

فرادرس

فراتر از یک کلاس درس
www.faradars.org

آموزش الگوهای طراحی
(Design Patterns) در
پایتون (Python)

الگوهای طراحی در پایتون

مدرس:

پژمان اقبالی

دانشجوی کارشناسی ارشد مهندسی مکانیک

دانشگاه علم و صنعت ایران

The Factory Design Patterns

Building Factories To Create Objects

فرادرس
پروژه‌های طراحی الگو

class → obj , methods

client → مدرس یا برنامه‌ای از
factory استفاده می‌کنند.



مزایای factory

1. برنامه از کوپلینگ خارج می‌شود.
2. client لازم نیست درباره‌ی نحوه‌ی پیاده‌سازی factory و ایجاد جزوه بدان و فقط باید interface را بشناسد.
3. راحتی افزاین کردن کلاس‌های جدید؛ factory برای ساخت جزوه‌ها با type دیگر.
4. factory می‌تواند از جزوه‌های موجود دوباره استفاده کند.
(reuse)

«کارخانه‌ی تولید اسباب‌بازی»

machine → تولید اسباب‌بازی → factory

CEO → client
مدیرعامل

۳ نوع مختلف الگوی Factory Pattern

1) Simple factory pattern →

این الگو ابزاری است جهت زود را بدین در مصرف کننده از این منطق جهت برود.

2) Factory method pattern →

جهت زود ؛ subclass ها و الگوها شود.

3) Abstract Factory pattern

- The simple factory pattern

factory نمک میزند به ساخت زوجه های جدید از انواع مختلف ، نه instantiate کردن آنها .

با استفاده از ABCMeta از ماژول abc
abstract class ایجاد کنیم.

Abstract class:
شامل یک یا بیش از یک روش abstract باشد.

↓
declare هر شود اما پیاده سازی نمی شود.

کلاس abstract، مقداردهی اولیه (instantiate) نمی شود و برای

پیاده سازی روش های abstract به subclass نیاز دارد.

eval ()

یک String را میگیرد (بفنوان argument)

و این str را بفنوان یک python expression ارزیابی میکند.

نتیجی آن یک جمله است.

```
if __name__ == '__main__':
```

اگر درون حلقه if، تنها زمانه اجرای شود
برنامه بصورت مستقیم run شود.



not imported

اگرما برنامه را بصورت مستقیم اجرا کنیم، اسم ماژول، -main- می شود.
↳ module name

UML diagram :

یک راه برای به تصویر درآوردن برنامه است.

→ inheritance

→ Aggregation (تجمع)

---> Dependencies

UML diagram:

یک راه برای به تصویر درآوردن برنامه

 inheritance

 Aggregation (تجمع)

 dependencies

 Association

Aggregation :

یک کلاس بصورت یک container باشد برای دیگر کلاس ها
محتویات این container از لحاظ lifecycle ، وابستگی پذیری
به container نزارند.



فرادرس

Dependencies :

