

Polimorfismo por sobreposição: É um caso de herança entre duas classes na qual a classe filha apenas reimplementa o comportamento do método sendo alterado (sem mudança de interface). Isso significa que os parâmetros de entrada e saída do método não são alterados.

Polimorfismo por sobrecarga: Nesse caso, a herança novamente é feita, mas não apenas o comportamento do método é sobrescrito, mas sim a sua assinatura. Isso significa que parâmetros de entrada e saída do método não são idênticos (em quantidade ou tipo, por exemplo).

Exemplo:

Obs: esse é um exemplo que eu criei, acredito que não funcione assim na prática hehehe.

Uma classe que realiza interações com uma tela touch screen e que pode ser utilizada tanto em smartphones quanto em notebooks. Ela possui um método apenas:

```
ScreenResult getScreenInput(String system){  
    ...  
}
```

Para a classe que interage com a tela de notebooks, é possível fazer um polimorfismo por sobreposição, através de herança. Nesse caso, não mudamos a assinatura do método, apenas o comportamento, descrevendo como é o comportamento de pegar o input do usuário considerando o sistema operacional.

Para a classe que interage com a tela de smartphones, não é suficiente sabermos o sistema operacional, uma vez que diferentes fabricantes possuem métodos diferentes de leitura, mesmo usando um sistema comum (android). Nesse caso, fazemos um polimorfismo por sobrecarga, passando também a fabricante e, dessa maneira, especificando não só um novo comportamento, mas uma nova assinatura pro método.

```
ScreenResult getScreenInput(String system, String manufacturer){  
    ...  
}
```