

1. オブジェクト指向について

オブジェクト(モノ)と操作に分けてプログラムを組み立てていく考え方のこと。

学ぶことのメリットとしては下記がある。

- 1) 効率よくプログラムを開発、設計できること
- 2) 物と操作にコードが分かれているので、不具合の原因が特定しやすくなる
- 3) プログラムの仕様が変わっても対応がしやすい

■特徴

・継承：なくてはならない共通特徴を共有している関係

Ex) 食べる、生きている、動くなど ⇔ 生物 > 哺乳類 > ニンゲン > 会社員 > 技術員・事務員

・カプセル化：複雑な仕組みを持つ「もの」を抽象化する作業。

(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カプセルのようにかぶせてふわつとさせる) カプセル化をすることでデータや機能の範囲が明確になるので知りたいところだけをカプセル化するとよい。

・ポリモーフィズム：日本語では「多能性・多様性」と訳される。すべてのパーツが一つの指示で、あらかじめ定めた動作をするような仕組みを作ること

■具体例

・web アプリケーション上でボタンを作成する

└モノ：1 つボタンを作る 操作：登録ボタン、更新ボタン、削除ボタンを色や文字を変えて作る

・カレー作り

└モノ：カレーを知っている人 操作：カレーの作り方を共有する(野菜を切る、肉を切る、炒める、煮る、ルーを入れる)

・TV リモコンづくり

└モノ：リモコン本体を作る、操作：電源 off やチャンネルの変更をする

■引用：

・侍エンジニア blog

<https://www.sejuku.net/blog/5973>

・CodeZine

<https://codezine.jp/article/detail/10215>

・Qiita

<https://qiita.com/Nossa/items/b6e2f4ed0fa079359fc5>

・ワカテク

https://wakatech.jp/object_oriented

2. Github flow とは何か

GitHub 社が提唱している効果的なワークフローのこと。

1. ローカルリポジトリ(ファイルやディレクトリの状態を記録する場所)の作成

└この時に main ブランチを常に展開できる状態にしておく

2. ブランチ(履歴の流れを分岐して記録していくためのもの)を作成し、そのブランチで自身が作業する

3. 作成したローカルリポジトリのブランチに合流させる

4. リモートリポジトリで自分の手元のローカルリポジトリの変更履歴を共有するために、定期的にプッシュする

5. メンバーからのアドバイス助けてほしい時やフィードバックがほしいときはプルリクエストを作成し、プルリクエストでやりとりする

6. 他の開発者がレビューし、作業終了を確認したら main ブランチにマージする

7. master ブランチへマージしたら、直ちにデプロイする

補足：

- ・リモートリポジトリ

専用のサーバに配置して複数人で共有するためのリポジトリ

- ・ローカルリポジトリ

ユーザー一人ひとりが利用するために、自分の手元のマシン上に配置するリポジトリ

- ・ブランチ：作業スペース、コードを乗せたトラック

■引用

- ・サル先生の Git 入門

<https://backlog.com/ja/git-tutorial/>

- ・Think it

<https://thinkit.co.jp/article/8410>

- ・TEC CHAMP BLOG

https://tech-camp.in/note/technology/15150/#Github_flow

3. サーバーサイドエンジニア・フロントエンジニアの違い

サーバーサイドエンジニア

サーバー側で行う処理に対して必要なプログラムの開発や、データを扱う管理を行うエンジニア

フロントエンジニア

Web 開発における UI/UX→ユーザーの目に見える部分の実装を行うエンジニア

■引用

- ・エンジニア就活

<https://engineer-shukatu.jp/column/archives/47227>

- ・Acrovision

<https://www.acrovision.jp/career/?p=2826>

- ・侍エンジニア blog

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

4. AWS とは何ですか。特徴を述べてください。

Amazon Web Service の略で、amazon が提供するクラウドコンピューティングサービスのこと。

自社の建物にサーバーを設置して運用するオンプレミス環境とは違い、インターネットからクラウドコンピューティングサービスに接続しさえすれば、すぐにでも必要なサーバーなどのリソースが使える。

※補足：クラウドコンピューティング

└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。クラウドコンピューティングでは、手元に 1 台の PC とインターネットに接続できる環境さえあれば、サーバーや大容量のストレージ、高速なデータベースなどを必要な分だけ利用できる。

■特徴

- ・サービスが豊富(役 100 種類以上)

└サーバー環境構築やデータ保存/コンテンツ配信、DB の利用、

- ・初期費用が掛らず、必要な分だけコストをかけたらい

- ・常にセキュリティが最新の状態であるため安心

- ・定期的にアップグレードされているためハイパフォーマンスである

■引用

- ・aws 公式サイト

<https://aws.amazon.com/jp/what-is-aws/>

- ・SKYARCH の IT あんちょ

<https://www.skyarch.net/column/whataws01/>

5. Docker とは具体的に何ができる技術か。また、Docker を導入するメリットとは

- ・Docker とは、「コンテナ型」の仮想環境を構築(作成、配布、実行)できる OSS(オープンソースソフトウェア)のこと。

■導入のメリット

- ・Docker コンテナはホスト OS のカーネル(OS の中心核)を利用するので軽くて速い
- ・Docker イメージがあれば簡単に環境構築ができる
- ・Dockerfile を使うと構築手順をファイルにまとめられる

■具体的に何ができるか

- ・もしも Apache + PHP+ MySQL の web サービスを作りたいと思った時に、通常ならそれぞれ個別に PC にインストールし実行環境を構築しなければならないが、Docker の場合は Apache、PHP、MySQL、それぞれを単なる Apache コンテナ、PHP コンテナ、MySQL コンテナといった具合に、コンテナという単位で起動・停止すれば実行環境の構築は完成でき、全ては Docker というシステム内での話なので、OS に依存しない環境構築が可能というメリットもある。

→OS 内に多面の実行環境を構築できるため多種多様の開発検証環境や開発環境の変更に柔軟に対応が可能

■引用

- ・SIOS TECH LAB

<https://tech-lab.sios.jp/archives/18811>

- ・SKYARCH の IT あんちょ

<https://www.skyarch.net/column/docker/>

- ・ちゃんとメモったか？

<https://hara-chan.com/it/infrastructure/docker-about/#Docker>