

1.

まず、オブジェクト指向プログラミング(Object Oriented Programming：以降 OOP)はプログラミングに関する一つ概念である。

OOP とはプログラミングにおけるプログラムを手順ではなく、モノの作成と操作をするものとする考え方。オブジェクトとはモノを意味している。

例えば、横スクロール型アクションゲームをプログラミングするとする。

各選択したキャラはそれぞれ A ボタンを押したらジャンプ、B ボタンでダッシュするというプログラムにする時にキャラクターそれぞれにコードを書いていたらきりが無い。

また、操作方法が変更になった時にはまた手間がかかってしまう。

そこで OOP の出番だ。

キャラクターは A ボタンでジャンプ、B ボタンでダッシュすると定義してしまえばいい。操作方法が定義されたキャラクターというモノを決めることでコーディング作業の簡略化や様々な変更にも対応ができる。この同じような動きで分類されたモノのことをクラスと呼ぶ。

そして OOP には特徴が 3 つある。

「継承」、「カプセル化」、「ポリモーフィズム」の 3 つの特徴がある。

継承とは同じ特徴を共有している関係のことを呼ぶ。オブジェクト指向では、実装の際に抽象クラスが持つ機能を具体クラスに継承することができる。

次にカプセル化とは機能とデータの範囲を明確にすることで、オブジェクトにカバーを被せて中身を隠してしまうようなこと。なぜ隠すかというと、オブジェクトが持つ属性を勝手に書き換えたり、属性を直接参照したりできないようにするため。

最後にポリモーフィズムについて、ポリモーフィズムとは多様性という意味で、同じ関数だとしても引数によって違う処理を行えるもの。

これらの特徴があることによって、プログラムのメンテナンスや分業化が見やすくなる。

2.

GitHubFlow とは GitHub におけるブランチ戦略の一つである。開発現場における開発フローであり、リポジトリの分岐モデルとも言える。

複数人での開発作業時に作業範囲の明確化ができ、混乱を防ぐ目的で使用される。

また、作業内容が履歴として残るため、誰がいつ編集したか追跡が可能。

おおまかな手順としては GitHub 上にリモートリポジトリを作成。それをクローンしてローカルで編集していく。

開発において、メインブランチが存在する。これは例えるなら原稿みたいなもので共同開発者達はメインブランチから各担当の部分を切り取って編集していく。

編集が終わればメインブランチに合流（マージ）させるために、共同開発者に確認を行う必要がある。これをプルリクエストという。

この流れで開発を行うことを GitHubFlow と呼ぶ。

3.

フロントエンジニアはユーザーが見ている画面のデザインをコーディングしていくエンジニアで、サーバーエンジニアは必要な情報を持っているサーバーでプログラムの実行・管理をしている。フロントで要求されたデータ（リクエスト）を受け渡す（レスポンス）のがサーバーエンジニアの仕事になる。

4.

AWS（AmazonWebService）はクラウドサービスの一つである。クラウドサービスとはインターネットを介してサーバー・ストレージ・データベース・ソフトウェアといったコンピューターを使った様々なサービスを利用することを指す。

クラウドサービスを使用するメリットは以下が挙げられる。

- ① 必要な時に必要な分だけ使用できる。
 - ② サーバー機器を購入する必要がないのでコスト面で安価。
- そしてその中でも AWS には特徴が 4 つある。
- ① 約 100 以上の豊富なサービスが有る。
 - ② 初期費用がかからず、必要な分だけなので安心・安価。
 - ③ 常に最新のセキュリティに保たれているので安心。
 - ④ 定期的なアップグレードによる快適なパフォーマンス。

以上、AWS はこのような特徴があるクラウドサービスだ。

5,

Docker とはコンテナ型の仮想環境を構築できる OSS（オープンソースソフトウェア）のこと。仮想環境とは、一つのハードウェアの中に元々入っている OS 以外の仮想的な OS 環境を構築したもので、実際のハードウェアの CPU やメモリを割り当てて仮想環境を分離・統合する技術。土台となる OS をホスト OS、仮想環境の OS をゲスト OS をと呼ぶ。コンテナ型とはホスト OS 上にアプリケーションに必要なものをひとまとめにして独立空間を形成しているもので、ホスト OS のカーネルを利用し、動作はコンテナエンジン上で行われる。Docker を利用するメリットは簡単に環境構築ができること、軽量でスピーディーに使用できる、インフラをコード化できることが挙げられる。

【参考文献】

サーバーエンドとフロントエンドの違い：<https://www.sejuku.net/blog/100525>

オブジェクト指向とは？：<https://www.vacslab.co.jp/object-orientation/>

AWS とは？：<https://www.skyarch.net/column/whataws01/>