Givery, Inc.

事前セットアップガイド

GitHub CopilotによるAI駆動開発研修　2026年11月17日（火）

当日使う3つのツールと役割分担

GitHub Copilot

作る

VS Codeのチャットから頼みます。Issueや仕様書を読んで、コードとテストを書きます。

VS Code

確かめる

Copilotが変えた箇所を見比べ、ターミナルでテストとアプリの起動を行います。

JDK 21

動かす

演習フォルダに入っているSpring Bootのアプリを、各自のPCで動かします。

使用モデル：当日は、Copilotのチャットの既定のモデルのまま進めます。選べるモデルは契約と貴社の組織の設定で変わるので、このガイドではモデルを選ぶ手順を設けていません。

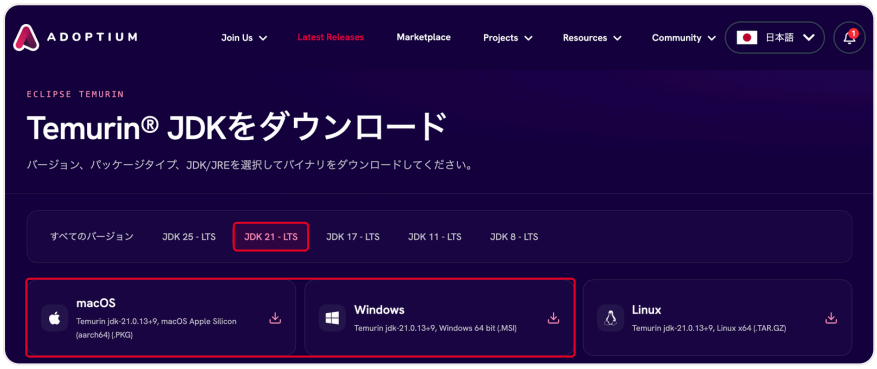
研修の前日までに、STEP 5のテストの成功まで確かめてください。 当日の開始後にインストールを始めると、最初の演習に間に合いません。会社のPCでインストールに申請が要る場合は、早めに申請してください。

Step 1 JDK 21のインストール

STEP 1　ターミナルでjava -versionを実行すると21と表示される

演習のアプリはJavaで動き、起動とテストのコマンドもJavaで書かれています。コンパイルに使うjavacが要るので、JREではなくJDKを入れます。

> **JDKを選んでください**　ダウンロードのページにはJDKとJREの2種類があります。JREにはjavacが入っていないため、演習のコマンドが動きません。



上の枠の **JDK 21 - LTS** が選ばれていることを確かめ、下の枠からお使いのOSのカードを押します。

1. ブラウザでAdoptiumのTemurin 21のページを開きます。
2. **Windowsの場合：** **Windows** のカード（.MSI）を押してダウンロードし、ダブルクリックしてインストールします。途中の選択肢は既定のままで進めます。
3. **macOSの場合：** **macOS** のカード（.PKG）を押してダウンロードし、ダブルクリックしてインストールします。
4. ターミナルを開き直して、次のコマンドを実行します。WindowsではPowerShell、macOSではターミナルを使います。

```
java -version
```

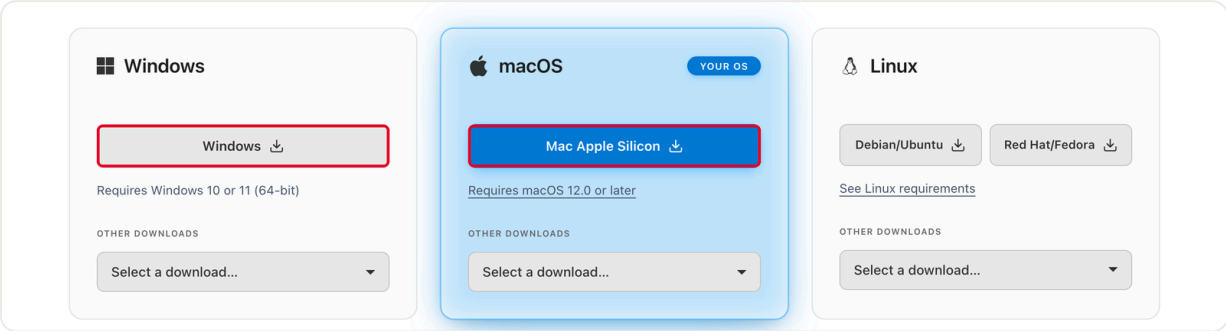
確認チェック

- [ ] java -versionとjavac -versionの両方で、「21」で始まる版の番号が出る（例 openjdk version "21.0.13"）

Step 2 VS CodeとJava拡張パックのインストール

STEP 2　VS CodeにExtension Pack for Javaが入っている

VS Codeは、Copilotのチャットとエディターとターミナルを1つの画面で使うために入れます。Java拡張パックを入れると、Javaのコードの誤りが編集中に分かるようになります。

