q ७२७४७(Ctrl+F)
ターミナル ファイル 差分	出力				
出力ウィンドウ設定					へ 復元(R)
フォンKF):	メイリオ 14 (折り返されているテキストには適用されません)				
フォント・サイズ(Z):	14				
背景色(B):		\sim	デバッグ:	■グレー	~
標準色(C):	■黒	\sim	警告:	□ オレンジ	~
エラーの色(E):	[191,0,0]	\sim	失敗:	■赤	~
入力色(1)	■黒	\sim	成功:	[0,124,0]	~
リンク色(L):	[0,0,191]	\sim			
重要リンクの色(P)	[0,0,191]	\sim			
リンク・スタイル(Y)	下線	\sim			
Standard Lin Tmportant Lin	K nk				^ v
エクスポード(T)	インボードロ		ОК	適用	取消 ヘルプ(H)

プロジェクト作成のすすめ

時々左上にこんなものを表示します





2.1 文字の出力

Hello NetBeans!

(A

既存プロジェクト

「Workshop1」 を使おう



「System.out.println("Hello NetBeans!");」と書いてみよう

Hello NetBeans!



[Ctrl] + [スペース] で強制的に候補を出すこともできる

Hello NetBeans!

♥ Workshop1 - NetBeans IDE 8.1 ファイル(E) 編集(E) 表示(Y) ナビゲート(N) ソー	ふりリファクタリング@実行®デバッグ®ファファ	Š
🛍 🛅 블 🖣 🦻 🦿 🕼		
□ プロジェクト × –	開始ページ × 國 Workshop1java ×	$\leftrightarrow \bullet \bullet \Box$
🗧 🖃 🧶 Workshop 1	ソース 履歴 🞯 👼 - 🦏 - 🔍 🖓 🖓 📇 📪 🔗 😓 🖄 🗐 🥥 🥥 📑 🖉 🚅	B€
下 単価 ソース・パッケージ 一 単一価 Workshop1	12 public class Workshop1 {	^ =
- Workshop 1. java	13	
⊞… 🎽 ライブラリ	14 📮 /**	
	15 * @param args the command line arguments	
	17 public static void <i>main</i> (String[] args) {	
	18 System.out.println("Hello NetBeans!");	
	20	
main - ナビゲータ× ー		
E- & Workshop1	22 3 出力ワインドワ	
🥼 🥼 main(String 🛛 ares)		>
	🖄 workshop1.Workshop1 义 🍈 main 🔪	×
	出力 - Workshop1 (run) ×	-
	run:	
	Hello NetBeans!	
	ビルド成功(合計時間: 0秒)	
		~
		INS

出力ウィンドウにHello NetBeans! と表示されたら成功!





データの入れ物「変数」

変数とは

[データ型][変数名];

の形で宣言される、数値や文字を入れておく入れ物です

データ型	種類	サイズ	扱える範囲
short	整数	2バイト	-32786~32767
int	整数	4バイト	-2147483648~2147483647
long	整数	8バイト	-9223372036854775808~9223372036854775807
float	浮動小数点数	4バイト	有効数字7桁 ±10 ⁻³⁸ ~10 ³⁸
double	浮動小数点数	8バイト	有効数字15桁 ±10 ⁻³⁰⁸ ~10 ³⁰⁸
char	文字定数	1バイト	Unicode文字(一文字)
String	文字列	不定	大体どんだけでも
boolean	論理値	1バイト	true, false

データの入れ物「変数」

変数とは

[データ型][変数名];

の形で宣言される、数値や文字を入れておく入れ物です

データ型	種類	サイズ	扱える範囲
int	整数	4バイト	-2147483648~2147483647
double	浮動小数点数	8バイト	有効数字15桁 ±10 ⁻³⁰⁸ ~10 ³⁰⁸
String	文字列	不定	大体どんだけでも
boolean	論理値	1バイト	true, false





