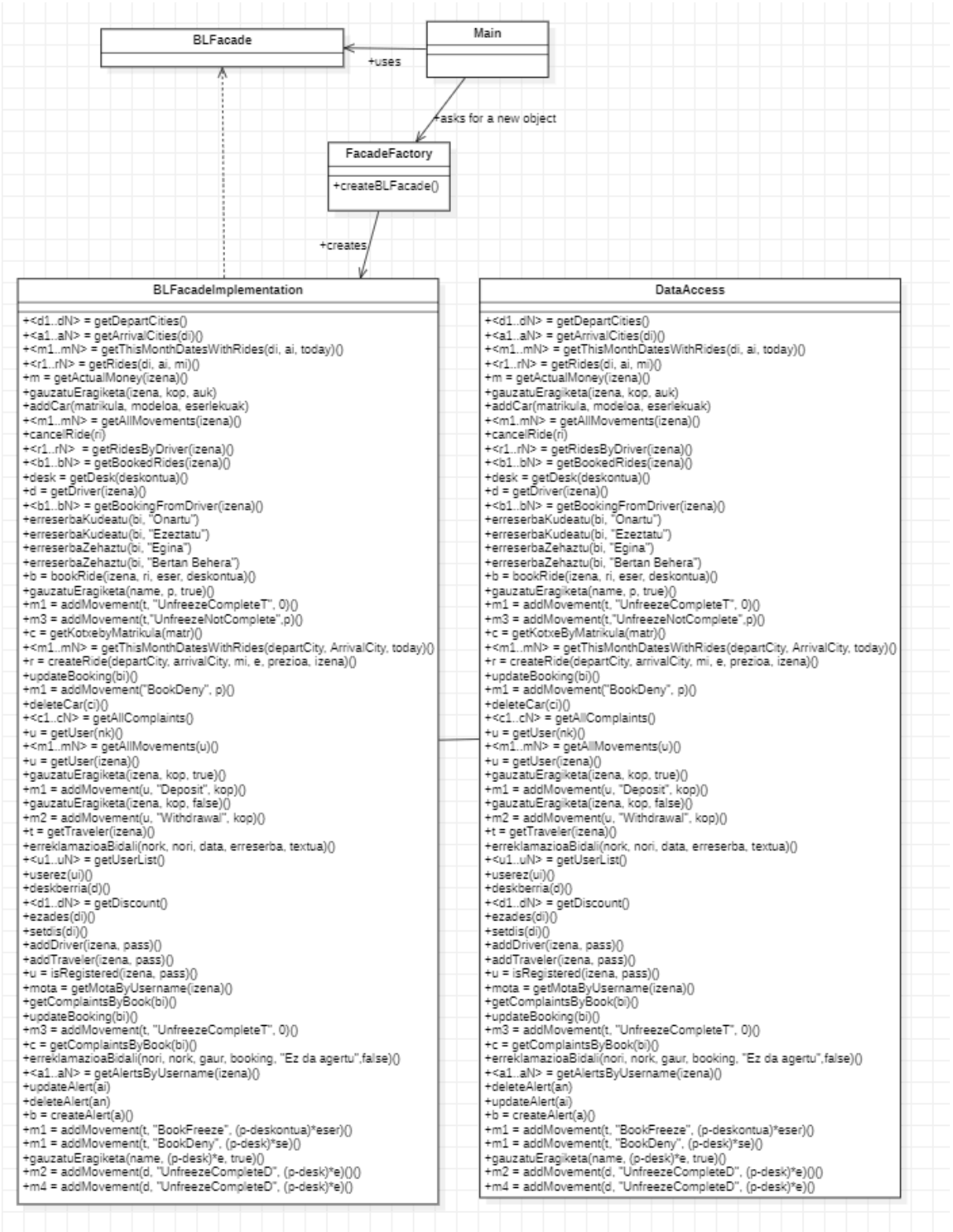
Factory Method Patroia

a) UMLa (osoa proiektuaren barnean ikusi daiteke):



b) Kodean aldaketak:
ApplicationLauncher-ean zegoen if-else kendu dut eta horren ordez FacadeFactoryrekin sortzen du BLFacade bat, behar den aukeraren arabera.

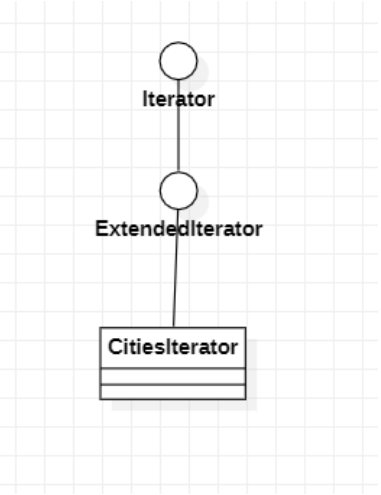
```
public class ApplicationLauncher {

    public static void main(String[] args) {

        ConfigXML c = ConfigXML.getInstance();
        Logger logger = Logger.getLogger(ApplicationLauncher.class.getName());
        logger.info(c.getLocale());

        Locale.setDefault(new Locale(c.getLocale()));

        System.out.println("Locale: " + Locale.getDefault());

        try {

            BLFacade appFacadeInterface;
            UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

            DataAccess da = new DataAccess();
            appFacadeInterface = FacadeFactory.createBLFacade(da);

            MainGUI.setBussinessLogic(appFacadeInterface);
            MainGUI a = new MainGUI();
            a.setVisible(true);

        } catch (Exception e) {
            // a.jLabelSelectOption.setText("Error: "+e.toString());
            // a.jLabelSelectOption.setForeground(Color.RED);

            System.out.println("Error in ApplicationLauncher: " + e.toString());
        }

    }

}
```