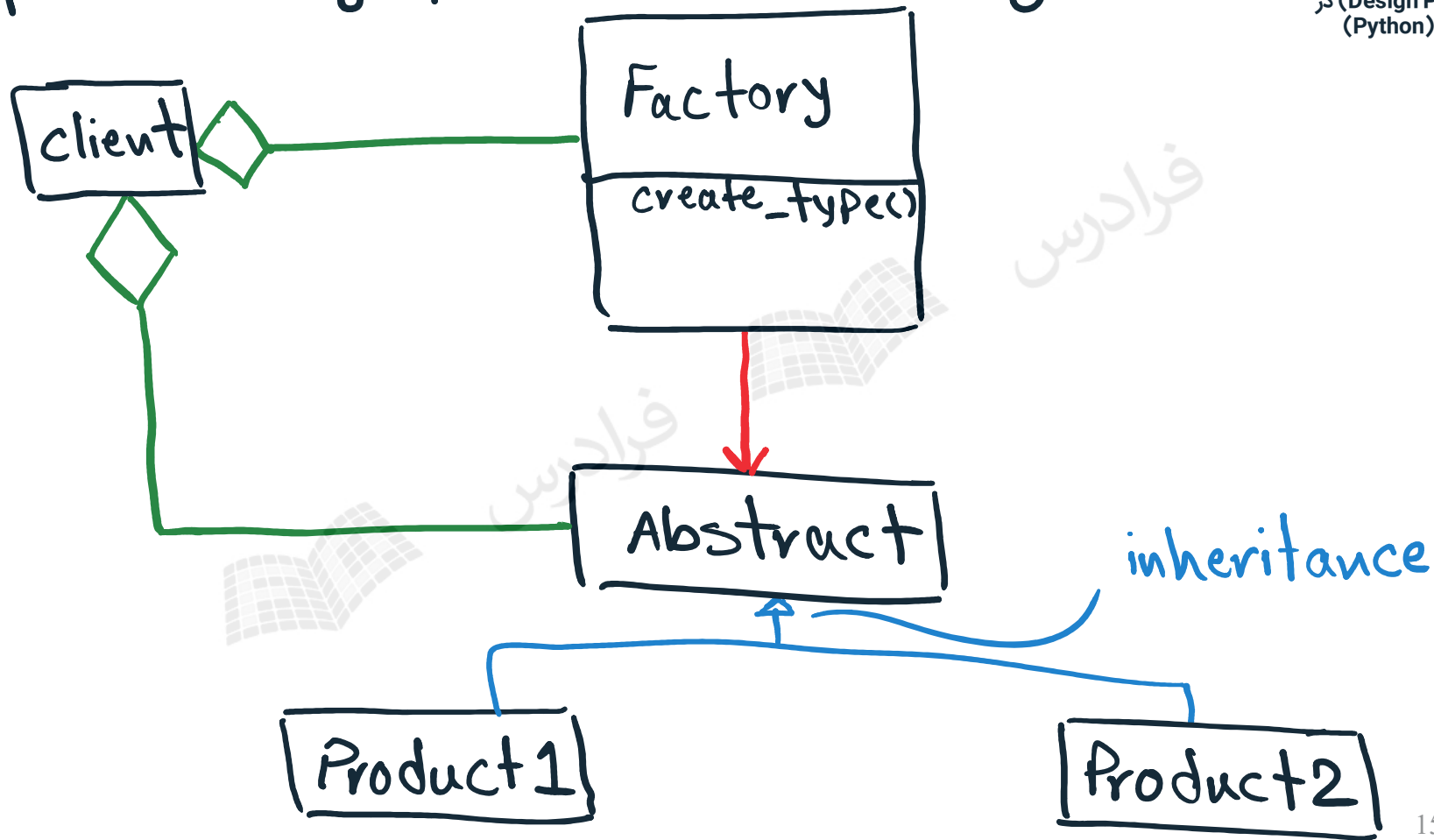
تغییر در تعریف یک کلاس، موجب تغییر در تعریف کلاس دیگر شود.

Association :

ارتباط static؛ اشتراک گذاشته شده بین رده‌های دو کلاس
(یک خانواده از کلاس‌ها)

Simple factory pattern UML diagram

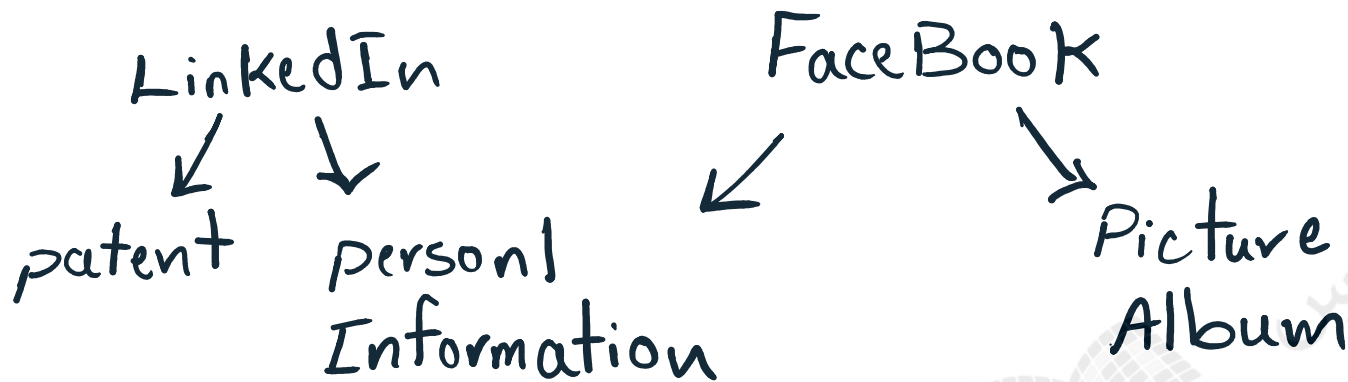
آموزش الگوهای طراحی
(Design Patterns) در
پایتون (Python)



- Factory method pattern

* مسئولیت ایجاد جزء در این الگو بر خلاف الگوی simple fac. برعهده‌ی subclass‌های factory می‌باشد.