変数の使い方





色々な計算



2.3 if文

if文





if文

2つの値を比較するときは、**比較演算子**を用います





2.4 for文


for文

for (初期化;評価式;次の一歩) { 処理 }

の形で記述され、次の流れで実行されます

- 1. 初期化を行う
- 2. 評価式が正しければ、処理を実行する(違えば終了)
- 3. 処理が終わったら、次の一歩を行う
- 4. 手順2に戻る

```
public static void main(String[] args) {
    int sum = 0;
    for (int n = 1; n <= 10; n++) {
        sum = sum + n;
     }
    System.out.println("1から10までの和は, " + sum);
}</pre>
```







for文とif文を使い、九九の表を出力しなさい 一桁の数字の前には 0 を挿入し、綺麗に揃えなさい

ヒント

print()メソッドを使うと改行されない for文の中にfor文を入れることができる どこで改行を入れるかがミソ 数字が一桁かどうかはif文で判定する



実行例



NetBeansの 便利な 機能

ナビゲート	>
Javadocを表示	Alt+F1
使用状況を検索	Alt+F7
呼出し階層	
コードを挿入	Alt+Insert
インポートを修正	Ctrl+Shift+I
リファクタリング	>
フォーマット	Alt+Shift+F
ファイルの実行	Shift+F6
ファイルをデバッグ	Ctrl+Shift+F5
ファイルをテスト	Ctrl+F6
ファイルのテストをデバッグ	Ctrl+Shift+F6
フォーカスしたテスト・メソッドの実行	ī
フォーカスしたテスト・メソッドのデバッ	ッグ
メソッドまで実行(M)	
新規ウォッチ	Ctrl+Shift+F7
行ブレークポイントを切替え(T)	Ctrl+F8
プロファイル	>
切取り(T)	Ctrl+X
コピー(Y)	Ctrl+C
貼付け	Ctrl+V
コード折りたたみ(の)	>
	-

3. ソフトウェアの作り方

新規プロジェクト 「**Software1**」 を作ろう

(D

プロジェクトの作成

ステップ	名前と場所		
1. プロジェクトを選択 2. 名前と場所	プロジェクト名(N):	Software 1	
	プロジェクトの場所(L):	C:¥Users¥Toru-ITX¥Documents¥NetBeansProjects	参照(0)
	プロジェクト・フォルダ(D):	:= ers F Toru-IT X Y Documents Y Net Beans Projects Y Software 1	
	□ ライブラルの核納用()	- 東田フォルガを使用(1))	
	ライブラリ・フォルダ(D)		参照(W)
		複数のユーザーとプロジェクトが同じコンパイル・ライブラリを共有	2-7000-0700
		できます(詳細はヘルプを参照)。	
	□メイックラフの作成((C) software 1 Software 1	
		チェッ ク を外す	
		チェックを外す	
		チェッ クを外す	
		チェッ クを外す	
		チェックを外す	

今回、メインクラスは自分で作っていく



🗊 New JFrameフォーム		名前をつける(FrameMain) ×	
<u>ステッフ</u>	名前と場所		
1. ファイル・タイプを選択 2. 名前と場所	クラス名(N): Frame	Main	
	プロジェクド(P):	Software 1	
	場所(L):	ソース・パッケージ 〜	
	パッケ <i>ージ(K)</i> :	gui 🗸 🗸	
	作成されるファイル(C	i): pru-ITX¥Documents¥NetBeansProjects¥Software1¥src¥gui¥FrameMain.java	
		適当にパッケージ名をつける(gui)	
		< 戻る(B) 次 > 終了(F) 取消 ヘルプ(H)	STID.
今後のため	かに、パッ	ケージ名をつけておくと吉	

ソースコードとGUI	デザイナを切り替えられる ^{M ッ-ルの ゥィッド}	GUI部品のパレット
日 フロジェクト× - Software1 ・ Software1 ・ Software1 eui FrameMain java ・ FrameMain java FrameMain java	開始ペジ × ■ FrameMain.java × ソース デザイン 履歴 ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	パレット× - Swineコンテナ パネル タブ付きペイン ゴ 分割ペイン コクロール・ペイン ツールバー デスクトッブ・ペイン 内部フレーム レイヤー・ペイン Swineコントロール レイヤー・ペイン Swineコントロール レイヤー・ペイン ラベル 図 ボタン ラベル 図 ボタン ラベル ア チェック・ボックス コンボ・ボックス ゴ リスト テキスト・フィールド テキスト/領域 ゴ ンガ・ボックス ゴ リスト フカール・バー スライダ プロパティ バイント プロパティ バイント alwaysOnTop コー alwaysOnTopSuported コー alwaysOnTopSuported コー alwaysOnTop コー ロのちロバティ ジ ロバティ alwaysOnTop コー alwaysOnTop コー alwaysOnTop コー コレー ジ 「JFrame] ?

