

1.1 Java 開発環境構築

- Java
- Java のインストール
- 初めての Java プログラム
- 統合開発環境




www.lighthouseit.academy



目次

- 1 Java
- 2 Java のインストール
- 3 初めての Java プログラム
- 4 統合開発環境

Java とは

- **Java** とは、**クロスプラットフォーム** [Cross-platform]、**オブジェクト指向** [Object-oriented]、**ジェネリックタイプ** などの特徴を持ち、広く使われているプログラミング言語の一つ。
デバイスやOSが異なる環境でも同じ仕様のアプリを動かすことができる
ソースをテンプレート化して繰り返しソースを書く考え方
- C++ という言語が複雑なプログラムを書くと、色々な問題が生じます。Java は、このような問題を解決するために発明されました。そのため、Java は C や C++ と構文的に似ています。
同じコードで様々な異なるデータを処理することができる仕組み

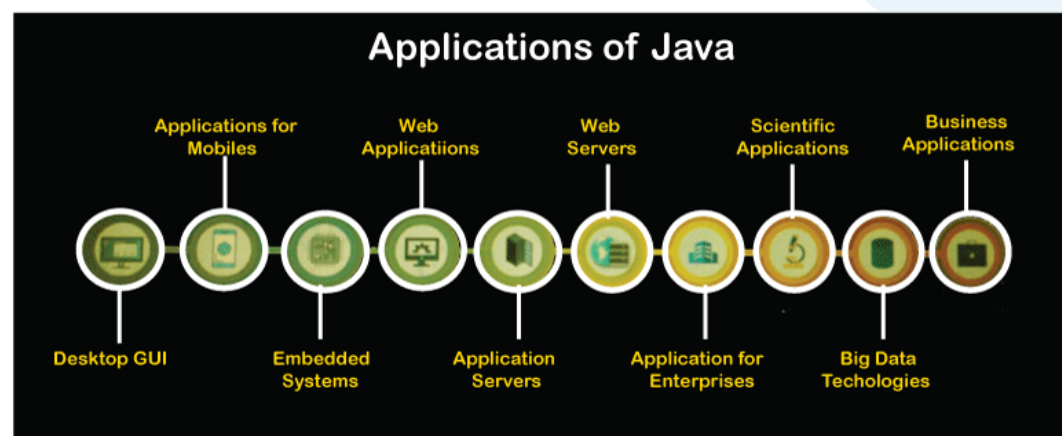


3

- 各用語の説明をする
口頭だけだと、生徒は絶対に理解できないので、必ずスライドにzoomのテキストを使用して記載する
- クラスプラットフォーム：デバイスやOSが異なる環境でも同じ仕様のアプリを動かすことができる
 - オブジェクト指向：Javaはオブジェクト指向型の代表的なプログラミング言語です。オブジェクト指向の「オブジェクト（object）」は「モノ」という意味で、オブジェクト指向とはその名の通り、プログラムをモノとして見る考え方を基本としています。動作手順を一からプログラミングしていくのではなく、使いたいモノの属性や動作をあらかじめ定義しておき、それを操作するという考え方です。
 - ジェネリックタイプ：同じコードで様々な異なるデータを処理することができる仕組み

Java の応用

- Java は、サーバーやモバイルアプリケーション、ビジネスシステム、ウェブアプリ、デスクトップアプリ開発など、さまざまな用途に使用することができます。



●アプリケーション

JavaはWebサービスやアプリ開発との相性が良く、Android Appなどのスマートフォン向けアプリやWeb App、PCアプリなどに活用されています。例えば、企業の業務システムや金融機関の会計システム、運送会社の配送システムなどのWebアプリ、TwitterなどのSNSアプリがJavaで作成されています。

●CG

2Dの画像ベースのCGはもちろん、3D描画ライブラリを利用することで3D CGのソフトウェアもJavaで開発可能です。『ファインディングニモ』『トイストーリー』などは、Javaを使って開発された代表的な作品です。

ゲームエンジンを利用すれば、オリジナルの3Dゲームも製作できます。

●クレジットカード

Javaが搭載されているICカードを「Javaカード」と呼びます。これまでの磁気カードに比べて数千倍に近いデータを記録・暗号化できるため、偽造防止に強いというメリットがあります。そのため、クレジットカードの決済処理には、Javaカードが多く使われています。また、記録できるデータ量が多いことで個人認証情報なども扱うことができます。身分証や社員証としての利用も進んでおり、世界中で50億枚以上のJavaカードが仕様されています。

●Blu-rayディスクプレイヤー

すべてのBlu-rayディスクプレイヤーに、Javaが使用されています。Blu-rayでは、高度なメニューやネットワーク機能などを実現するために、Blu-ray Java (BD-J) という技術を採用しています。Blu-ray Javaは、2007年11月以降に発売されるすべてのプレイヤーに搭載が義務付けられています。

Java の主な特性

- オブジェクト指向プログラミング [Object-oriented Programming, OOP]。

→ 外部のソフトが持つ機能を共有できる仕組み
- 様々な **API** (Application Programming Interface) :
 - グラフィカル・ユーザー・インターフェイス [GUI]。→ ユーザーの使いやすさを重視してアイコンやボタン等を使って直感的にわかりやすく指令を出す
 - マルチスレッド [Multithreading]。→ 複数の処理を並行して行う処理のこと
 - ネットコミュニケーション。
- メモリ管理：ガベージ・コレクション [Garbage Collection, GC]。

→ プログラムで使ったメモリの開放
- 静的型付け [Static Typing]：コンパイル時に変数の型はチェックされる。

人間語から機械語に翻訳
- プラットフォームに依存しない、「Write Once Run Anywhere」、略して **WORA** という性質。

© Suporich Co.Ltd.
5

オブジェクト指向の部分は先ほど話したので割愛する

●APIとは：APIは「Application Programming Interface」の略称で、外部のソフトウェアが持つ機能を共有できる仕組みを指します。APIを使うことにより、プログラムのコードを一から書く必要がないため、開発を効率良く進めることができる。

●APIについては以下の記事を参照してみせると良い

<https://www.internetacademy.jp/it/programming/java/using-java-api-and-class-library.html>

●GUI:ユーザーの使いやすさを重視してアイコンやボタン等を使って直感的にわかりやすく指令を出せるようにしたもの

●マルチスレッド：複数の処理を並行して行う処理のこと

●ガベージコレクション：ガベージコレクションは、直訳すると「ゴミ収集」という意味で、使用しなくなったメモリを自動で消してくれます。そのため、プログラミングで不要になったメモリを消し忘れることはありません。

●コンパイル：「人間語で書いたプログラム」を「機械語に翻訳する」ことをコンパイルという。その際にプログラムの文法もチェックしてくれる
→間違った文章では正しく翻訳することはできないから

目次

- 1 Java
- 2 Java のインストール
- 3 初めての Java プログラム
- 4 統合開発環境

Step 1 開発ツールのダウンロード

- **JDK** (Java Development Kit) とは、Java の開発ツールです。
- オラクルの公式 JDK はこちらでダウンロードできます：
<https://www.oracle.com/java/technologies/downloads/>
- 自分のコンピューターに合ったパッケージを探し、ダウンロードしてください。本講義では、JDK17 を使用します。
- これからのステップは、Windows 用 (➡ p7) と Mac 用 (➡ p14) に分かります。
- すでにインストールされている方は、スキップしても構いません。

