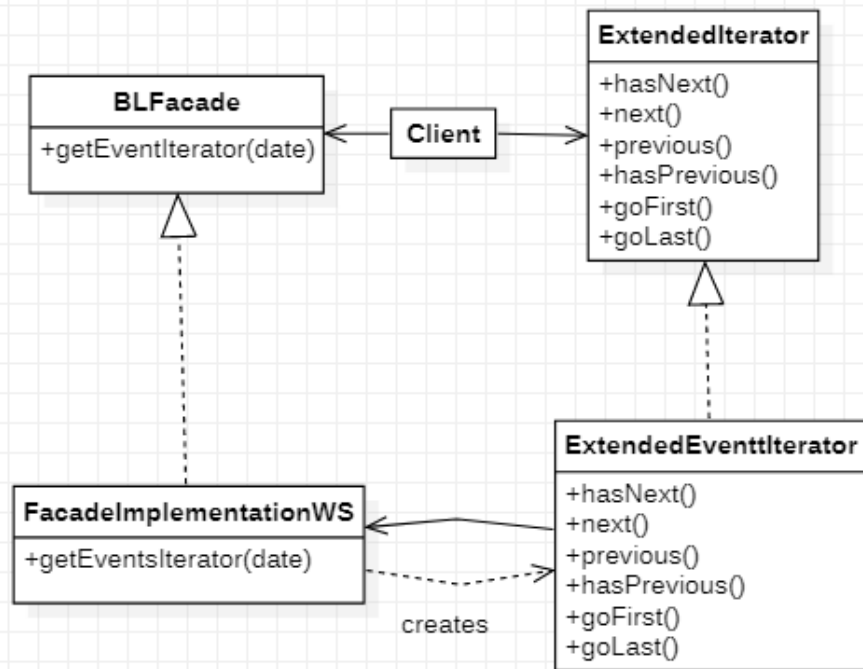


# Iterator diseinu partroia bets21Aplikazioan

## UML DIAGRAMA



## ALDATUTAKO KODEA

ExtendedIterator interfacia sortu dugu, iterator eta iterable zabalitzen dituen. Ondoren ExtendedEventIterator klasea sortu dugu ExtendedIterator zabalitzen duena. Iterator interfaceko hasNext eta next metodoak gainidazteaz gain, ExtendedIteratorean adierazitako hasPrevious, previous, goFirst eta goLast metodoak implementatu ditugu. Horrek ongi funtzionatzen duela ziurtatzeko IteratorProba klasea sortu eta bertan dokumentuan emandako probak burutu ditugu.

## Interfacea:

```
package domain;

import java.util.Collection;
import java.util.Iterator;

public interface ExtendedIterator<Event> extends Iterator<Event>, Iterable{
    //uneko elementua itzultzen du eta aurrekora pasatzen da
    public Event previous();
    //true aurreko elementua existitzen bada.
    public boolean hasPrevious();
    //lehendabiziko elementuan kokatzen da.
    public void goFirst();
    //Azkeneko elementuan kokatzen da.
    public void goLast();

    void add(Event ev);

    public boolean isEmpty();

    public int size();
}
```

## Implementazioa:

```
package domain;

import java.util.Iterator;
import java.util.Vector;

public class ExtendedEventIterator implements ExtendedIterator<Event> {
    private int pos;
    private Vector<Event> array;

    public ExtendedEventIterator() {
        this.pos = 0;
        this.array = new Vector<Event>();
    }

    public ExtendedEventIterator(Iterable<Event> array) {
        this.pos = 0;
        this.array = new Vector<Event>();
        for(Event e : array) {
            this.array.add(e);
        }
    }

    @Override
    public boolean hasNext() {
        return (pos >= 0 && pos < array.size());

        /*
        if(pos > array.size()-1) {
            return true;
        } else {
            return false;
        }*/
    }
}
```

```

@Override
public Event next() {
    Event e = array.get(pos);
    pos++;
    return e;
}

@Override
public boolean hasPrevious() {
    return (pos >= 0 && pos < array.size());
}

@Override
public Event previous() {
    Event e = array.get(pos);
    pos--;
    return e;
}

@Override
public void goFirst() {
    pos = 0;
}

@Override
public void goLast() {
    pos = array.size()-1;
}

@Override
public void add(Event ev) {
    array.add(ev);
}

```

```

@Override
public Iterator iterator() {
    return this;
}

@Override
public boolean isEmpty() {
    return array.isEmpty();
}

@Override
public int size() {
    return array.size();
}
}

```

## PROBA APLIKAZIOA

```
package domain;

import java.net.MalformedURLException;

public class IteratorProba {
    public static void main(String[] args) {
        ConfigXML c=ConfigXML.getInstance();
        Locale.setDefault(new Locale(c.getLocale()));
        boolean isLocal=true;
        BLFacadeFactory blFacade = new BLFacadeFactory();
        BLFacade facade = null;
        try {
            facade = blFacade.createBLFacade(c);
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
        try {
            blFacade.createBLFacade(c);
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
        Date date = parseDate("2022-11-14");
        ExtendedIterator<Event> i=facade.getEventIterator(date);
        Event ev;
```

```
        //lista osoa errepasatzen du atzetik aurrera
        i.goLast();
        while (i.hasPrevious()){
            ev=i.previous();
            System.out.println(ev.toString());
        }

        //lista aurretik atzera errepasatu
        while (i.hasNext()){
            ev=i.next();
            System.out.println(ev.toString());
        }

        //lehen posiziara bueltatu
        i.goFirst();
    }
    public static Date parseDate(String date) {
        try {
            return new SimpleDateFormat("yyyy-MM-dd").parse(date);
        } catch (ParseException e) {
            return null;
        }
    }
}
```

Ez dugu proba aplikazioaren exekuzioa erakusten datu basea hasieratzean errorea ematen duelako. Datu basea ez da ondo ezabatzen.