Software1 - NetBeans IDE 8.1		- 🗆	×
ファイル(E) 編集(E) 表示(V) ナビゲート(N) ソース(S) リファクタリング(A) 実行(R) デバッグ(D) プロファイル(P) チーム(M) ツール(T) ウィンド	ウ(₩) ヘルプ(円) 🔍 検察	춖 (Ctrl+1)	
*** *** **********************************			
プロジェクト × - ● ● Software 1 >> ● ● Y-ス・パッケージ ● ● Given ● ● FrameMain.java >> ● ● ● FrameMain.java >> ● ● ● ● ● ● FrameMain.java >> ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	 パレット× ブベル ブベル ブベル ブベル ブベル ブジオ・ボックス テキスト・フィールド スクロール・バー 進捗バー パスワード・フィールド セパレータ エディタ・ペイン 表 Smine < 他のコンボーネント - 1 	 本タン エタン・ボックス ボタン・グループ リスト テキスト領域 スライダ オ式付きフィールト スピナー テキスト・ペイン ツリー 	
フレームのサイズをマウスドラッグで変更	<70/4	ティなし>	
	他のコンポーネント		0
	Q	1:1	INS

メインクラスに「FrameMain」を指定

לפל 🎧	ェクトの実行	×
1	プロジェクトSoftware1にはメイン・クラスが設定されていません。	
	メイン・クラスを選択(S): cui Franco Main	
	gulframeman	
	OK 取消	¥.

GUIソフトウェアの土台が完成!







パレット ×	-
🗆 Swingコンテナ	
🔲 パネル	🚞 タブ付きペイン
Ⅲ 分割ペイン	🔜 スクロール・ペイン
💷 ツールバー	🖻 デスクトップ・ペイン
🛄 内部フレーム	💽 レイヤー・ペイン
□ Swingコントロール	
label ラベル	📧 ボタン
◎ トグル・ボタン	፼一 チェック・ボックス
◎ー ラジオ・ボタン	8= ボタン・グループ
📧 コンボ・ボックス	<u>=</u> リスト
💷 テキスト・フィールド	t× テキスト領域
💷 スクロール・バー	ѿ ス ラ イダ
💷 進捗バー	🖅 書式付きフィールド
📼 パスワード・フィールド	💷 スピナー
─ セパレータ	🔳 テキスト・ペイン
🔤 エディタ・ペイン	记 ツリー
■ 表	

ArkOakホームページに使い方が載っています http://arkoak.com/java/

第1章 ソフトウェア開発の準備

- 1.1 いきなりソフトウェア開発? (読み飛ばしてOK!)
- 1.2 Java開発環境のインストール

第2章 いざ、ソフトウェア開発!

- 2.1 ソフトウェアの画面作りとルック・アンド・フィール
- 2.2 ボタン (JButton) の使い方
- 2.3 ラベル (JLabel) の使い方
- 2.4 チェックボックス (JCheckCox)の使い方
- 2.5 ラジオボタン (JRadioButton) とボタングループ (ButtonGroup) の使い方
- 2.6 コンボボックス (JComboBox)の使い方
- 2.7 テキストフィールド (JTextField) の使い方
- 2.8 テキスト領域(JTextArea)の使い方
- 2.9 スピナー (JSpinner)の使い方

2.10 スライダ (JSlider) の使い方 🎔

重要なものを幾つかピックアップして説明するよ

3.1 ボタンの使い方

GUI名: JButton 機能を持った押しボタン

既存プロジェクト ボタンの使い方 Software1 を使おう



名前の付け方のすすめ



「例」 ● 開始ボタン

- アニメ開始ボタン
- 終了ボタン
- 描画ラベル
- 設定コンボボックス

- ⇒ btnStart
- ⇒ btnStartAnimation
- ⇒ buttonEnd
- ⇒ IbIDraw
- ⇒ cmbSetting
- 出力用テキスト領域 ⇒ taOutput

変数名の頭は小文字、単語の切れ目で大文字にする習慣



ボタンの使い方

7. ボタンをダブルクリックしてソースコードを記述

private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {
 System.out.println("こんにちは!");
}



