疑念論述

１．　オブジェクト指向とは

オブジェクト指向は「概念」であり、形のない概念のようなモノを中心に考える事である。

オブジェクト指向と言えば「カプセル化」「継承」「ポリモーフィズム」の３つの特徴があげられる。

「カプセル化」とは他のプログラムから干渉されないようにつくる考え方である。「継承」とは同じようなプログラムは共有化して使う考え方である。「ポリモーフィズム」とは汎用的な形にできるようにしましょうという考え方である。

具体的な例として、テレビというモノを操作する時、中でどのようなプログラムが動いているか知る必要はなく、私たちはリモコンさえ操作できれば思いのまま好きな番組を見ることが出来る。

「こういう“モノ”を作りましょう」「そして、その“モノ”を使いましょう」というのがオブジェクト指向という考え方である。

参考にしたリンク

https://eng-entrance.com/what-oop

https://www.youtube.com/watch?v=SjwbqGK-H1o

２．　Github flow とは

＊含めるべき文言（リポジトリ・main・リモート・ブランチ）

Github flow は、開発ワークフローのブランチモデルの一種である。

　main ブランチのものであれば何であれデプロイが可能で、新しい何かに取り組む際は説明的な名前のブランチを main から作成する。作成したブランチにローカルでコミットし、サーバー上の同じ名前のブランチにも定期的に作業内容を push する。

　作成したものは「リポジトリ」というファイルやディレクトリの履歴を管理する場所に保管される。リポジトリの中にはローカルリポジトリとリモートリポジトリがあり、ローカルリポジトリとは自分のパソコン内のリポジトリのことでリモートリポジトリとはネットワーク上のリポジトリの事である。その関係上、リモートリポジトリを作成したら、ローカルリポジトリと同期する必要性があります。それを経てデプロイをする。

参考にしたリンク

https://style.potepan.com/articles/31071.html

３．　サーバーサイドエンジニア・フロントエンドエンジニアの違い

まず、簡単にサーバーサイドとフロントエンドの違いを説明するならば、「サーバーサイド」とは、サーバーでプログラムの実行・管理をすることで、「フロントエンド」とは、ユーザーが見ている画面のデザインである。

　サーバーサイドエンジニアの仕事内容は、サーバー側で実行する処理に必要なプログラムを開発がメインの仕事内容です。Web 上でユーザーが操作したときに操作内容に応じたプログラムの開発と保守をおこなう。

他にもサーバー側で扱う顧客情報などのデータ管理やデータベースにあるファイルへのアクセスなど、目に見えない処理を開発するのがサーバーサイドエンジニアである。

　具体的には、Web アプリなどの会員登録機能やログイン機能、自動メール配信機能などがサーバーサイドの開発業務です。経験を積むことで、開発業務だけでなくシステム全体の要件定義や設計などより上流工程の仕事を行うことが多くなり、サービス全体の機能に関わるところなので、フロントエンジニアや顧客と接する営業側の要望などをヒアリングし開発することが一般的です。

サーバーサイドというとサーバールームで一人コツコツという訳ではなく、よ

いサービスを開発するために多くの人とコミュニケーションをとりながら業務である。

フロントエンジニアの仕事は、Web デザイナーが作ったデザインをともに画像ファイルなどを組み合わせながら、コードを記述し Web ページを作成することで、作成が終わるとサーバーにデータをアップロードし、表示崩れの有無やサーバーと通信できるかを確認をしてリリースをする。

実際の Web サービスの開発では、Web デザイナーやコードを書く人だけでなく、サーバーサイドエンジニアや Web ディレクターと言われる、進捗管理や仕事内容の調整をする人とやり取りをしながら作成をしている。

Web サービスでは Web サイトの見た目や使い心地が重要視されており、理由はユーザーが見ている画面の印象や、使いやすさから生まれる良いユーザー体験が売上やブランドイメージにつながるからです。これを UI/UX といったりもする。

