

5. 概念論述

1. オブジェクト指向とは、システム全体を「モノ」に見立てた「オブジェクト」の集合体として考えることで、より人間が分かりやすい構造を保つための仕組みです。オブジェクト指向に基づいてプログラミングを行うことを「オブジェクト指向プログラミング」と呼びます。

①オブジェク:

オブジェクト指向の基本概念として、システムとして考慮すべき対象を「モノ＝オブジェクト」としてとらえています。

「学校」がひとつのシステムだとすると、「教室」「校庭」「下駄箱」など、大小さまざまなオブジェクトが考えられます。

②クラス：オブジェクトはあくまでも概念的な存在であるのに対し、クラスはプログラム上でオブジェクトを表すための「設計図」です。

現実世界においては、同形の車がいくつも存在していますが、それぞれ違う個体として存在しています。

オブジェクト指向プログラムの世界においては、このクラスの情報をもとにオブジェクトの実体が生成されます。プログラムの世界においては、クラスから生成されたオブジェクトを「インスタンス」と呼びます。

例)「車」のクラスがあるとする、同じ「クラス」のオブジェクト(=インスタンス)には、「赤色の車、白色の車、カスタムされた車」などがあります。

③プロパティ：プロパティは、そのオブジェクトが持つ「情報」を意味します。

例) 車であれば「色」や「ドア数」などといった情報がプロパティです。

プログラム上では、文字列や数字といった情報がプロパティとして格納される他、ほかのプロジェクトをプロパティとして保持することもあります。

④メソッド：メソッドは、そのオブジェクトが持つ「ふるまい」を意味します。

例) 車であれば、「カギを開ける」「エンジンをかける」「アクセルを踏む」などといった、操作に関する事項がメソッドです。

2. GitHub Flow とは、Git・GitHub を利用した開発方法の一つで、GitHub 社が使用しているワークフローのことです。GitHub 社が自社で使用しているワークフローを GitHub Flow と名付けています。

GitHub Flow の流れは、

ローカルレポジトリとリモートレポジトリ (GitHub) があり、

- ① まず、main ブランチからブランチを作成します。
- ② ブランチを作成したら、ファイルを変更し、コミットします。
- ③ コミット出来たら、作成したブランチと同じ名前のブランチを GitHub へプッシュします。
- ④ GitHub へプッシュ出来たら、プルリクエストを送ります。
- ⑤ プルリクエストを送ったら、コードレビューをし、main ブランチにマージします。
- ⑥ main にマージしたら、その main ブランチをすぐに本番サーバーにデプロイ (リリース) します。

3. フロントサイドエンジニアとは、Web 開発における UI や UX の実装を行うエンジニアです。また、フロントエンジニアは主に HTML や CSS、JavaScript などを使用し、Web ユーザーの目に見える部分を実装します。クライアントの要望を反映させた Web サイト開発を行います。それに対して、サーバーサイドエンジニアは、ユーザーの目に見えない部分の制作に関わっており、サーバー側のプログラムの実装やデータ処理などを行います。

4. AWS とは、Amazon Web Services 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です。近年よく使われているクラウドサービスの一つです。

また、クラウドコンピューティングとは、インターネットを介してサーバーやストレージ、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します。

クラウドコンピューティングサービスとは、必要なときに必要な分だけ利用することが可能で、世界のクラウド市場には Microsoft や google などがあります。物理サーバーを設置して運用する、「オンプレミス」という運用形態で運用され、サーバー機器を購入、管理し、設置するスペースを考慮する必要があります。

AWS のメリットは、約 100 以上の豊富なサービスがあるので、システム構築や運用に必要な様々なサービスも提供されている。また、コスト面に関しても、初期費用がかからず、利用した分だけ料金を支払う形なので、低コストで利用できる。そして、常に最新のセキュリティを搭載されていて、定期的にアップグレードされているので、サーバーの質が常に高く保たれている。さらに、クラウド市場のシェア率

がトップで、現在大小様々な企業のウェブサイトやウェブサービスで利用されているので、導入事例を目にすることも数多くあります。

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) のことです。

ホスト OS 上にアプリケーションの起動に必要なアプリケーション本体・必要なライブラリ・設定ファイルなどを「ひとまとめにした箱」のようなものを「コンテナ」といいます。動作させるときは「コンテナエンジン」を使い、ホスト OS のカーネルを利用しているので、少ないリソースでのアプリケーション実行が可能な技術です。そのため、メモリや CPU リソースを余分に使う必要がなく、処理が早くなる、といった特徴があります。

また、仮想環境とは、一つのハードウェアの中に仮想的な環境を構築したものです。実際の CPU やメモリなどを割り当てて、環境を分離したり統合したりする技術です。そのため、一つのハードウェアで、一元管理できるようになります。

Docker のメリットは、

- ① Docker コンテナはホスト OS のカーネルを共有して使うので軽量で起動が高速です。
- ② Docker イメージを共有することで、誰でも簡単に同じ環境構築ができます。
- ③ インフラをコード化できるので、Dockerfile を使えば構築手順を分かりやすくファイルにまとめられ、管理できます。

引用

: <https://camp.trainocate.co.jp/magazine/whats-object-oriented-programming/>

引用: <https://phoeducation.work/entry/20210914/1631573880>

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