出力画面は開発者にしか見ることができないんだよね~

ボタンの使い方

10. ボタンの処理を以下のように書き換えてみましょう

private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {
 btnHello.setText("こんにちは!");
}



3.2 ラベルの使い方

GUI名:JLabel 編集不可能なテキスト領域

既存プロジェクト ボタンの使い方 Software1 を使おう



ラベルの使い方

7. ボタンのダブルクリックしてソースコードを記述



ラベルの使い方

10. ボタンを押した回数を表示するようにソースコードを変更

11. 実行結果

```
int count = 0;
private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {
    count = count + 1;
    lblHello.setText(count + "回目のこんにちは!");
}
```



ラベルの使い方

12. ボタンを押した回数が10回以上になったとき、違う処理をさせてみよう

```
int count = 0;
private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {
    count = count + 1;
    if (count < 10) {
        IblHello.setText(count + "回目のこんにちは!");
    } else {
        IblHello.setText("もういいよ・・・");
    }
}</pre>
```



3.3 チェックボックスの使い方

GUI名: JCheckBox 選択・選択解除可能な項目



チェックボックスの使い方

1. 次のように、ボタン、チェックボックス、ラベルを配置



チェックボックスの使い方

2. ボタンをダブルクリックして、何倍返しかをラベルに表示するソースコードを記述

privat	e void btnRevengeActionPerformed	l(ja che	ckMulti2 がチェックされていたら 変数 revenge を2倍にする
	int revenge = 1;		
	if (checkMulti2.isSelected()) reven	ge *= 2;	
	if (checkMulti5.isSelected()) reven	ge *= 5;	
	<pre>if (checkMulti10.isSelected()) reven</pre>	ge *= 10	;
	lblRevenge.setText(revenge + "倍返し	ノだ!");	
}		if3	なの中の処理が1行のときに限り、 中括弧を省略することができる

4	—		×
	やり返す	ब 🏻	6
□ 2倍	□ 5倍	□ 1	.0倍
1	旧伝返した	≝!	

<u></u>	_		×
	やり返	ब [¢
☑ 2倍	□ 5倍	□ 1	0倍
2	倍返した	ビ!	

4	_		Х
	やり返す	đ	6
☑ 2倍	☑ 5倍	☑ 1	.0倍
10)0倍返し	だ!	

メンバ変数とローカル変数

3. メンバ変数を使ったプログラムに変更して、動作の違いを確認してみよう



- メンバ変数(フィールド)
 - クラス内(メソッド外)で宣言した変数
 - クラスのどこからでも利用可能で、値は保持される

● □ーカル変数

- メソッド内で宣言した変数
- メソッド内でのみ利用可能で、メソッド処理終了時に値は破棄される

3.4 テキストフィールド/ テキストエリアの使い方

GUI名:JTextField・・・1行のテキスト編集領域 **GUI名:JTextArea**・・・複数行のテキスト編集領域



1. ラベル、ボタン、テキストフィールド、テキストエリアを配置





2. ボタンをダブルクリックして、ソースコードを記述

private void **btnHelloActionPerformed**(java.awt.event.ActionEvent evt) {
 String name = txtName.getText();
 taOutput.append(name + "さん、こんにちは!¥n");
}

		(*)	8.実行		Ę		
<u>لا</u>					_		×
名前	ちくわ					挨拶	ಕೃತ
あ- お- ちく	-くさん、 -くさん、 こわさん、	こん こん こん	こちは! こちは! こちは!				

3.5 スピナの使い方

GUI名:JSpinner 要素間をクリックで移動可能な入力フィールド



スピナの使い方

1. ラベル、スピナを配置



2. スピナを右クリック ⇔ [プロパティ] ⇔ [model] 横の […]ボタンをクリック

🜍 spYear [JSpinner] - プロパティ	×
プロパティ バインド イベント コー	K.
□ プロパティ	^
font	メイリオ 14 プレーン …
model	[SpinnerModel]
toolTipText	
⊡他のプロパティ	
UI	くデフォル 🗸 🖉 🛄
UIClassID	Spinner クリック …
alignmentX	0.5
alignmentY	0.5
autoscrolls	
baselineResizeBehavior	CENTER_OFFSET
border	[CompoundBorderUIResource]
componentPopupMenu	〈なし〉 🔍 🔍
cursor	デフォルトのカーソル 🔻 📖
debugGraphicsOptions	NO_CHANGES 🛛 🗸 🛄
doubleBuffered	
editor	デフォルト …
enabled	✓ …
focusCycleRoot	
spYear [JSpinner]	3