Jdkのインストール作業については、こちらのスライドより以下のサイトの方が1つ1つ手順が詳しくのっているため、以下のサイトを参考にとするとよいかも知れない
<https://www.techfun.co.jp/services/magazine/java/windows-jdk-install.html>

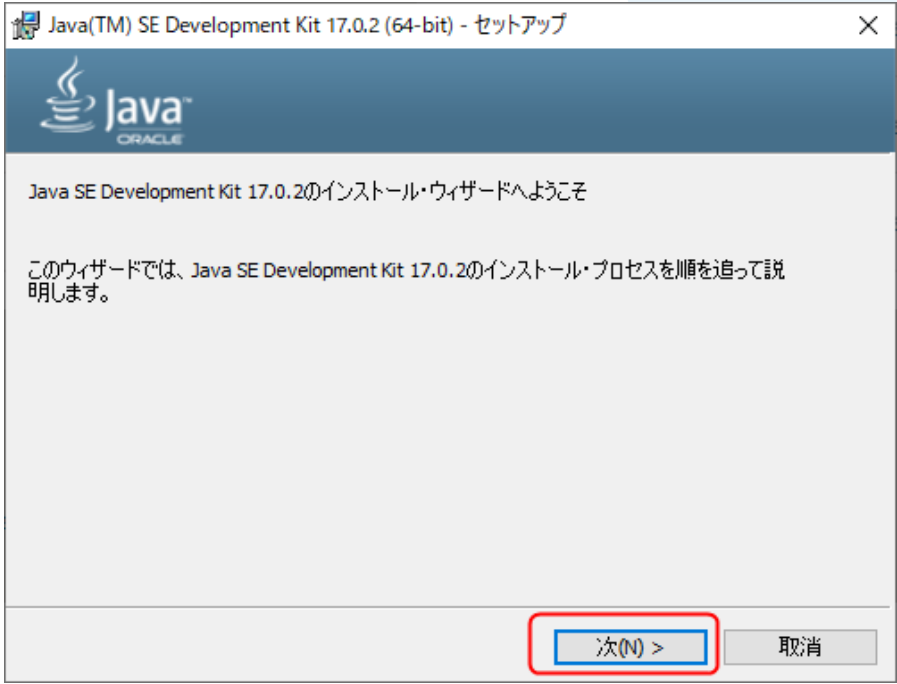
インストール作業については、その作業を行うビデオを配布し、そのビデオを見てインストール作業をしてもらう

インストール作業ができていない学生のみを対応する。

注意点としては、中国人の使用するパソコンは、OSの言語が中国語になっている可能性が非常に高いため、中国人のインストールが上手くいかない場合は、サポートの先生に投げるのが一番である。

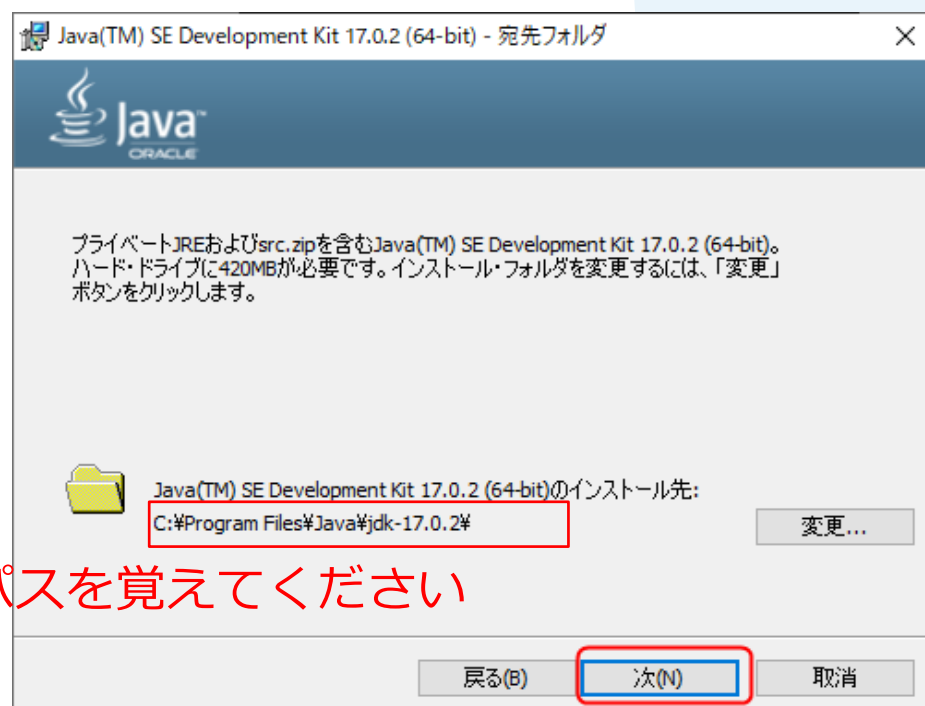
Step 2 JDK のインストール (Windows)

