

Adaptör Tasarım Deseni

In [2]:

```
from abc import ABCMeta, abstractmethod
```

In [3]:

```
class Duck:
    __metaclass__ = ABCMeta

    @abstractmethod
    def quack(self):
        pass

    def fly(self):
        pass

class Turkey:
    __metaclass__ = ABCMeta

    @abstractmethod
    def gobble(self):
        pass

    @abstractmethod
    def fly(self):
        pass
```

In [6]:

```
class MallardDuck(Duck):
    def quack(self):
        print("Quack")

    def fly(self):
        print("I'm flying")

class WildTurkey(Turkey):
    __metaclass__ = ABCMeta

    def gobble(self):
        print("Gobble gobble")

    def fly(self):
        print("I'm flying a short distance")
```

In [7]:

```
class TurkeyAdapter(Duck):  
  
    def __init__(self, turkey):  
        self._turkey = turkey  
  
    def quack(self):  
        self._turkey.gobble()  
  
    def fly(self):  
        for i in range(5):  
            self._turkey.fly()
```

In [18]:

```
def duck_test_drive():  
    def test_duck(duck):  
        duck.quack()  
        duck.fly()  
    def test_Turkey(turkey):  
        turkey.gobble()  
        turkey.fly()  
  
    turkey = WildTurkey()  
    print("The Turkey says...")  
    test_Turkey(turkey)  
  
    duck = MallardDuck()  
    print("\nThe Duck says...")  
    test_duck(duck)  
  
    turkeyAdapter = TurkeyAdapter(turkey)  
    print("\nThe TurkeyAdapter says...")  
    test_duck(turkeyAdapter)
```

In [19]:

```
duck_test_drive()
```

The Turkey says...
Gobble gobble
I'm flying a short distance

The Duck says...
Quack
I'm flying

The TurkeyAdapter says...
Gobble gobble
I'm flying a short distance
I'm flying a short distance
I'm flying a short distance
I'm flying a short distance
I'm flying a short distance