

Iterator UML

1. Iterator UML

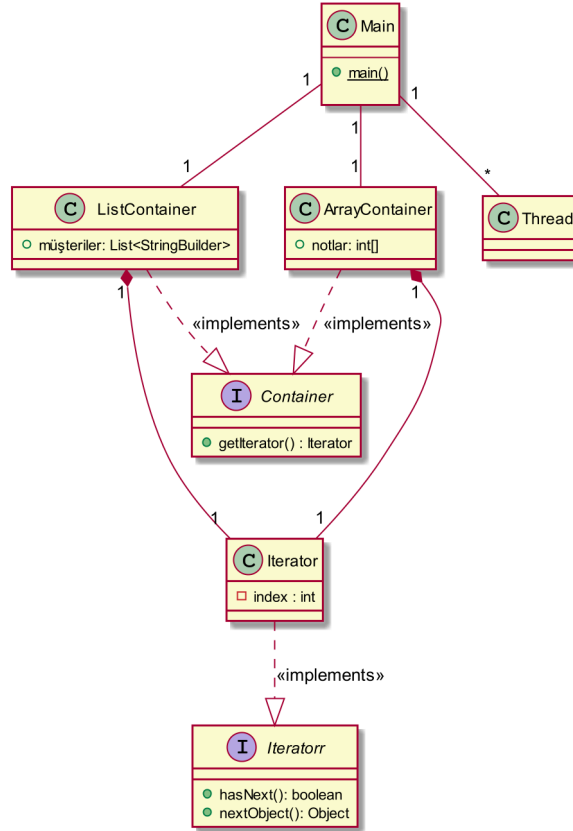


Figure 1. Iterator Uml

2. Iterator Tasarım Kalıbı Kod İncelemesi

```
public interface Iterator { ❶

    public boolean hasNext();

    public object nextObject();

}
```

```

public interface Container { ❷

    public Iterator getIterator();

}

public class ListContainer implements Container {❸

    public List<StringBuilder> müşteriler = new ArrayList<>();

    @Override
    public Iterator getIterator() {❹
        return new Iterator();
    }

    private class Iterator implements com.company.Iterator { ❺

        private int index=0;

        @Override
        public boolean hasNext() { ❻
            if (index< müşteriler.size()){
                return true;
            }else{
                return false;
            }
        }

        @Override
        public Object nextObject() {
            if (this.hasNext()){ ❼
                return müşteriler.get(index++); ❽
            }else{
                return null;
            }
        }
    }
}

```

```
}

```

- ❶ İteratörün interface kısmıdır. İteratörü kalıtım yoluyla alacak sınıfların sahip olması gereken sınıflar ve dönüş tiplerini göstermektedir.
- ❷ Container interfacesidir. Bu interface'yi implement methoduyla kalıtım alan sınıfların override etmesi gereken fonksiyonların kalıtımı sağlanır.
- ❸ ListContainer Container interface'ini kalıtım alan bir sınıftır.
- ❹ Kalıtım aldığı için getIterator() fonksiyonu override edilir. Iterator tipinde bir nesne döndürür.
- ❺ İterator adında interfaceden kalıtım sağlayan İterator adındaki sınıf
- ❻ override yaparak hasNext() fonksiyonu ile Listedeki bir sonraki elemanın olup olmadığı kontrol ediliyor.
- ❼ Listenin bir sonraki elemanı var mı diye kontrol eder.
- ❽ Listenin indexteki elemanını döndürür ve daha sonra indexi artırır.