- ダウンロードした .exe ファイルをダブルクリックして実行。



次へ ➡

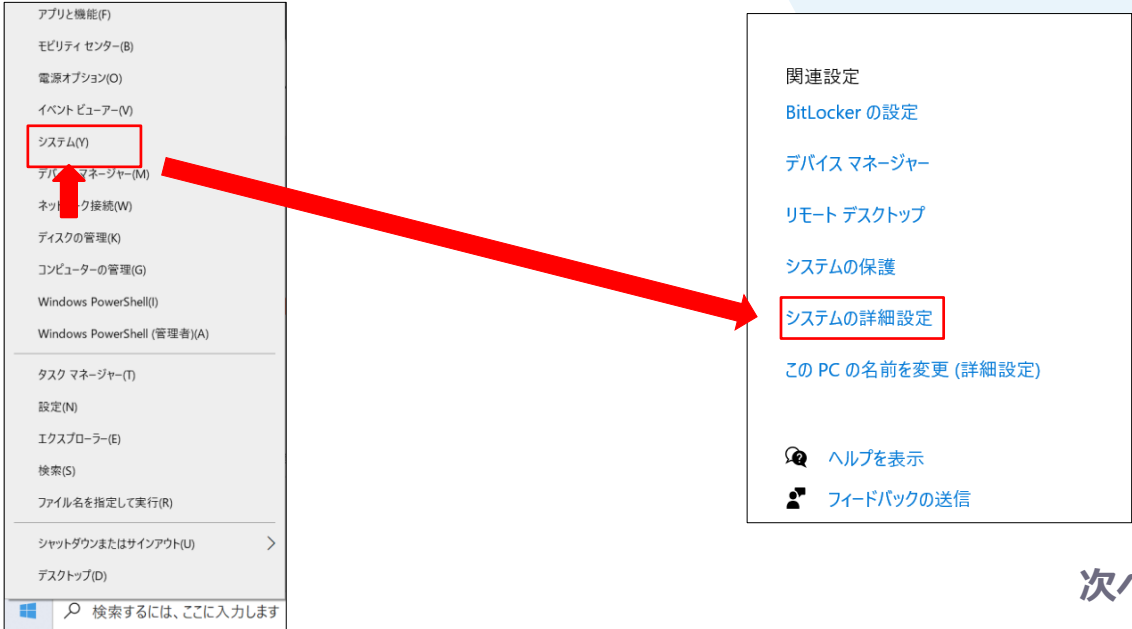
前へ

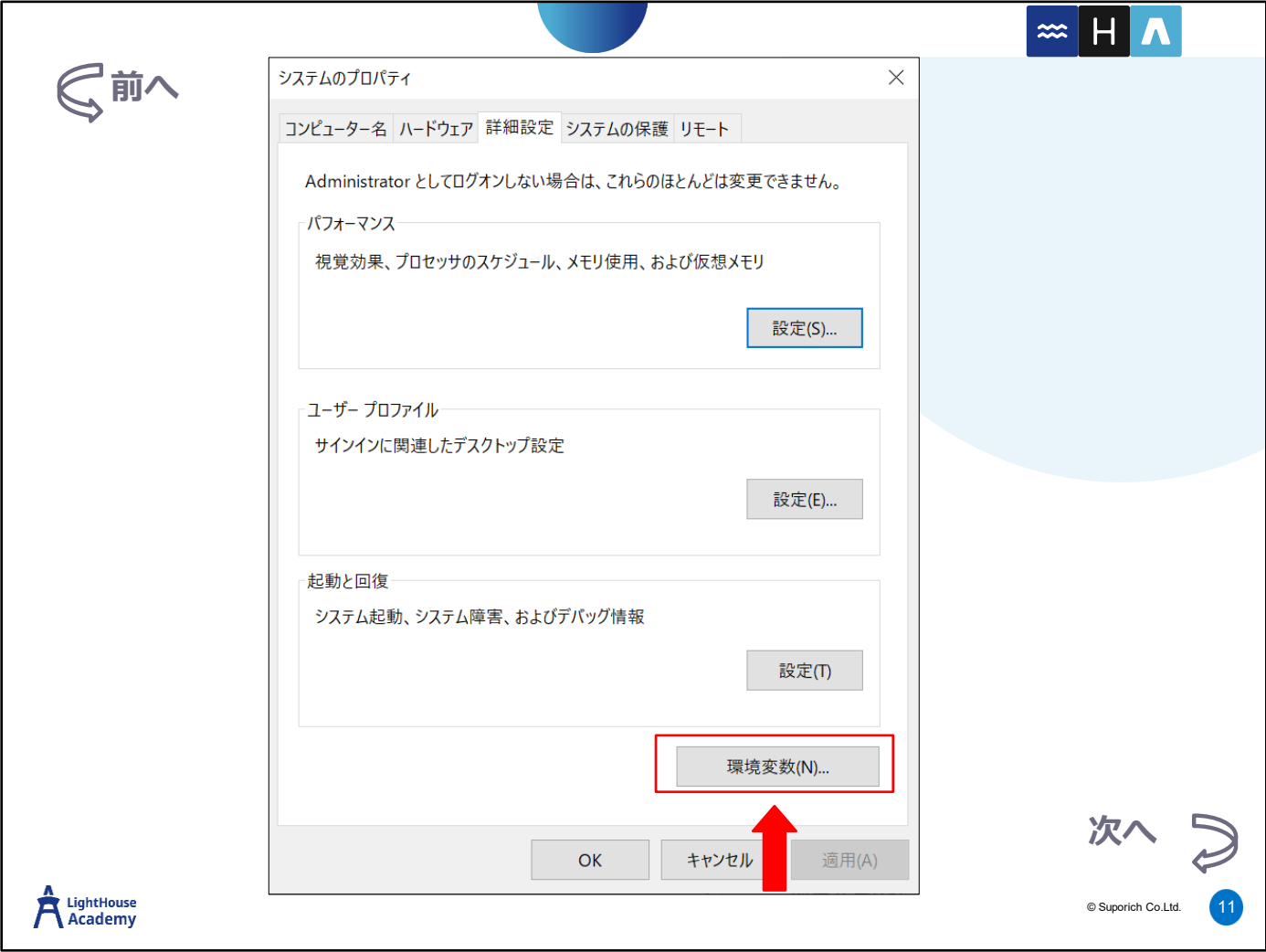


このパスを覚えてください

Step 3 環境変数の設定（Windows）

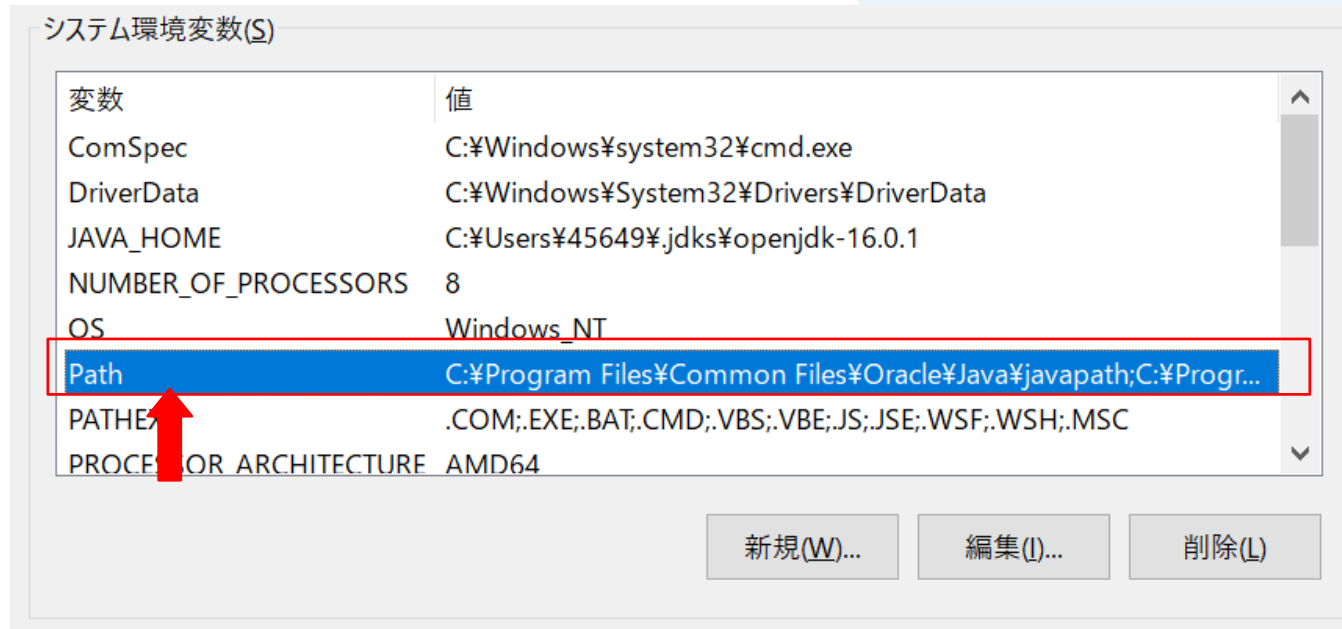
- システム → システムの詳細設定 → 環境変数。





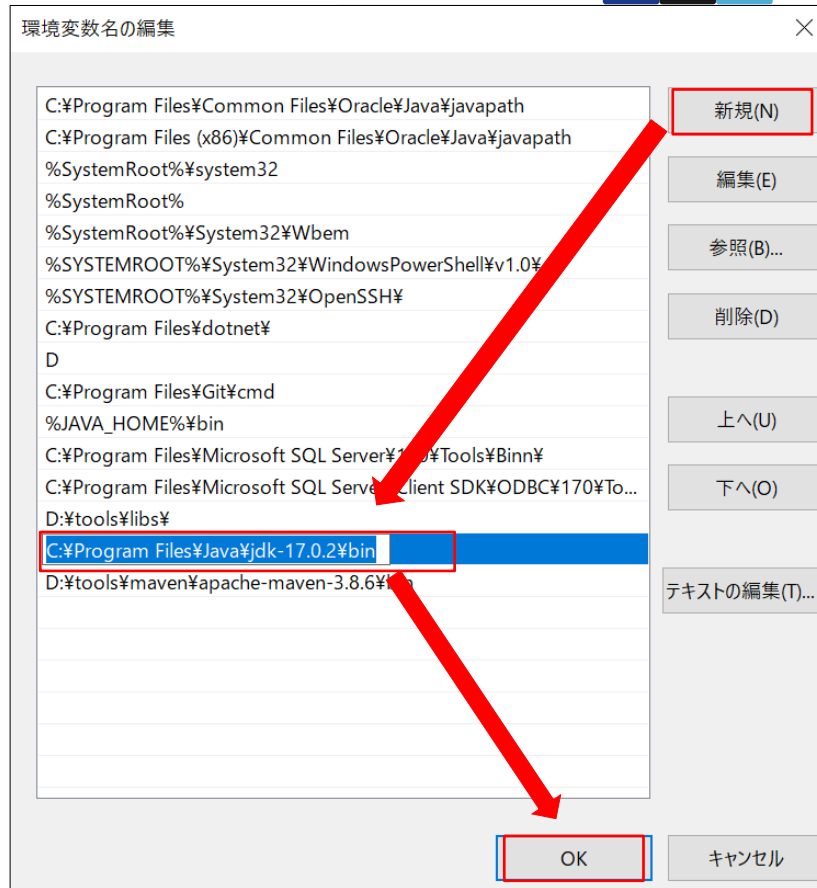


- 「Path」 をダブルクリック :





- 新規：
[JDK のパス]¥bin¥





Step 4 インストールが成功したことを確認 (Windows)

- cmd を起動し、「**java -version**」 (半角スペースに注意) と入力し、Enter キーを押します。

```
C:\Users\Your Name>java -version
```

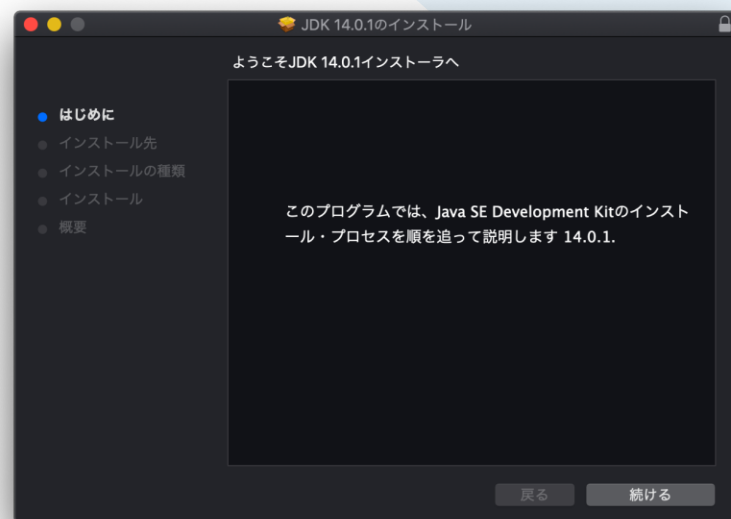
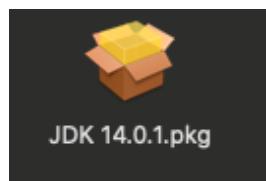
- 環境の設定に成功すると、以下の結果が表示されます。

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed mode)
```

- こうして、無事に環境が整い、Java プログラムを書き始めることができます。

Step 2 JDK のインストール (macOS)

- ダウンロードした .dmg ファイルをダブルクリックし、その中の .pkg ファイルをダブルクリックしてオープン。



Step 3 インストールが成功したことを確認 (macOS)

- Mac は自動的に環境変数を設定します。
- ターミナルを開き、「**java -version**」(半角スペースに注意) と入力し、Enter キーを押します。

```
(base) zmnnoMacBook-Pro:~ zmn$ java -version
```

- このような結果が表示されれば、設定は成功で、Java プログラムを書き始めることができます：

```
(base) zmnnoMacBook-Pro:~ zmn$ java -version
java version "14.0.1" 2020-04-14
Java(TM) SE Runtime Environment (build 14.0.1+7)
Java HotSpot(TM) 64-Bit Server VM (build 14.0.1+7, mixed mode, sharing)
```







© Suporich Co.Ltd.

17

目次

- 1 Java
- 2 Java のインストール
- 3 初めての Java プログラム
- 4 統合開発環境

「Hello World」を書きましょう

- テキストエディタでファイルを作成し、次のコードを入力：

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.println("Hello world!");  