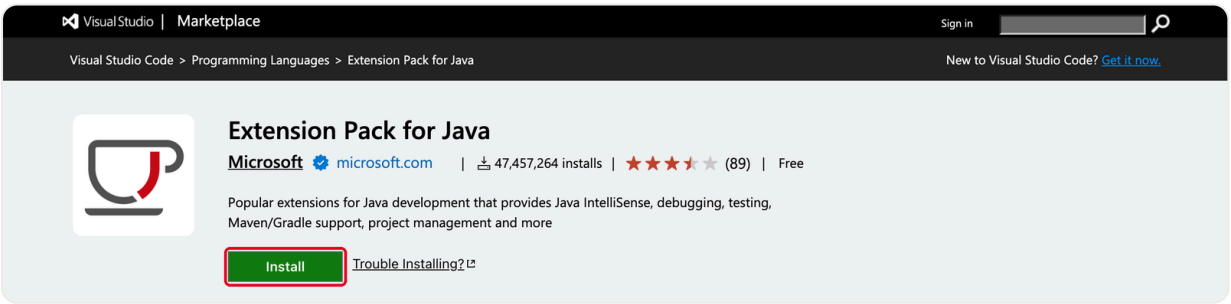


お使いのOSのボタン（枠で囲んだ2つのどちらか）を押してダウンロードします。

1. VS Codeのダウンロードページから、お使いのOSの版をダウンロードしてインストールします。すでに入っている場合は、メニューから最新版に更新します。

2. VS Codeを開き、Ctrl+Shift+X（macOSはCommand+Shift+X）で拡張機能のビューを開きます。

3. 検索欄に `Extension Pack for Java` と入力し、発行元がMicrosoftのものの **Install** を押します。



入れる拡張機能のページです。名前が **Extension Pack for Java** で、発行元がMicrosoftであることを確かめてから、枠の **Install** を押します。

確認チェック

- ☐ 拡張機能のビューで、Extension Pack for Javaがインストール済みになっている

Step 3 GitHub Copilotのサインイン

STEP 3 VS Codeのチャットに送った質問に、Copilotが返事をする

Copilotは、有償版のライセンスが割り当てられたGitHubアカウントでサインインすると使えるようになります。研修で使うのはこのアカウントだけで、リポジトリは作りません。

> **会社から割り当てられたアカウントでサインインしてください**　個人のGitHubアカウントで無料版のCopilotを使うと、チャットの回数の上限で演習の途中に止まることがあります。

1. ブラウザでgithub.com/settings/copilotを開き、Copilot BusinessかCopilot Enterpriseが割り当てられていることを確かめます。
2. VS Codeの左下にあるアカウントのアイコンを押し、GitHubでサインインする項目を選びます。
3. ブラウザが開いたら、手順1で確かめたアカウントでサインインし、VS Codeへ戻る許可をします。
4. Ctrl+Alt+I（macOSはControl+Command+I）でチャットのビューを開き、「Javaのrecordとは何か、1文で答えて」と送ります。

> **Tips：**サインインのあとに組織の承認を求められた場合は、貴社のGitHubの管理者に、Copilotの席が割り当てられているかをお問い合わせください。

確認チェック

- ☐ github.com/settings/copilotに、組織から割り当てられたプランが表示される
- ☐ チャットに送った質問に、Copilotが日本語で返事をする

Step 4 演習フォルダの展開

STEP 4　デスクトップに展開した1117_handsonフォルダ

演習で使うアプリと、Copilotへの指示とプロンプトが入ったフォルダです。Spring Bootなどのjarも入っているので、Mavenの設定やライブラリのダウンロードは要りません。

1. お使いのPCに合う方のボタンを押して、ZIPを保存します。約34MBあり、読み込みに少し時間がかかります。
2. **Windowsの場合**：ZIPを右クリックして `すべて展開` を選び、展開先にデスクトップを指定して `展開` を押します。
3. **macOSの場合**：ZIPをデスクトップに移してから、ダブルクリックします。
4. デスクトップの1117_handsonフォルダを開き、中に次の4つがあることを確かめます。

ボタンで保存できないとき

会社のブラウザの設定によっては、ボタンで保存できないことがあります。その場合は、分けた2つのファイルを保存してから、コマンドで1つにつなぐ手順に切り替えてください。Windowsの方はpart1とpart2を同じフォルダに保存し、そのフォルダで開いたコマンドプロンプトで次を実行します。

```
copy /b 1117_handson.zip.part1+1117_handson.zip.part2 1117_handson.zip
```

Macの方はpart1とpart2を同じフォルダに保存し、そのフォルダのターミナルで次を実行します。

```
cat 1117_handson_Mac.zip.part1 1117_handson_Mac.zip.part2 > 1117_handson_Mac.zip
```

どちらも難しい場合は、研修の担当の方を通じて講師にお知らせください。ZIPを別の方法でお渡しします。

```
1117_handson
├── lib          Spring Boot と H2 と JUnit の jar（98本）
├── skillsheet   Session03〜05 で開くフォルダ
├── vibe         Session03 で開くフォルダ
└── README.md
```

