

- コミュニケーション図とはオブジェクト間の一連のやり取りを視覚的に表現したもの
  - オブジェクト：現実のユーザーや利用しているシステムなどシナリオにすべて関わっているもののこと
  - メッセージ：リクエスト・レスポンスをユースケース表記したもの。データが含まれたり、しなかったりする
  - シナリオ：特定のユースケースに基づいた一連の流れのこと
  -

影響範囲が

外部のコードからの呼び出しでも、エラー原因がそのオブジェクトの中だけであるから、修正するのはメソッドの中身だけであり、メソッド名は変わらないので、呼び出し側でも変更する必要がないということ

開発者や該当するオブジェクトを生成するプログラムの実装を行なっているテスターや他の開発者などからの直接的な操作を防

振る舞いの中身も見えないので

メソッドの中身が見えないこととカプセル化に関連はあるのですか？

データにも触れない、  
マニュアルにあるメソッドだけを使わせる

カプセル化は

特定の機能のみを強制させることでどんな利用者にも一貫した機能を提供することができるということですか？

誰が利用しても終始変わらない

定義した特定の操作からでしか紐づいたデータの操作ができない  
→

直接インスタンスのフィールドにアクセスできることはなぜ問題ですか？

外部からインスタンスを直接扱えることは、定義しているフィールドやメソッドを変更したりできるのですか？

- —————

終始変わらない動作を保証するために、適した値を入力するための処理が必要になるということと、そのために直接インスタンスからフィールドの操作を避けるためにカプセル化を施すのですね

インスタンスの直接操作を防ぐ必要がある

—————

- オブジェクトが正常に扱えなくなってしまうためのプロテクトのようなもの
- 処理によっては定義されてい