4     }  
5 }
```

- **インデント**^[Indent]（字下げ）に注意。
- 入力ソフトを英語・半角入力にしてください。
- 「**Hello.java**」という名前で保存します。（Mac のテキストエディットを使用する場合は、プレーンテキスト形式で保存する必要があります。ショートカットキー：**Shift + Command + T**。）

ここからは、プログラムを書く内容だが、このスライドを見せて何をやるかを説明する

1：先ほどインストールができたからJAVAのプログラムを実施に動かしてみる

2:その際には、プログラムを書くためのソースが必要である

3：プログラムを書く時は綺麗にかかないと後から見た人がわからいにくいため、プログラムの構造の見通しをよくすること（インデント）を揃える

上記内容を説明したら、一緒に作業を進めていく

手順

1：デスクトップにフォルダを作成する

2:その中にjavaファイルを作成する

3：拡張子に注意点があり、隠しファイルが見えない生徒がいて、txt.javaという拡張子になっていることが非常に多いため、隠しファイルに必ずチェックを入れるところも行う

macについては、「**command**」+「**shift**」+「**.**」で、隠しファイルが表示される（Macを使っている人がいる場合はその説明をする）

4：どこかのソースをわざと間違えて書いた場合、エラーが起きてしまうこともわざと説明する

Syntaxエラーとは構文エラーといって、構文が間違っているというエラーがでてくることを説明。エンジニアはこういうエラーを読み解けなければいけないことを説明する

Hello World の実行

- cmd (Windows) またはターミナル (macOS) を開いて、Hello.java のあるディレクトリに cd します。
- 「**javac Hello.java**」と入力し、Enter キーを押します：


```
1.1 Java Environment Setup¥src>javac Hello.java
1.1 Java Environment Setup¥src>java Hello
```

- 正しく書き込まれていれば、画面には「Hello world!」が表示されるはずです。
- おめでとう！Java プログラムを作成しました！

実行が終わったことを必ず確認する。

もし、うまくいってない生徒がいた場合は、そこに時間をかけても意味がないので
できない生徒を集めて10分間休憩にの際に対応する


注意点としては、エラー文が中国語で出てくる可能性があるため、あらかじめ、翻訳ツールや
chatgptは絶対に準備が必要である



洗濯機の外側のケース	→	<code>public class Main</code>
洗濯機の「スタート」ボタン	→	<code>public static void main(String[] args)</code>
洗濯機内で行っている処理	→	<code>System.out.println("Hello");</code>
洗濯機の操作に追加の設定やオプションを指定する場所 例えば、特定の洗濯モードや温度設定など	→	<code>String[] args</code>