* ایجاد جزء توسط الگوی Factory method، از طریق inheritance، instantiation.



پرفایل‌ها با type های مختلف و بخش‌ها (Section)
که به پرفایل اضافه می‌شوند.

Concrete class and Abstract class



قابل مقدار دهی اولیه
(instantiation)

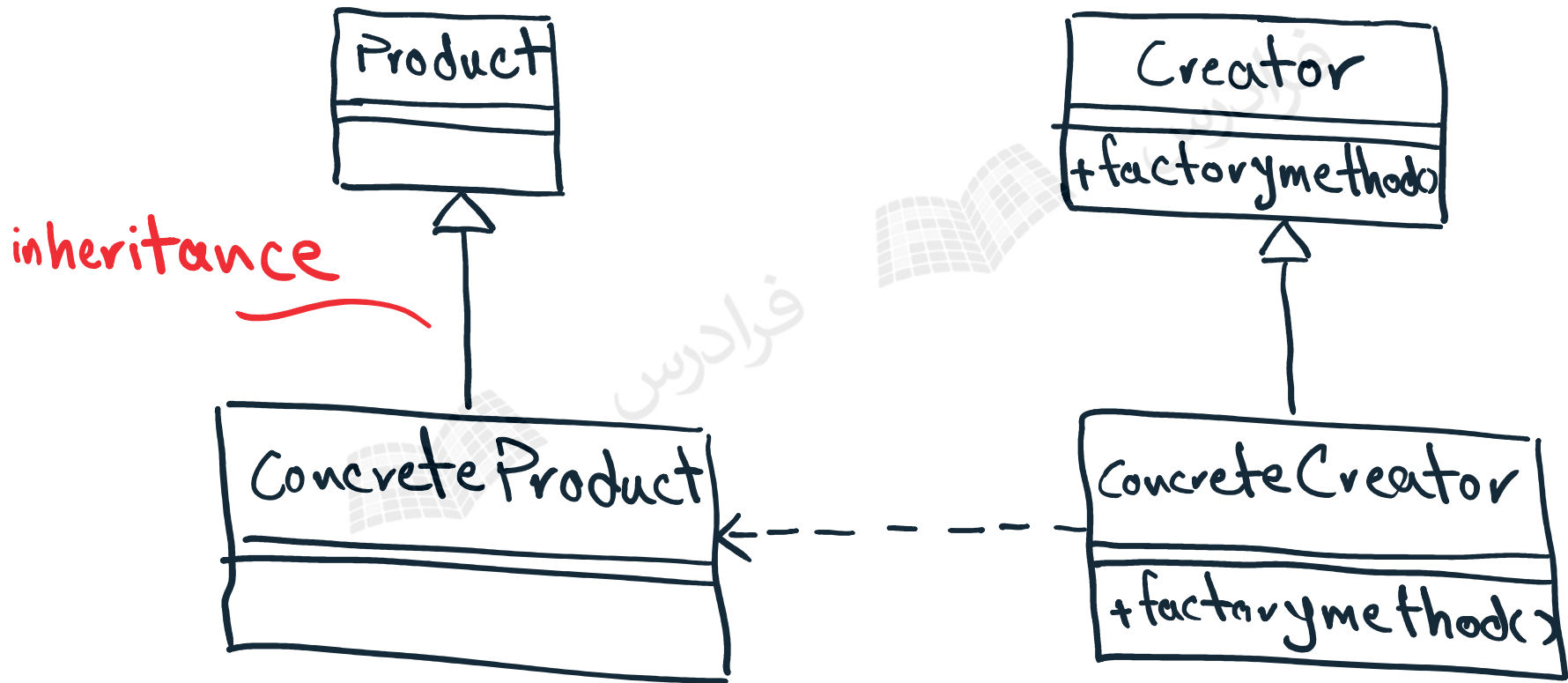
هستند.



قابل مقدار دهی اولیه
نیستند.

