



インターフェース

応用編5日目

インターフェースとは

- 抽象クラス概念をさらに推し進めたもの
- 抽象クラス同様**インスタンスを生成できない**
- 原則的に記述するのは**抽象メソッドのみ**
- 使用方法などは抽象クラスとは根本的に異なる
- 既存のクラスに追加していく
- **一つのクラスに複数の側面がある場合**によく用いられる
- **SampleEx401**参照

インターフェースの利用方法

- インターフェースの定義と実装

インターフェースの定義

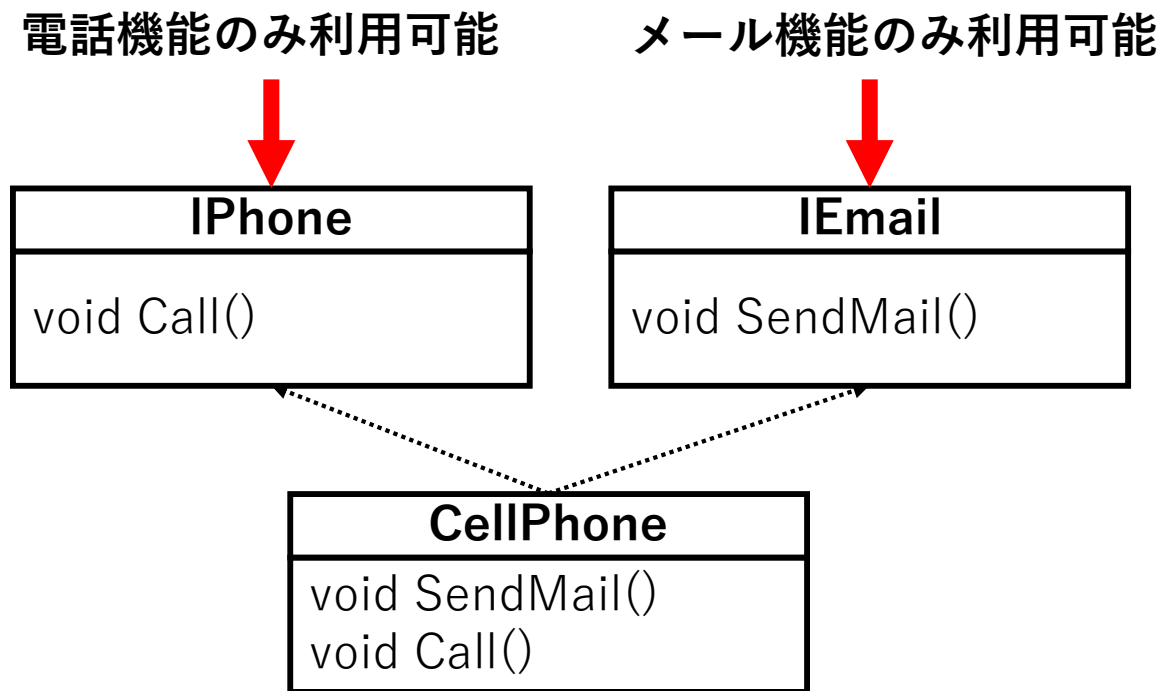
```
interface (インターフェース名) {  
    ...  
}
```

インターフェースの実装

```
class (クラス名) : (インターフェース名),(インターフェース名)…{  
    ...  
}
```

- 継承と違いインターフェースは**複数定義することができる**
- その場合、インターフェース名の間を“,”で区切る

クラスとインターフェースの関係性



インターフェースの必要性①

- 大規模なシステム開発（スマートフォン開発など）
- 大勢のプログラマーが一つのクラスを操作するケース
 - 電子メールの機能 … **IEmail**のメンバを使えば十分
 - 電話の機能 … **iPhone**クラスのメンバが使えば十分
- 無関係の人が無関係の部分进行操作する危険性を減らせる
 - バグなどのトラブルの軽減
 - 役割分担の明確化

インターフェースの必要性②

- クラスのバージョンアップ
- 旧来のクラスとの互換性を持たせたまま中身を変更したい
- インターフェースを同一にし、ユーザーにはインターフェースのみを他わすことにより、バージョンアップを意識させない。