板書例

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         System.out.println("Hello world!");  
4     }  
5 }
```



© Suporich Co.Ltd. 21

ソースに対してイメージを説明する
上記は補足資料となるため、自分で絵や図を使って説明できると良い
プログラムを確認は入れもが必要でそのケースが class Mainになる
プログラムを実行するときには、実行するためのトリガーが必要になる。そのトリガー洗濯機
でいうところのスタートボタンが
Public static void main(String[] args)で、この中に書いた処理が実行される
String args[]は洗濯機の操作に追加の設定やオプションを指定する場所である

Hello World の解説 : クラス

- Java コードは、すべてなにかの「**クラス**^[Class]」に属します。この例では、「**Hello**」というクラスが定義されています :

```
public class Hello {
```

- Java コードファイルの名前は、それが含むクラスの名前と**同じでなければなりません**。保存する際は、ファイル名としてクラス名を使用し、末尾に「**.java**」を付けてください。
- クラスは、オブジェクト指向プログラミングの基本要素であり、後 (➡ §2.1.1) で詳しく説明します。

説明する要点

- 1 : クラス : 洗濯機のケースだった
JAVAのソースは必ず何かしらの入れ物に入っている
入れ物の名前のことをクラス名という
- 2 : クラス名はファイルと名前と必ず同じでなければならない
イメージとしてファイル名は、商品のラベルをイメージする
商品ラベルに電子レンジと書いているのに打っているものが洗濯機では、どういうこと？と
なってしまうので、必ず、ファイル名とクラス名は同じにする
- 3 : ここで、クラス名は名前を書くルールとして必ず大文字で書くことを説明
小文字でももちろん実行できるが、大文字にすることで、後で習う変数等と区別することができるため。

Hello World の解説：メインメソッド

- 「**メインメソッド**^[Main Method]」とは、すべての Java コードの出発点です。メインメソッドの中に書かれるコードは、Java 起動時に順番に実行されます。
- メインメソッドの書き方はどのプログラムでも同じ：

```
public static void main(String[] args) {  
    実行させたいコード  
}
```

- 「**メソッド**^[Method]」については、後（[➡ §1.3.3](#)）で詳しく説明します。今は、メインメソッドのコードをコピーして、各プログラムに貼り付けるだけで大丈夫です。

説明する要点

- 1：先ほどの洗濯機のスタートボタンはメインメソッドという
- 2：メソッドは、この後の章でお話をするけど、簡単に言うとメソッドはある機能を持った部品という意味
- 3：今回であれば、メインメソッド何に書いたものを実行させるスタートさせる一つの部品ととらえると良い

Hello World 説明 : System.out.println()

- **System.out.println ()** メソッドは 1 行の文章を画面にプリント（出力）します：

```
System.out.println( "出力させたいテキスト" );
```

- 文章は必ず二重引用符「" "」で囲むようにしてください。
- このメソッドを使って、見たい結果を出力したり、デバッグしたりすることができます。

説明の要点

1：System.out.printは命令の文

スタートボタンを押したときにどういう処理をさせたいのかというのが命令の文
今回であれば、Helloを表示させる命令の文である

2：文字等の文章を書く時は必ずダブルクォーテーションを記載する

3：この命令文を使うことで、見たい結果を出力することができる。また、ソースがどこまで動いているかを判断するときに使用したりもする

エラーがでてしまうとそれ以降のソースは実行されないなので、そういう意味でどこまで実行されているかをコントロールに表示させて確かめたりする

Hello World 説明 : Java の基本文法

- Java 文は、上から下へ**順次実行**されます。
- 各**文**は、セミicolon「`;`」で終了する。**改行は文の区切りには使えません！**
- 各**単語**は、スペース、タブ、ラインフィード（改行）などの**空白記号**で区切ることができます。
- 中括弧「`{ }`」で囲まれた文は、**ブロック**^[block]といい、例えば、同じクラスや、同じメソッド内の文をまとめることができます。また、特別な場合を除いて、「`}`」の後にセミcolonで区切る必要は**ありません**。

説明の要点


1：JAVAは上から下へ実行される

これについては、実行する時間があった際には、先ほどのソースにSystem.out.printlnを追加し、命令文が2文あった際にどちらが先に実行されているかを実際にみせると非常に良い


2：命令をする文はセミcolonで終了する。先ほどのソースを見せて確かにセミcolonで終わっているところを見せる

3:各単語は…そのまま読み上げる

4：ブロック：これについては次のスライドのように板書して説明する



```
1 package practice;
2
3 public class Main { クラスブロック
4
5     public static void main(String[] args) {
6         メソッドブロック
7         System.out.println("Hello");
8     }
9 }
10
11 }
```

© Suporich Co.Ltd.26

JAVAのソースコードは、波かっこ「{}」で囲まれた部分が、この波かっこで囲まれた部分を「ブロック」という。ブロックには、「クラスブロック」と「メソッドブロック」の2つがあって、JAVAのソースコードは必ずこれらのブロックによる2重構造を持っている。

ここでインデントの話もするどこからどこまでがどこに所属しているブロックなのかを明確にしてソースの読みやすを向上させるためにインデントを揃える必要がある。

コメント

- **コメント**[Comment]とは、コードとして実行されないテキストです。コメントでコードの説明を加えたり、コードを一時に消したり（**コメントアウト**という）することができます。

- 「**//**」で**一行コメント**が書けます：

```
System.out.println("A"); // This will print "A".
```

- 「**/* */**」で**複数行コメント**が書けます：

```
1 /* The next code will print "A" onto the screen.  
2    You can change "A" to any sentence you want. */  
3  
4 System.out.println("A");
```

要点の説明
コメントについては、文字コードの関係で上手く実行できない生徒がいるため、エクリプスをインストールさせたときに生徒に書かせる

コメント：よりプログラムを読みやすくするために、解説文なんかあるとすごく便利である
実は、ソースコードには、解説文を入れることができてそのことを「コメント」と言う。

前回まで、プログラムは、すべて「英語」だったけど、コメントは、解説文で、読み手にわかりやすくするために存在するものだから、「日本語」で記載しても問題ない

練習

- 以下のコードを完成せよ：

```
1 public class Hello {  
2     public static void main(String[] args) {  
3         // Write your code here.  
4     }  
5 }
```

- 以下の文字を出力できるようにしてください：

