



ITの基礎力と実践力を身に付ける

I TECH SCHOOL

- 新人・中途採用向けJava基礎コース -

はじめに

本研修は、Spring FrameworkでのWebシステム作成を最終的な目標としています。以下のように研修の期間を3つに分け、各4週間にわたる講義を通して段階的に学習していきます。

- Javaの基本的な技術を学び、勤怠管理用のコンソールアプリを作成
 - Webについて学習後、チームで図書館の蔵書管理用Webシステムを作成
 - フレームワークについて学び、総合演習として日報管理用Webシステムを作成
- これによりJava言語のプログラミングだけでなく、開発工程を一通り経験することができます。

演習スケジュール表

カリキュラム	テキストタイトル	所要期間
コンピュータ	• コンピュータ 基礎	約4週間
プログラミング	• プログラミング 基礎	
Java	• Java 基礎 • Java 応用 • Java 単体テスト	
ネットワーク	• ネットワーク 基礎	
DB	• データベース 基礎 • MySQL 基礎	
JDBC	• Java DB • 設計手法	
Java総合演習	• Java 総合演習	

カリキュラム	テキストタイトル	所要期間
Web	<ul style="list-style-type: none"> • Web ブラウザ • HTML 基礎 • CSS 基礎 • CSS レンダリング • Web デザイン 	約 4 週間
	<ul style="list-style-type: none"> • Apache HTTP 基礎 	
	<ul style="list-style-type: none"> • JavaScript 基礎 • JavaScript DOM • jQuery 基礎 • Bootstrap 基礎 	
開発ツール	<ul style="list-style-type: none"> • Git 基礎 	
Webシステム	<ul style="list-style-type: none"> • Jakarta EE 	
	<ul style="list-style-type: none"> • 詳細設計 	
	<ul style="list-style-type: none"> • Webシステム総合演習 	

カリキュラム	テキストタイトル	所要期間
フレームワーク	<ul style="list-style-type: none"> • maven 基礎 • Spring 基礎 • Spring Boot • Spring Security 	約 4 週間
システム開発工程	<ul style="list-style-type: none"> • システム開発工程 	
フレームワーク総合演習	<ul style="list-style-type: none"> • Spring 総合演習 	

Javaに入る前に

Javaをより深く理解をするために必要なコンピュータに関する基礎知識について学習します。システム開発に必要な知識だけでなく、システムエンジニアとして知っておくべき知識やハードウェア・OSについても詳しく学びます。

コンピュータ基礎

コンピュータ基礎では、そもそもパソコンとは何かを学びます。プログラムが実行されている環境や構造について知り、実際にデータがパソコンのどの部分に保持されているのかを理解することができます。

【詳細】

- コンピュータについて
- ファイル
- ディレクトリ
- 2進数
- ユーザ管理・グループ管理
- ハードウェア
- ソフトウェア
- OS
- RAID

プログラミング基礎

アプリケーションやプログラミングに関する基礎知識、各プログラミング言語の実行手順の違いについて学習します。コマンドプロンプトを使いながら実践的に学習することで、内部で行われている処理を具体的にイメージできるようになります。

【詳細】

- アプリケーションとは
- プログラミングとは
- プログラミング基礎
- ネイティブ実行方式
- 中間言語方式
- インタプリタ方式

Java総合演習に向けて

Javaアプリケーションを作成するに当たって、Javaプログラミングやネットワーク、データベースについて学びます。コンソールアプリを題材にすることで、ユーザインターフェース設計の負担を軽減することができ、データのやり取りや処理方法に集中して学ぶことができます。

Java

テキスト内のサンプルコードを実行して動作を確認しながら学んでいきます。サンプルコードの実行結果を予想・考察したり、例題に取り組んだりすることで、主体的に学ぶことが出来る構成となっています。また、統合開発環境でアプリケーションを開発する前に、コマンドプロンプトを使うことで、コンパイルから実行までの一連の仕組みについても理解することが出来ます。

【詳細：Java基礎】

- Javaについて
- エスケープシーケンス
- リテラル
- 変数とデータ型
- 定数
- キャスト
- 算術演算子
- コメント
- コーディング規約
- if文
- switch文
- 三項演算子
- 配列
- コマンドライン引数
- for文
- while文
- 拡張for文

