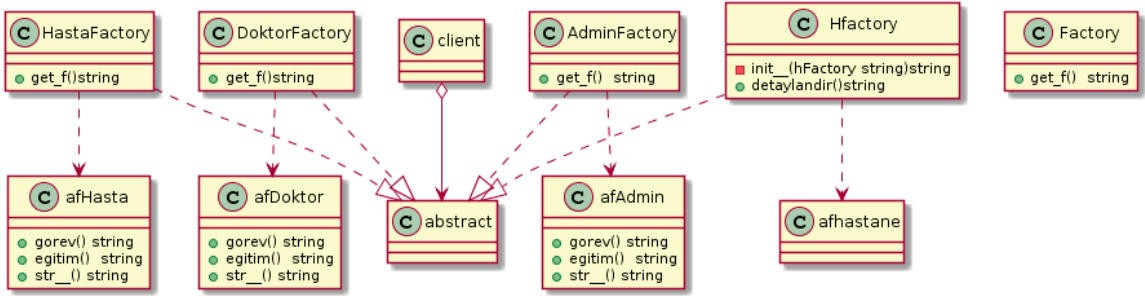


# ABSTRACT FACTORY

## DESİGN PATTERN

Abstract factory tasarım deseninin amacı; birbirleri ile ilişkili sınıfların oluşturulmasını düzenlemektir. Aralarında ilişki bulunan nesnelerin üretiminden sorumlu olan soyut fabrikaların tasarlandığı tasarım deseni olarak düşünülebilir. Abstract Factory tasarım deseni sayesinde böyle durumlarda bu nesneler her oluşturulduğunda if gibi karar mekanizması yazmaya gerek kalmaz. Abstract factory deseninin uygulanabilmesi için bu desendeki nesnelerin aynı abstract class veya interface i uygulamış olması gerekir. Bu desen üretilen nesnelerin uyguladıkları arayüzler ile ilgilendiğinden desene yeni bir nesne eklemek kolaydır.

Classes - Class Diagram



```
@startuml

title Classes - Class Diagram

class HastaFactory {

    +get_f()string
}

class DoktorFactory {

    +get_f()string
}

class afHasta

class afDoktor

class afAdmin

class abstract
```

```
{
+gorev() string
+egitim() string
+str__() string

}
class afDoktor
{
+gorev() string
+egitim() string
+str__() string

}
class AdminFactory{
+get_f() string

}
class Factory{
+get_f() string

}
class afAdmin
{
+gorev() string
+egitim() string
+str__() string
}
class Hfactory
{
-init__(hFactory string)string
+detaylandir()string

}

class afDoktor
class HastaFactory
class DoktorFactory
class afHasta
class afAdmin
class AdminFactory
class Hfactory
Hfactory ..> afhastane
Hfactory ..|> abstract
HastaFactory..|> abstract
```

```
AdminFactory..|> abstract
client o-->abstract
AdminFactory..>afAdmin
HastaFactory..> afHasta

DoktorFactory..> afDoktor
DoktorFactory ..|> abstract

@enduml
```