```
hello  
world
```

Tips 💡

System.out.print() メソッドと
System.out.println() の違いを試
してみてください。

ここについては、時間が無ければ飛ばしても良い
命令の文は少し文字が変わるだけで効果が異なることがわかればよい







CoffeeBreak

プログラミング作法と可読性

同じ機能を持つコードでも、さまざまな書き方があります。最近のプログラミング言語では、スペース、インデント、改行などの空白記号は、プログラマーが自由に配置できるようなものが多い。そのため、開発者ごとにコードのスタイルや習慣、つまり**プログラミング作法**^[Programming Style]が異なります。

一方、一貫した書き方は、コードの**可読性**を高め、共同開発時のコミュニケーションの効率化を図るための鍵になります。ぜひ、主流で権威のある書き方から学んでください。

本講義で使用している Java コードは、Google の開発スタイルを参考にしています：

<https://google.github.io/styleguide/javaguide.html>

Lighthouse Academy

© Suporich Co.Ltd. 30

ここについては必ず説明する
プログラムを書く時にこれといったルールは存在しないため、書き方は人によって異なる
ただ、それだと、非常にプログラムが読みづらく可読性がさがるため、書き方のルールを定めている現場がほとんどである。
今回のプログラムの名前やお作法については、記載のリンクに基づいたルールを使用して皆様と勉強していくことを伝える

目次

- 1 Java
- 2 Java のインストール
- 3 初めての Java プログラム
- 4 統合開発環境

統合開発環境

- **統合開発環境**[Integrated Development Environment, IDE]とは、プログラムの開発を支援するソフトウェアのことです。IDE は、テキストエディタで直接開発するよりも便利な機能をたくさん備えます：
 - ワンクリックでコードを実行、
 - 文法エラーの自動検出、
 - デバッグ機能、
 - コードの自動生成、
 - 多言語開発、
 -
- Java 開発に使える代表的な IDE には、Eclipse、VS Code、IntelliJ IDEA などが挙げられます。この講義では Eclipse を使って開発します。

ここからは統合開発環境のインストールを行う

この統合開発環境を使うメリットについて、スライドに記載されている内容を3つほど列挙する

例えば