スピナの使い方

3. スピナのモデルを以下のように変更



スピナの使い方

4. スピナから数値を取得して、年齢を算出するコードを記述

<pre>private void btnHelloActionPerformed(java.awt.event.ActionEvent evt) {</pre>
String name = txtName.getText();
<pre>int age = 2016 - (Integer) spYear.getValue();</pre>
taOutput.append(age + "歳の" + name + "さん、こんにちは!¥n");
}

	Ę	5. 実行結	果	
	<u></u>		_	
:	名前ちくわ	生まれ年	2,013 🔹	挨拶する
	309歳のオイラーさ 3歳のちくわさん、	さん、こんにちに こんにちは!	t !	

3.6 Look and Feel



Look and Feel

ソフトウェアの見た目(Look)と感じ方(Feel)を変更してみましょう





Look and Feel

114	// <editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) "></editor-fold>
115	/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
116	* For details see <u>http://dow</u> "Nimbus"の文字を変更 <u>ookandfeel/plaf.html</u>
117	*/
118	try {
119	for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
120	if ("Nimbus".equals(info.getName())) {
121	javax.swing.UIManager.setLookAndFeel(info.getClassName());
122	break;
123	}
124	}
8	<pre>} catch (ClassNotFoundException ex) {</pre>
126	java.util.logging.Logger.getLogger(FrameMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
127	<pre>} catch (InstantiationException ex) {</pre>
128	java.util.logging.Logger.getLogger(FrameMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
129	<pre>} catch (IllegalAccessException ex) {</pre>
130	java.util.logging.Logger.getLogger(FrameMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
131	<pre>} catch (javax.swing.UnsupportedLookAndFeelException ex) {</pre>
132	java.util.logging.Logger.getLogger(FrameMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
133	}
134	- //

Look and Feel


4. お絵かきソフトウェアの開発

Javaでお絵描きする流れ







1. 以下のようにラベルとボタンを配置



Javaでお絵描き

2. 以下のソースコードを記述

幅512, 高さ512の RGB描画領域を作成



Points!

描画領域にBufferedImageクラスを使うことができる

描画命令はGraphicsクラスに搭載されている

描画結果は**ラベル**に貼り付けることができる





Javaでお絵描き

3. 以下のソースコードを追加で記述

g.drawRect(100, 60, 310, 200);

g.drawRect(150, 100, 60, 20);

g.drawOval(150, 150, 60, 30);

g.fillRect(300, 100, 60, 20);

g.fillOval(300, 150, 60, 30);

g.drawLine(200, 210, 255, 240);

g.drawLine(255, 240, 310, 210);

Points!

短形の描画は drawRect(x, y, width, height) 楕円の描画は drawOval(x, y, width, height) 矩形の塗り潰しは fillRect(x, y, width, height) 楕円の塗り潰しは fillOval(x, y, width, height) 直線の描画は drawLine(x1, y1, x2, y2) 円弧の描画は drawArc(x, y, width, height, θ_0 , θ_1) 文字の描画は drawString(string, x, y)



どんな図形が描画されるかな?



4. 実行結果



胴体や手足を追加で描画してみよう



Javaでお絵描き





カラフルにしてみるのもいいね!



消去機能の作成

1. 以下のように消去用ボタンを配置



消去機能の作成

2. 以下のソースコードを記述

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    Graphics g = bufImage.createGraphics();
    g.setColor(Color.white);
    g.fillRect(0, 0, 512, 512);
    g.setColor(Color.black);
    g.drawRect(0, 0, 511, 511);
    lblDraw.setIcon(new ImageIcon(bufImage));
}
```





起動と同時に描画領域を準備

1. コンストラクタに以下のソースコードを記述



コンストラクタの名前はメインクラスの名前と同じ

ソースコードの再利用



ソースコードの再利用





Java 講習 会の 流れ



お疲れ様でした アンケートにご協力下さい

http://goo.gl/forms/YA1Ktyp0qc