ただのフロントエンドエンジニアではなく、ユーザーの使い心地や Web サイト上でのユーザー体験を向上させることができるフロントエンドエンジニアが重宝されているのである。

年収はサーバーサイドエンジニアで 400 万～800 万、フロントエンドエンジニアは約 300 万～700 万である。（参照元：indeed）

参考にしたリンク

https://www.sejuku.net/blog/100525

４．　AWS とは何か

AWS（Amazon Web Services）は、Amazon が提供するクラウドサービスである。2006 年から販売が開始され、クラウドコンピューティング業界で大きなシェアを誇っている。

まず、クラウドコンピューティングとは、インターネット経由でデータベース、ストレージ、アプリケーションなど様々な IT サービスをオンデマンドで利用できるサービスのことである。

従来、ハードウェアを導入する際に必要であった多額の初期投資やリソースの調達、メンテナンス、容量の仕様計画などがクラウドサービスを利用することで削減でき、人的リソースを費やす必要がなくなるのである。

Amazon が提供するサーバー、ストレージ、データベースなど 200 以上のフル機能サービスをクラウド上で管理することができるサービスで、Amazon が 15 年以上提供しており、シェアも非常に大きなクラウドサービスとなっている。

大きなメリットがいくつかあり、AWS は、軍隊、国際銀行など気密性の非常に高い組織のセキュリティ要件を満たすことができるよう非常に高いセキュリティレベルで構築されているところである。

また、初期費用ゼロで導入することができ、利用量に応じた従量課金型なので導入コストを削減することができる。簡単にサーバーの台数やスペックの変更が可能で、柔軟性や拡張性が高いところも魅力的だ。

加えて、AWS はデータセンターを世界中に設置していることから、データ障害や大規模災害が起きた際にリスク分散ができる。AWS では定期的にハードウェアのメンテナンスや最新のハードウェアへのアップデートが行われているので、安心して管理・運営を任せることができる。

他にも、サーバーの準備から管理・運用を AWS が担っているため、これまで必要だった人的リソースを割く必要がなくなり人的リソース不足を解消することができるのである。

メリットだけでなくもちろんデメリットも存在し、「毎月の利用料が変動する」「自由度が低い」「ダウンタイム対応を考えなければならない」など、AWS だからこその声もある。AWS には独特の概念や考え方があり、AWS を利用するためには、それを学ぶ必要があり、もし AWS の障害など何らかの問題が発生した時には自前で対処しなければならないのである。

これらを踏まえ、AWS を学び自分の作りやすい環境を構築していくことが出来れば、自身の開発の幅も広がるだろう。

参考にしたリンク

https://www.yume-tec.co.jp/column/awsengineer/4497

５．　Docker とは具体的になにが出来る技術か？またそれを導入するメリットとは

Docker とは、コンテナ仮想化を用いてアプリケーションを開発・配置・実行するためのオープンソースソフトウェアあるいはオープンプラットフォームである。

コンテナ仮想化を用いた OS レベルの仮想化によりアプリケーションを開発・実行環境から隔離し、アプリケーションの素早い提供を可能にする技術である。

メリットとして、OS は既に共有して利用しているため、それらの設定は省き、システムを構築するために必要となる最低限のプログラムしかインストールする必要はなく、すぐにプログラムを適用し、導入までのスピードを高速化することができる。更に、完全仮想化であれば、サーバーを起動する際に OS

レベルから立ち上げていく必要があり時間がかかるが、Docker では OS は既に立ち上がっているため、その分サーバーの起動時間を短縮化することができる。また、リソースの使用量が少なく済むためサーバへの負荷が低く、1 度に多くのプログラム処理を実装することが可能である。

1 度作成したコンテナは、Docker イメージを作成し、他のコンテナへ適用することにより再利用が可能となり、検証してみたい時や、リソースを拡大したい時など、すぐに同じ環境設定が適用されたコンテナを準備することができるのである。

参考にしたリンク

https://udemy.benesse.co.jp/development/system/docker.html