コマンドを使わなくてもワンクリックで実行できる

コードの自動生成もしてくれる

文法のエラーも自動抽出してくれるなど

いたれり尽くせりの開発環境がある

その開発環境を今から作っていく

ということを説明できれば良い

Eclipse のダウンロード

- こちらのリンクでダウンロード：
<https://www.eclipse.org/downloads/>
- お使いのシステムに適したものを選んでダウンロードしてください。
- 既にインストールした方は、再度ダウンロードする必要はありません。

エクリプスのインストールについての注意点について
中にはインストーラーが起動できない人があり、どうしてもインストールできない人がいるため、
その人については、日本語版のエクリプスをインストールする
以下が該当リンクである
<https://willbrains.jp/>

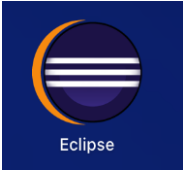
Eclipse をインストール

- ダウンロードした .exe あるいは .dmg ファイルを開いてインストールしてください。
- インストール時は、「Eclipse IDE for Java EE」を選択してください。

こちらを選択してください



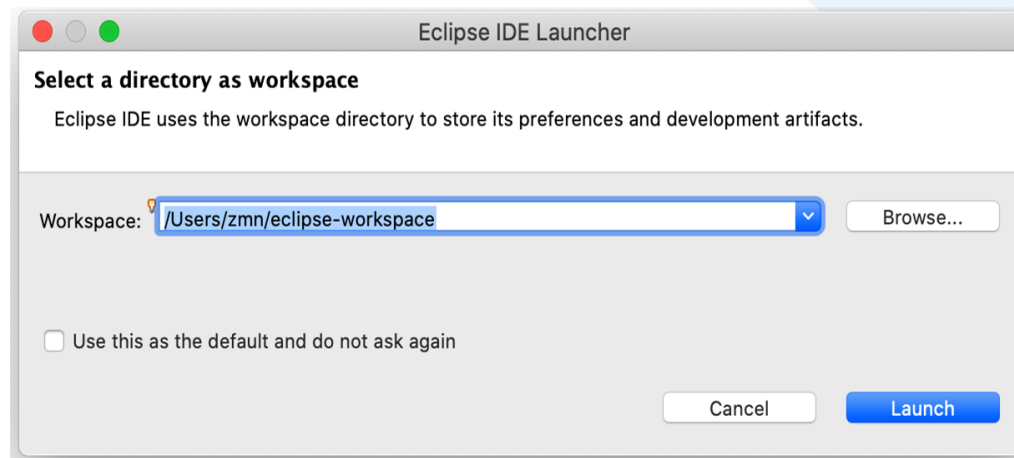
- インストールが終わったら、以下のアイコンが見れます：



インストールさせる際に必ずここは強く言う
3人に1人は話を聞いてくて違うことをするため、必ず「Eclipse IDE for JavaEE」を絶対にインストールさせる

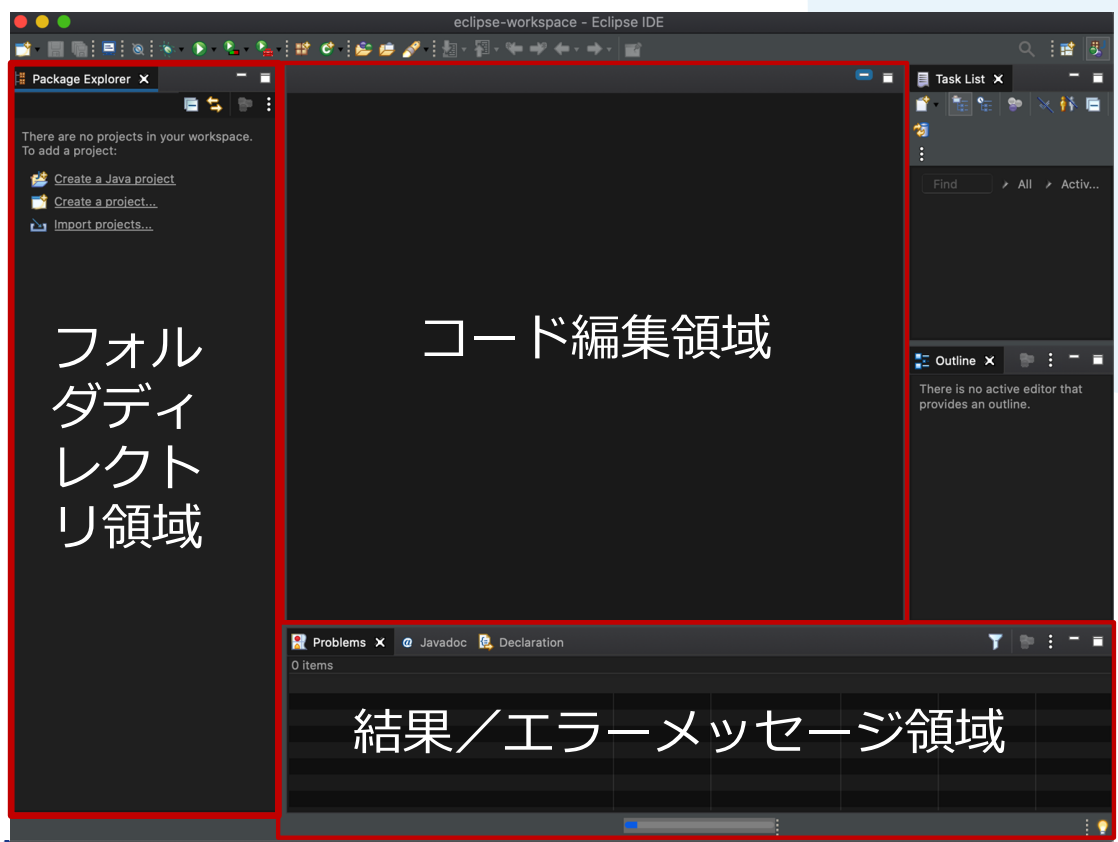
Eclipse を起動

- ダブルクリックで Eclipse を起動します。開くたびにワークスペースの選択ダイアログが表示されますが、デフォルトのフォルダを使用しても構いません。Launchをクリックします。



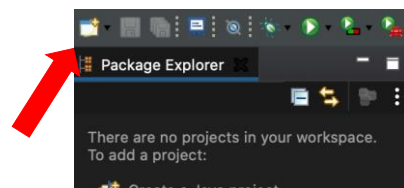
残りのエクリプスの使い方については、すべてハンズオンで実施する
まずは、先ほどの同じHello.javaを作成し、
それを実行できるかやコメント及びコメントアウトも説明する
この時点で作業についてこれない生徒については、ITエンジニアには向いてないことを企業様に報告してもらうように凌さんに連絡する。

Eclipse 画面

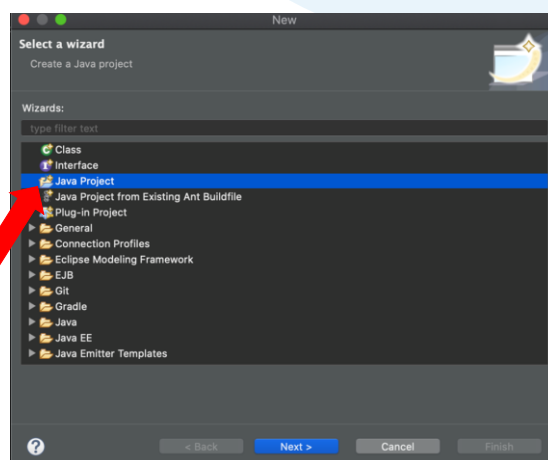


Java プロジェクトの新規作成 (1 / 2)

- 左上の新規作成アイコンをクリック :