> **Tips：**展開したあとは、フォルダの中の構成を変えないでください。skillsheetとvibeは、1つ上の階層のlibを探して動きます。

確認チェック

- ☐ 1117_handsonの中に、lib、skillsheet、vibe、README.mdがある

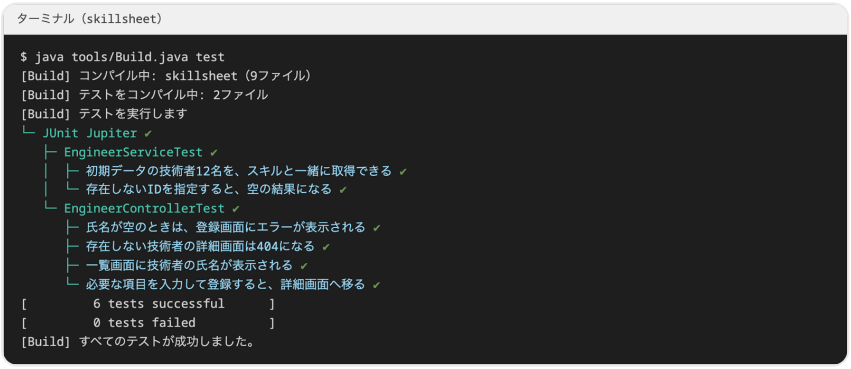
Step 5 テストとアプリの起動

STEP 5　6件のテストが成功し、ブラウザに技術者一覧が表示される

STEP 1から4までがそろっているかを、演習と同じコマンドで確かめます。ここまで通れば、当日の演習を始められます。

1. VS Codeのメニューで **File** から **Open Folder...** を選び、1117_handsonの中のskillsheetフォルダを開きます。日本語表示の場合は **ファイル** から **フォルダーを開く** です。
2. フォルダの作成者を信頼するかを聞かれたら、信頼する方を選びます。
3. メニューの **Terminal** から **New Terminal** を選び、ターミナルを開きます。
4. 次のコマンドでテストを実行します。終わるまで数秒から数十秒かかります。

```
java tools/Build.java test
```



> 最後の行に「[Build] すべてのテストが成功しました。」と出れば合格です。その上の行で、6 tests successfulと0 tests failedを確かめます。

5. 続けて次のコマンドでアプリを起動し、ブラウザでhttp://localhost:8080/を開きます。

```
java tools/Build.java run
```



見出しの下に「12名」と出て、12名の技術者が表に並べば合格です。

6 確かめたら、ターミナルでCtrl+Cを押してアプリを止めます。

確認チェック

- ☐ ターミナルに「すべてのテストが成功しました」と表示された
- ☐ ブラウザに技術者一覧が表示され、見出しの下に「12名」と出た
- ☐ Ctrl+Cでアプリを止めた

最終確認

STEP 6 開催前日までにそろえる5項目

- ☐ java -versionとjavac -versionで21の版が表示される（ Step 1 ）
- ☐ VS CodeにExtension Pack for Javaが入っている（ Step 2 ）
- ☐ VS Codeのチャットで、Copilotが返事をする（ Step 3 ）
- ☐ デスクトップに1117_handsonフォルダがある（ Step 4 ）
- ☐ skillsheetでテストが6件とも成功し、技術者一覧が表示される（ Step 5 ）

よくある質問

「javac が見つかりません」と表示されます

JDKではなくJREが入っているか、JDKを入れたあとにターミナルを開き直していない状態です。Step 1でJDKを入れ直してから、VS Codeを閉じて開き直し、もう一度実行してください。

java -versionの表示が21ではありません

以前に入れた別の版のJavaが先に見つかっています。17以上の版であれば演習は動くので、そのまま進めて構いません。社内でMicrosoft Build of OpenJDKのような別のJDKが決まっている場合も、17以上であれば同じです。17より古い版が表示される場合は、Step 1のJDK 21を入れてから、PCを再起動してください。

テストの途中で「8080番ポートを別のアプリが使っています」と出ます

runで起動したアプリが動いたままです。起動したターミナルでCtrl+Cを押して止めてから、もう一度実行してください。

VS Codeの右下に、Javaのプロジェクトについての通知が出ます

Java拡張パックがフォルダを読み込んでいる間の通知です。演習はターミナルのコマンドで動かすので、閉じて構いません。MavenやGradleの設定を求められても、設定は要りません。

VS Codeのメニューが英語で表示されています

英語の表示のままで受講できます。このガイドとハンズオンガイドは、メニューの名前を英語の表示で書き、日本語の表示が違う所だけ併記しています。日本語の表示にしたい場合は、拡張機能のビューでJapanese Language Pack for Visual Studio Codeを入れて、VS Codeを開き直してください。

会社のPCで、ZIPのダウンロードやインストールができません

社内の申請が必要な場合があります。研修の担当の方を通じて講師にお知らせください。演習フォ

ルダは、メールやファイル共有で別にお渡しすることもできます。

※本教材の演習に登場する企業・人物・数値はすべて架空です。