

## 課題 1

オブジェクト指向プログラミングとは何かを述べてください

オブジェクト(モノ)中心に考えたプログラミングスタイルです。

オブジェクト(モノ)中心に考えるため、プログラムに落とし込む際はオブジェクト(モノ)属性(プロパティ)と手続き(メソッド)がどういったものがあるかを知る必要があります。

a. 特徴3つ、また、その説明を含めてください

特徴3つ

- ・継承

継承は既存のクラスの内容を引き継いで、新たな別のクラスを定義すること。

- ・カプセル化

カプセル化は、クラスのメンバーの中で、クラスを使う人に見せる必要のないものを隠すこと。

- ・ポリモーフィズム

多態性は、同じメッセージを与えたときに、それを受け取るオブジェクトによって異なる振る舞いをすることです。

b. 具体例を含めてください

- ・継承

銀行の窓口業務を1つのクラスとして考えた場合、窓口業務クラスを継承して、ATMというクラスを作成することにより、預金残高やお金の出し入れができるようになり、窓口業務時間外でも対応してもらえます。

- ・カプセル化

銀行に登録しているユーザーを1つのクラスで考えた場合、預金額を本人以外に見せないようにしています。

- ・ポリモーフィズム

各銀行のATMに別の銀行のカードを入れると、ATMと異なる銀行からでもお金を引き出す事ができます。

これはポリモーフィズムによるものです。

## 課題 2

Github flow とは何かを述べてください

GitHub Flow とはブランチ戦略の 1 つで、リポジトリをブランチごとに分散させる事により、各ブランチごとに並行して開発しても影響を受けません。

また、運用はネットワーク上のサーバにあるリモートリポジトリで行われ、安定したブランチを main ブランチといいます。

## 課題 3

サーバーサイドエンジニア・フロントエンジニアとはどのような違いがあるかを述べてください。

フロントは人が目にする画面や機能を指し、その部分を作る人をフロントエンドエンジニアと呼びます。

サーバーサイドは、フロントと違い、人が目にする事が無いサーバー側のことを指し、その部分を作る人をサーバーサイドエンジニアといいます。

#### 課題 4

AWS とは何ですか。特徴を述べてください。

AWS は、世界でもトップクラスのシェア率を誇っているクラウドコンピューティングサービスです。

#### 特徴

##### ●従量課金制

AWS は初期費用が無料な上、料金体系は従量課金制を採用しています。サービスを利用した時間や通信量に応じた費用だけを払うため、使っていないサービスにもコストがかかってしまうことはありません。

タイミングに応じて必要な機能を追加・停止させて、料金を変動させたりリソースを調整したりすることも可能です。

時間単位での課金になるため、サービスの検証や実験的なプロジェクトなどにも活用できます。

##### ●必要なサービスのみを選べる

AWS はたくさんのサービスを提供しています。

たくさん存在するサービスを必要に応じて必要なだけ柔軟に組み合わせて、利用することが可能です。

##### ●セキュリティが高い

AWS は、もともと Amazon が「自社サーバとして利用していたもの」を提供しているサービスです。

そこから、クラウドコンピューティングサービスとして提供され始めたのが、AWS の始まりとなっています。

そのため、Amazon が持つ高いセキュリティレベルを AWS でも保っており、その高い水準は第三者にも認証/認定されております。

##### ●パフォーマンスに優れている

AWS は、世界の 11 エリアに地域にサーバを設置しています。

ユーザーに最も近い設備からサービスが提供されているため、常に高いパフォーマンスを保ったままサービスを利用できます。

さらに、ハードウェアの更新も定期的に行われているのも特徴です。

●安定したネットワーク

AWS の特徴としては、ネットワークの安定性と可用性（システムが継続して稼働できる度合いや能力）の高さも挙げられます。

複数の AZ（アベイラビリティゾーン=データセンター）の構成などの対策により、非常に高い耐障害性を誇っています。

## 課題 5

Docker とは具体的に何ができる技術ですか。また Docker を導入するメリットを述べてください。

IT ソフトウェアの「Docker」とは、Linux コンテナの作成と使用を可能にするコンテナ化テクノロジーのことです。

コンテナとは一般的な意味では『(内部に物を納めるための)容器』ですが、ここでのコンテナ型とは『ソースコードやそのすべての依存関係をパッケージ化するソフトウェア』を示します。

### Docker 利用のメリット

簡単に環境構築でき、軽量かつ素早い開発ができ、ハードウェアの資源削減やシステムの共有化も可能です。

### Docker のデメリット

同一の OS（カーネル）を利用するため、提供できるホスト OS の種類が少ないことや、Docker の習得に時間がかかります。