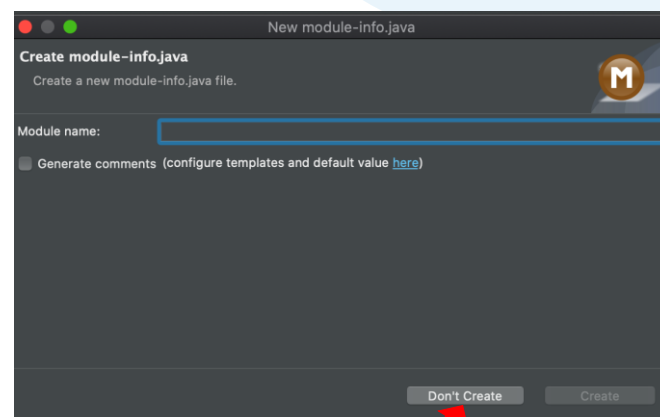
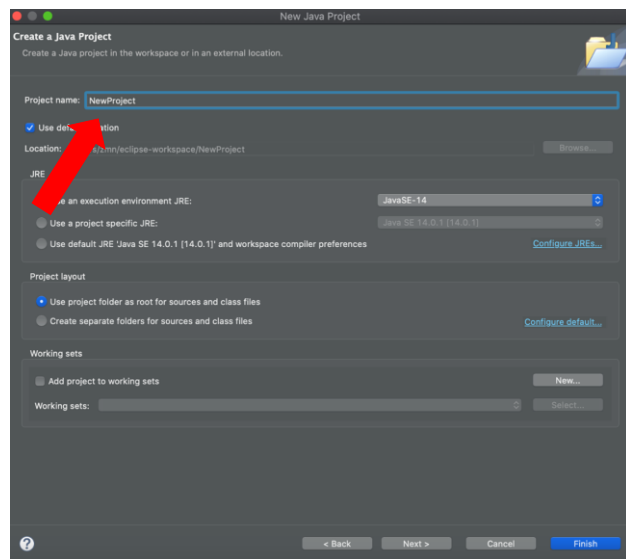


- Java Project → Next を選択 :



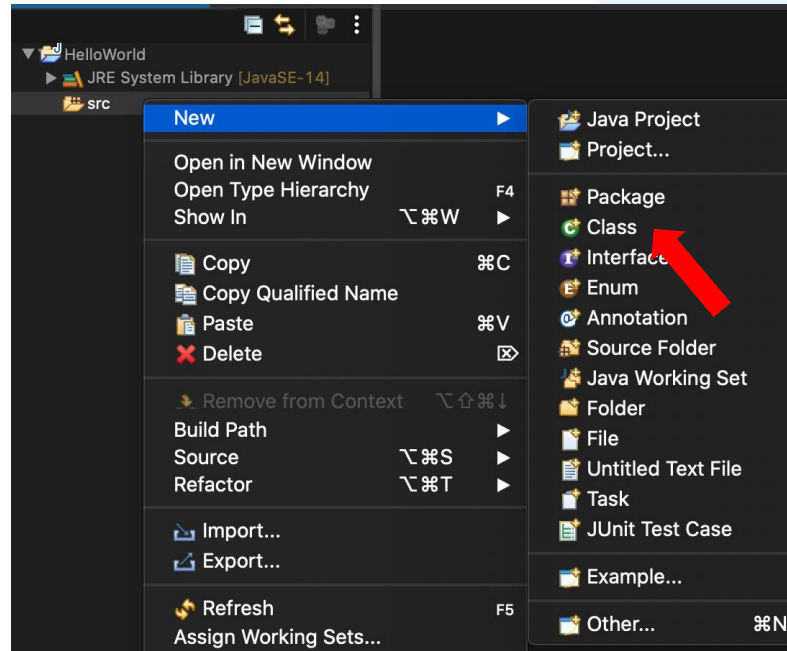
Java プロジェクトの新規作成 (2 / 2)

- Project name に、プロジェクトの名前を書く。
- あとはデフォルトで問題ないので、Finish をクリック。
- モジュールを作成するかどうかを尋ねるウィンドウが表示されますが、Don't Create を選択してください。



Java クラスの新規作成（1 / 2）

- プロジェクトディレクトリにある src フォルダを右クリックし、New → Class。

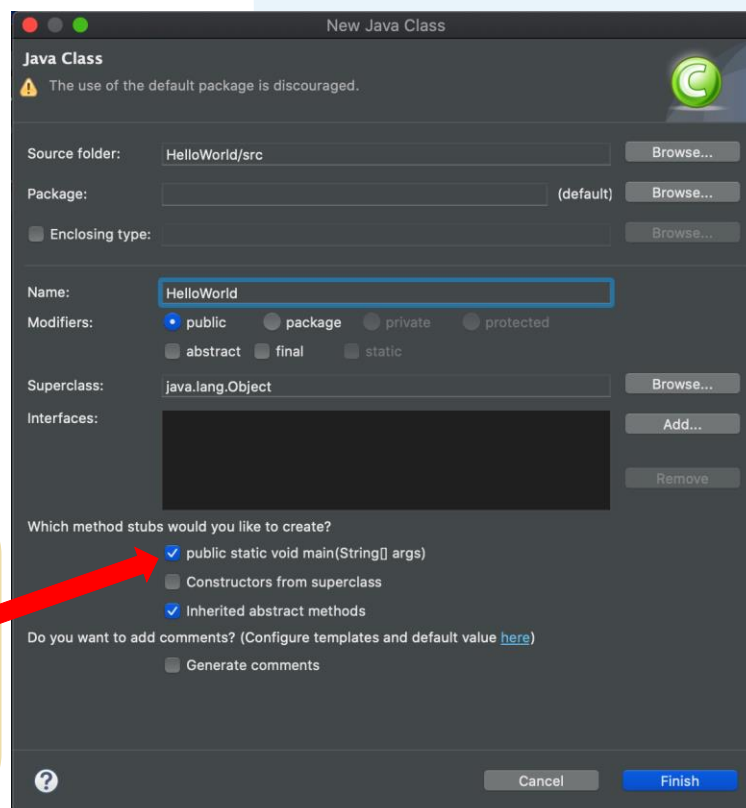


Java クラスの新規作成（2 / 2）



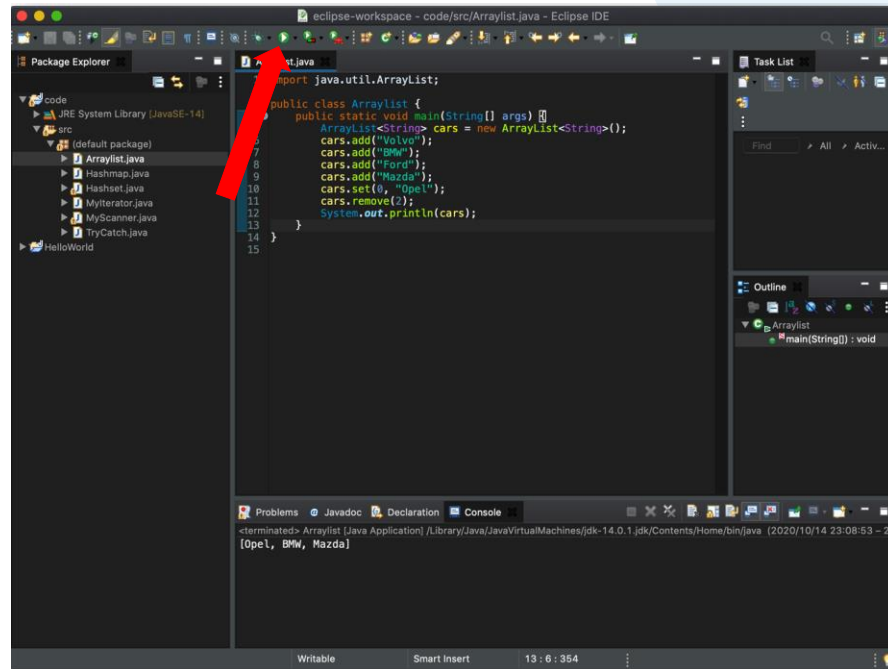
Tips

このボックスを
チェックすると、メ
インメソッドが自動
的に生成されます。



コードの実行

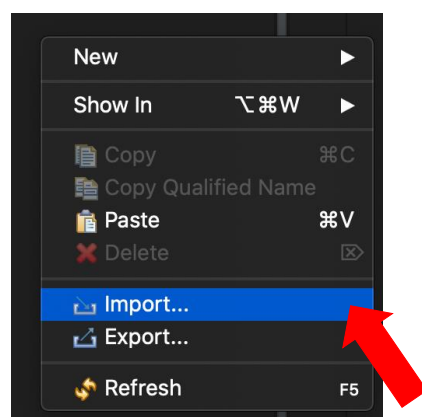
- 緑色のロゴ  をクリックして、今編集中のコードを実行することができます：



プロジェクトをインポート（1 / 3）

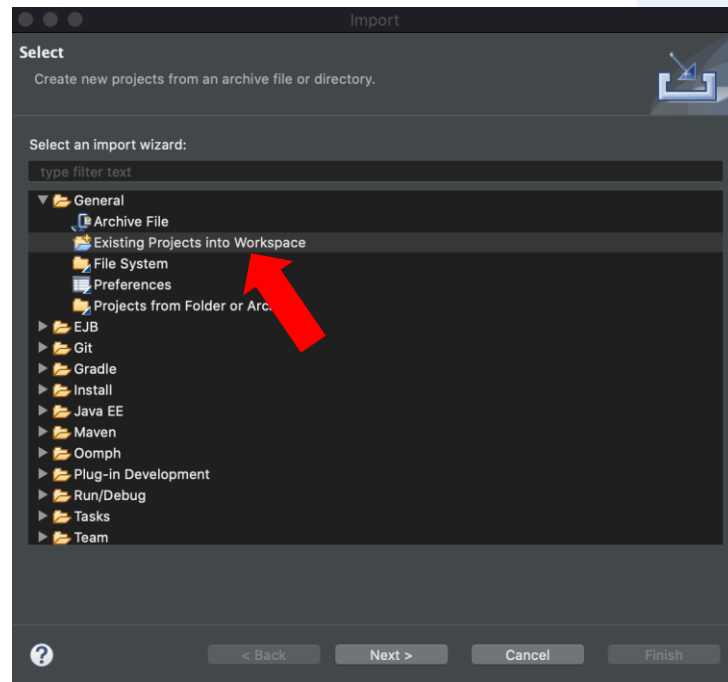


- 今後コードの一部は、Eclipse プロジェクトとして送られますので、次の方法でインポートする必要があります。
- フォルダ領域で右クリックし、「Import」を選択します。



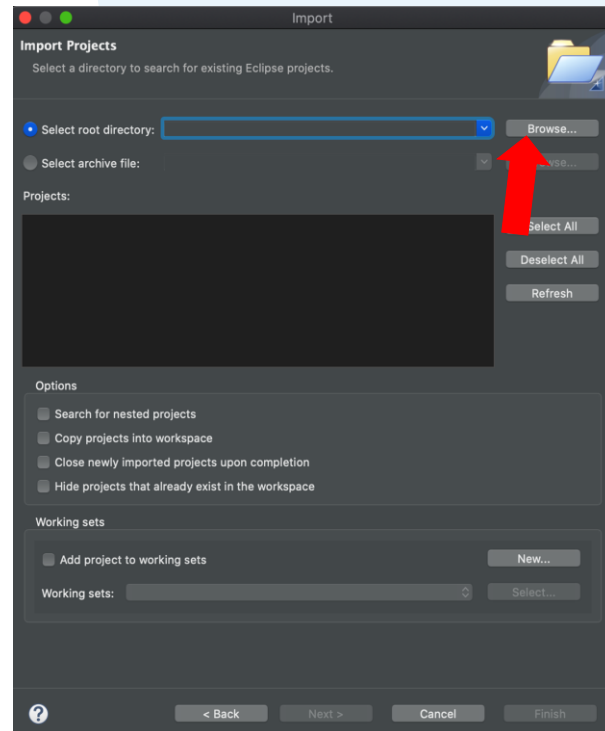
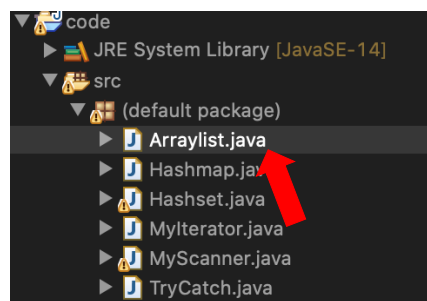
プロジェクトをインポート (2 / 3)

- General を選択 → Existing Projects into Workspace → Next。



プロジェクトをインポート (3 / 3)

- Browse をクリックし、プロジェクトを探して選びます。
「Open」「Finish」を順次クリック。
- インポート後、Java ファイルをダブルクリックしてコードを表示します。









LightHouse Academy

© Suporich Co.Ltd.

45



LightHouse
Academy



Light in Your Career.

THANK YOU!

© Suporich Co.Ltd.