【詳細：Java応用】

- クラス定義
- メンバ変数・定義
- メソッド
- クラスを使う
- インスタンス
- アクセス修飾詞
- 動的/静的メソッド・変数
- 変数スコープ
- エディションの種類
- JDKとJRE
- 実行とクラスパス
- パッケージ
- 統合開発環境
- オブジェクト指向
- メソッドのオーバーロード
- コンストラクタ
- thisキーワード
- カプセル化とは
- クラスの継承
- オーバーライド
- 抽象クラス
- ポリモーフィズム
- ArrayList
- HashMap
- ラッパークラス
- 例外
- JavaBean
- インターフェイス
- 正規表現

【詳細：Java単体テスト】

- テストの目的
- コードレビュー
- ブラックボックステスト
- ホワイトボックステスト
- テスト仕様書
- assert文
- JUnit

ネットワーク

コンピュータ内部でどのようにIPアドレスが使われているかを実際に確認しながら学習します。また、ネットワークの周辺機器や、コンピュータ同士が通信している実態を学ぶ事により、セキュリティの必要性や具体的な安全策を理解することに役立ちます。

【詳細：ネットワーク基礎】

- 通信の範囲と方法
- プロトコルとは
- TCP/IPモデル
- イーサネット
- NIC
- MACアドレス
- HUB
- IP
- ルータ
- IPv6
- ポート
- TCPとUDP
- DNS
- URL
- HTTPとFTP
- アドレス変換
- ネットワークコマンド
- SSL通信

DB

データベースシステムやデータの扱い方について学習した後、実際にMySQLを使用して演習に取り組みます。インストールからDB作成、操作を行う一連の流れを通して、自ら考えてデータの操作ができるようになります。

【詳細：データベース基礎】

- DBMSとは
- DBオブジェクト
- ユーザとロール
- RDBMS
- SQL言語
- SQLインジェクション
- ER図

【詳細：MySQL基礎】

- MySQLを使う
- カラムの定義
- SQL文
- 集合関数
- 内部結合・外部結合
- グループ化・HAVING句
- CASE文
- SQL関数
- SQLパラメタ
- インデックス
- ビュー
- プロシージャ
- トリガ
- トランザクション
- コミットモード

JavaとDBについての知識を習得した後、JDBCについて学ぶことで、さらに理解しやすい構成になっています。設計手法では、演習で必要となるDAO・DTOやMVCを中心に、既存のデザインパターンを学び、効率化を意識したソースについて考えます。

【 詳細 : Java DB 】

- Java DB
- JDBCでのDB操作
- パラメタを使ったSQL
- ER図

【 詳細 : 設計手法 】

- デザインパターンとは
- Singletonパターン
- Facade
- FrontController
- DAOとDTO
- IoC(DI)パターン
- MVC
- REST

Webシステム総合演習に向けて

ソフトウェアの設計モデルであるMVC（Model View Controller）の考え方に沿ったWebシステムの作成を目的としており、Java総合演習で学んだデータ処理を「Model」として、システムの表示部分「View」やユーザからの情報を「Controller」によって制御・操作する方法を学びます。

演習では、習得した技術の確認だけでなく、チームで効率的にプロジェクトを進めるために何をすべきか試行錯誤する機会となっています。

Web

Webページに関して、HTMLの書き方とデザイン・JavaScriptによる画面操作を学び、システムの表示部分を作成できるようになります。デザインではページの見やすさや使いやすさに配慮する工夫を学び、画面操作では、HTML文書やXML文書进行操作するための規約であるDOM（Document Object Model）について理解を深めてからjQueryを学ぶことで、段階的な理解を促します。Webページで多用される非同期通信についても習得し、実践的な力が身に付きます。

【詳細：Webブラウザ】

- 通信機能
- レンダリング機能
- JavaScript機能
- デベロッパーツール
- WebGL
- WebAssembly

【詳細：HTML基礎】

- マークアップ言語
- エレメント（要素）
- 属性
- HTMLの基本構造
- ブロックとインライン
- HTMLのタグ
- エスケープシーケンス
- メタデータについて
- フォーム

【詳細：CSS 基礎 / CSS レンダリング】

- スタイルの書式
- CSSの定義方法
- サイズ・色の指定方法
- スタイル - テキスト
- スタイル - ボックスモデル
- スタイル - 要素
- スタイル - 表示効果 -
- セレクタ
- レンダリングの基本
- 要素の配置
- フロートレンダリング