- Factory method pattern UML diagram

آموزش الگوهای طراحی
(Design Patterns) در
پایتون (Python)



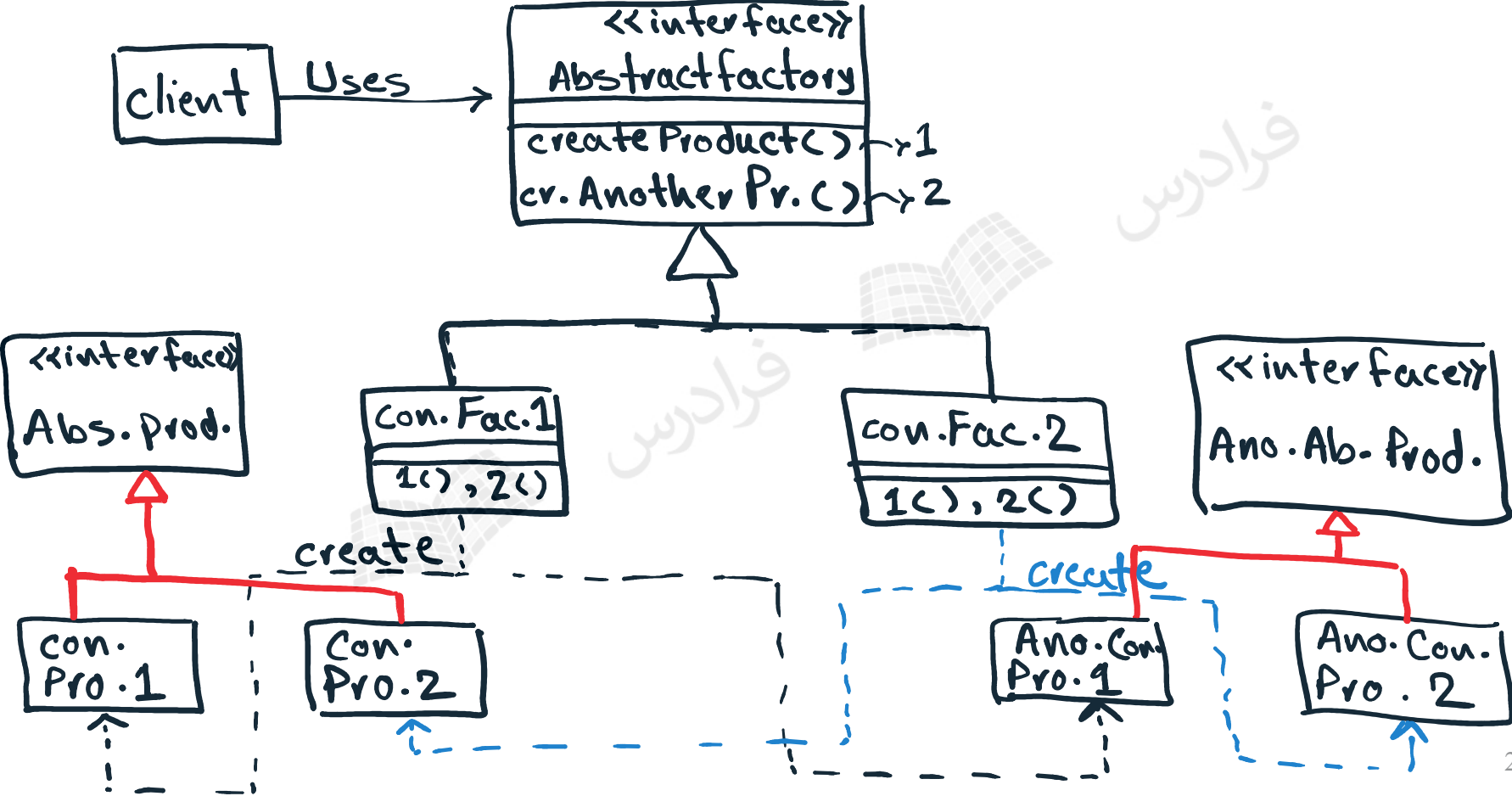
- Abstract Factory pattern

هدف: ایجاد خانواده‌ای از جزوهای مرتبط



- Abstract factory pattern UML diagram

آموزش الگوهای طراحی
(Design Patterns)
در پایتون (Python)



- Implementing Abstract Factory pattern



این اسلاید ها بر مبنای نکات مطرح شده در فرادرس
«آموزش الگوهای طراحی (Design Patterns) در پایتون
(Python)»
تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید.

faradars.org/fvpht96081