

4-Factory Method

Seeeeeee:D

楊明哲

Factory Methodって？

TemplateMethodをもうちょっと使いやすく！

- Factory Methodでは、クラス内で使うオブジェクト（インスタンス）の生成方法について工夫を行う.
- 柔軟にオブジェクトを生成することが可能になる.
- 継承を利用するときに使うデザインパターン

Factory Methodって？

実装の工夫のやりかた

- インスタンスの生成を，抽象クラス（親クラス）にするのではなく，サブクラス(子クラス)に行うことで，生成するインスタンスを簡単に変更することができる.
- オブジェクトの生成を関数を経由して行うことで実装できる.

じゃあ具体的には??

実際につくってみた

設定を確認していくよ

- 今回は料理の設定で話を進めていく.



- 料理する人は、それぞれレシピを見ながら、「切る→焼く→煮る」の行動を習得
して、調理を行う.
- レシピは、Cookpadやクラシルなど同じ食材でも異なるレシピが存在する.

初心者が調理していくよ(Not Factory Method)

とりあえずCookpadを見るか

```
class Shoshinsya_Chouri(metaclass=ABCMeta):
    @abstractmethod
    def kiru(self):
        raise NotImplemented

    @abstractmethod
    def yaku(self):
        raise NotImplemented

    @abstractmethod
    def nikomu(self):
        raise NotImplemented

    def chouri(self):
        cookpad = Cookpad(self.shokuzai)
        self.kiru(cookpad.recipe)
        self.yaku(cookpad.recipe)
        self.nikomu(cookpad.recipe)
        print("できた！")
```

親オブジェクトで
レシピを指定して

```
class Shoshinsya(Shoshinsya_Chouri):
    def __init__(self, shokuzai):
        self.shokuzai = shokuzai

    def kiru(self, cookpad):
        print(f'{cookpad}を参考にして{self.shokuzai}を切るぞ')

    def yaku(self, cookpad):
        print(f'{cookpad}を参考にして{self.shokuzai}を焼いていく')

    def nikomu(self, cookpad):
        print(f'{cookpad}を参考にして{self.shokuzai}を煮込むよ')
```

見ているレシピ

別にこれでよくない？

よくないよ！

何が良くないのか

Cookpadしかレシピ見ないの？



- この実装方法だと，初心者の調理方法を習得している人は，Cookpadしかレシピを見ることができなくなってしまう.
- Cookpad以外も見ろよね？他のレシピも見たいよね？？



人によってみる
レシビ^o違うよね？

いろんなレシピに対応しよう(Factory Method)

サブクラスでレシピを生成していくよ

```
class Chukyusha_Chouri(metaclass=ABCMeta):
```

```
#snip
```

```
def chouri(self):
```

```
    recipe = self.nanidetsukuru()
```

```
    self.kiru(recipe.recipe)
```

```
    self.yaku(recipe.recipe)
```

```
    self.nikomu(recipe.recipe)
```

```
    print("できた！")
```

サブクラスに
まかせる

```
@abstractmethod
```

```
def nanidetsukuru(self):
```

```
    raise NotImplementedError
```

```
class Chukyusha(Chukyusha_Chouri):
```

```
#snip
```

```
def nanidetsukuru(self):
```

```
    return Kurashiru(self.shokuzai)
```

親クラスに任せられ
たレシピを決める

Factory Methodを会得した料理人

Cookpad以外もレシピ見れるようになったよ

- 初級者の調理方法しか会得していない人は、Cookpadしかレシピを見ることができなかつた。
- しかし、中級者の調理方法を会得している人はCookpadにこだわらない調理ができる。
- 中級者ごとにそれぞれ自分が見るレシピが生成されるので、柔軟に対応できる。
- Template methodだと調理方法のクラスをレシピ数だけ変更が入る
- しかし、Factoryだとただサブクラスで変更を加えるのみ！

まとめ

Factory Methodに自由を

- Factory Methodは、Template Methodと違って、サブジェクトの生成はサブクラスに任せるようにする.
- サブクラスがオブジェクトを生成するので、サブクラスに自由度が増して色々できるようになる.
- Abstract FactoryはFactory Methodをもっと抽象的にしたもの？✧



感想

夏休みがんばるぞ！

- デザインパターン難しい
- Rustでやりたかったけど、まだ学習が足りなくて実装できなかった。次はRustでかきたいかな
- 次はPython使わないでやりたい。→ C++かRustで実装します(宣言)