【詳細：Web デザイン】

- 文章の可読性・視認性
- 空白と余白
- フォントの種類
- 色相・彩度・明度
- トーン
- 類似色・補色
- コントラスト
- 配色の基本
- ページデザインの流れ
- ワイヤフレーム

【 詳細 : Apache HTTP 基礎 】

- ApacheHTTP
- Apacheの設定

【 詳細 : JavaScript 基礎 / JavaScript DOM 】

- JS基本構文
- 演算子
- 関数とは
- コールバック関数
- 選択文
- 配列
- 繰り返し文
- オブジェクト
- DOMの操作
- Formオブジェクト

【 詳細 : jQuery 基礎 】

- jQueryオブジェクト
- メソッド
- 兄弟・親子要素の取得
- イベント
- readyメソッド
- Ajax
- json

【 詳細 : Bootstrap 基礎 】

- Bootstrap
- Bootstrapコンポーネント

開発ツール

演習でのグループワークを念頭に置きつつ、バージョン管理の必要性や仕組みについて、Gitを使って学びます。また、bash（シェルプログラム）を使って操作することで、コミットまでの流れを一つ一つ確認します。

【 詳細 : Git基礎 】

- バージョン管理システム
- Gitについて
- リポジトリ管理
- ファイルの状態
- リモートリポジトリ

Webシステム

サーバの構造や仕組みを学び、内部での処理の呼び出しについて具体的な手順を考えながら実装できるようになります。モデリング言語や本番環境でのサーバ配備など、実用的な知識についても触れます。

【 詳細 : Jakarta EE (旧 : Java EE) 】

- Jakarta EEとは
- Tomcatについて
- サーブレット
- アノテーションの利用
- リクエストパラメタ
- Servletのライフサイクル
- リダイレクト
- サーバ内転送
- ページ遷移
- スコープについて
- JSPとは
- スクリプト
- アクション
- ディレクティブ
- サーバへの配備方法

【 詳細 : 詳細設計 】

- UML

- クラス図

- シーケンス図

Spring総合演習に向けて

研修の最終段階として、これまでのJava・Web総合演習から得た知識・経験を繰り返し活用する事で、受講生自身が研修の成果を実感できるようになっています。また、設計・デザインから実装・テストまでの一連の開発工程をグループで進めることにより、現場で必要とされる技術や実践的な知識を習得し、最終的にSpringフレームワークを使用したWebシステムを作成できるようになります。

フレームワーク

Springに関する基礎技術を習得し、最終的に実用性の高いSpring Bootを習得することを目指します。これまでに学んだMVCやDB接続の知識を生かし、Springの多様なフレームワークやライブラリについて学習していきます。不具合の修正を視野に入れ、Springの扱い方だけでなく、根幹となる考え方や手順が省略された部分とその背景についても理解することが出来、エラー時の対応も視野に入れた講義となっています。

【詳細：Maven基礎】

- apache Mavenとは
- pom.xml
- ライフサイクル
- フェーズ
- プラグイン
- Tomcatとの連携

【詳細：Spring基礎】

- フレームワークとは？
- Springプロジェクトに関して
- Spring DIについて
- インジェクション
- Bean構成クラス
- AOPとは
- タグを使ったAOP
- アノテーションを使ったAOP
- Spring JDBCについて
- 関数型インターフェース
- ラムダ式
- Spring MVCの特徴
- Front Controller
- HTTPメソッドによる遷移
- パラメタとリクエスト変数
- テンプレートエンジン
- Thymeleaf - 属性・変数
- Thymeleaf - インクルード
- @Autowired
- @Componentのスコープ

【詳細：Spring Boot】

- Spring Bootの構成
- 設定ファイル - yaml
- RESTful
- RequestMappingの応用

【 詳細 : Spring Security 】

- Spring Security
- 認証・認可について
- ログイン・ログアウト
- ログインページのカスタマイズ
- 認証のカスタマイズ
- アクセス制限
- 認証後の処理

システム開発工程

開発者として打ち合わせをする際に考慮すること、要件や議事録に関する留意点、テスト工程について学習します。実際の現場でも重要とされている「技術以外の知識」について、講師の経験をもとにした実例を交えながら知ることが出来ます。

【 詳細 : システム開発工程 】

- 打ち合わせに関して
- 開発とユーザ
- システム開発工程に関して
- 要件について
- ソフトウェア品質
- テスト用語
- テスト工程の種類