FacadeFactory klase berrian:
Lehen ApplicationLauncher-ean egiten zen if-else orain hemen egiten da, horrela beste momenturen batean beste inplementazio bat gehitu nahi badugu, ez dugu ApplicationLauncher aldatu beharko.

```
public class FacadeFactory {

    public static BLFacade createBLFacade(DataAccess da) {
        ConfigXML c = ConfigXML.getInstance();
        boolean isLocal = c.isBusinessLogicLocal();

        if (isLocal) {
            return new BLFacadeImplementation(da);
        } else {
            try {
                String serviceName = "http://" + c.getBusinessLogicNode() + ":" + c.getBusinessLogicPort() + "/ws/"
                        + c.getBusinessLogicName() + "?wsdl";
                URL url = new URL(serviceName);
                // 1st argument refers to wsdl document above
                // 2nd argument is service name, refer to wsdl document above
                QName qname = new QName("http://businessLogic/", "BLFacadeImplementationService");
                Service service = Service.create(url, qname);
                return service.getPort(BLFacade.class);
            } catch (MalformedURLException e) {
                System.out.println("Error creating remote BLFacade: " + e.getMessage());
                e.printStackTrace();
                return null;
            }
        }
    }

}
```

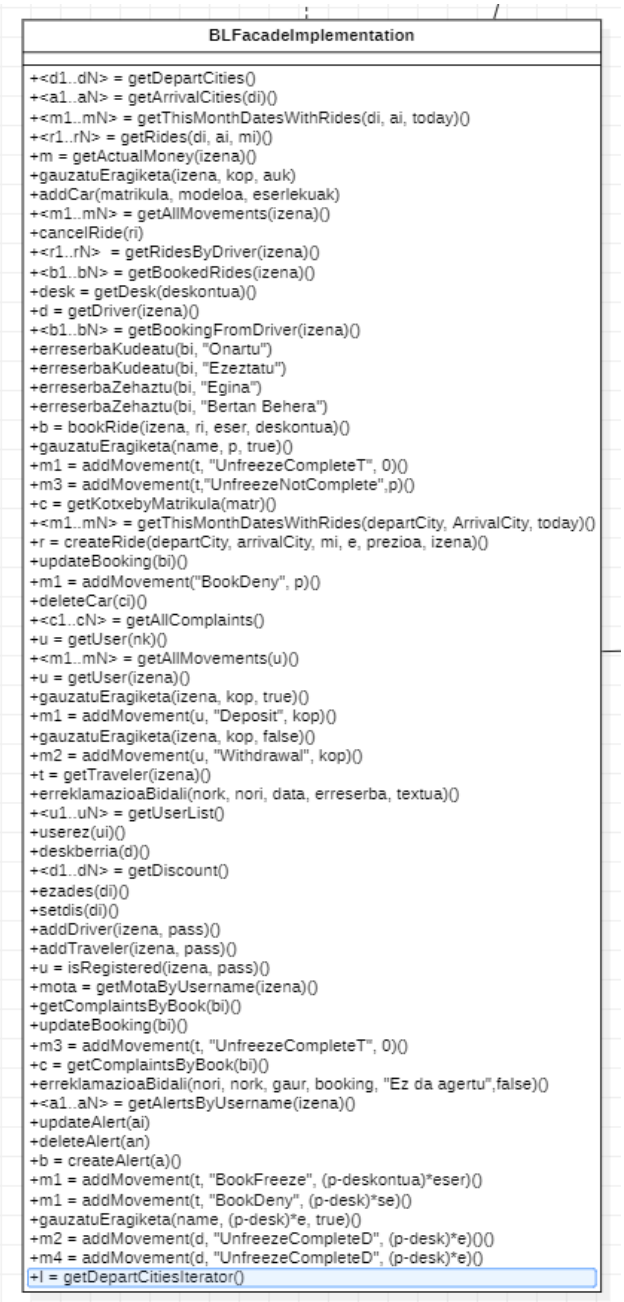
Iterator patroia

a) UML



CitiesIterator ExtendedIterator inplementatzen du, eta ExtendedIterator-ek Iterator extenditzen du.

Honez gain, BLFacadeImplementationen, metodo berria jarri dugu:



b) Kode berria:

Hasteko, BLFacaden sartu dugun metodoaren inplementazioa hurrengoa da:

```
public  ExtendedIterator<String> getDepartingCitiesIterator(){
    dbManager.open();
    List<String> departLocations = dbManager.getDepartCities();
    CitiesIterator iterator=new CitiesIterator(departLocations);

    dbManager.close();

    return iterator;
}
```

Ondoren, ExtendedIterator klasea, Iterator extenditzen duena. Klase honek iterator normal batek egiten dituen funtzioak egiten ditu eta guk inplementatu ditugun beste batzuk ere, horregatik hau interfaze moduan sortu dugu, eta klase bezala sortzen ditugun iteradoreek hau inplementatuko dute.

ExtendedIterator.java

```
1  package iterator;
2
3  import java.util.Iterator;
4
5  public interface ExtendedIterator<Object> extends Iterator<Object>{
6      //return    the actual  element and go  to  the previous
7      public Object   previous();
8      //true  if  there   is  a   previous    element
9      public boolean hasPrevious();
10     //It    is  placed  in  the first   element
11     public void goFirst();
12     //It is placed  in  the last    element
13     public void goLast();
14 }
```

Gero IteratorCities klasea sortu dugu, ExtendedIterator interfazea inplementatzen duena. Klase honek cities izeneko lista bat jasotzen du, hiriez osatuta dagoena (String-ak).

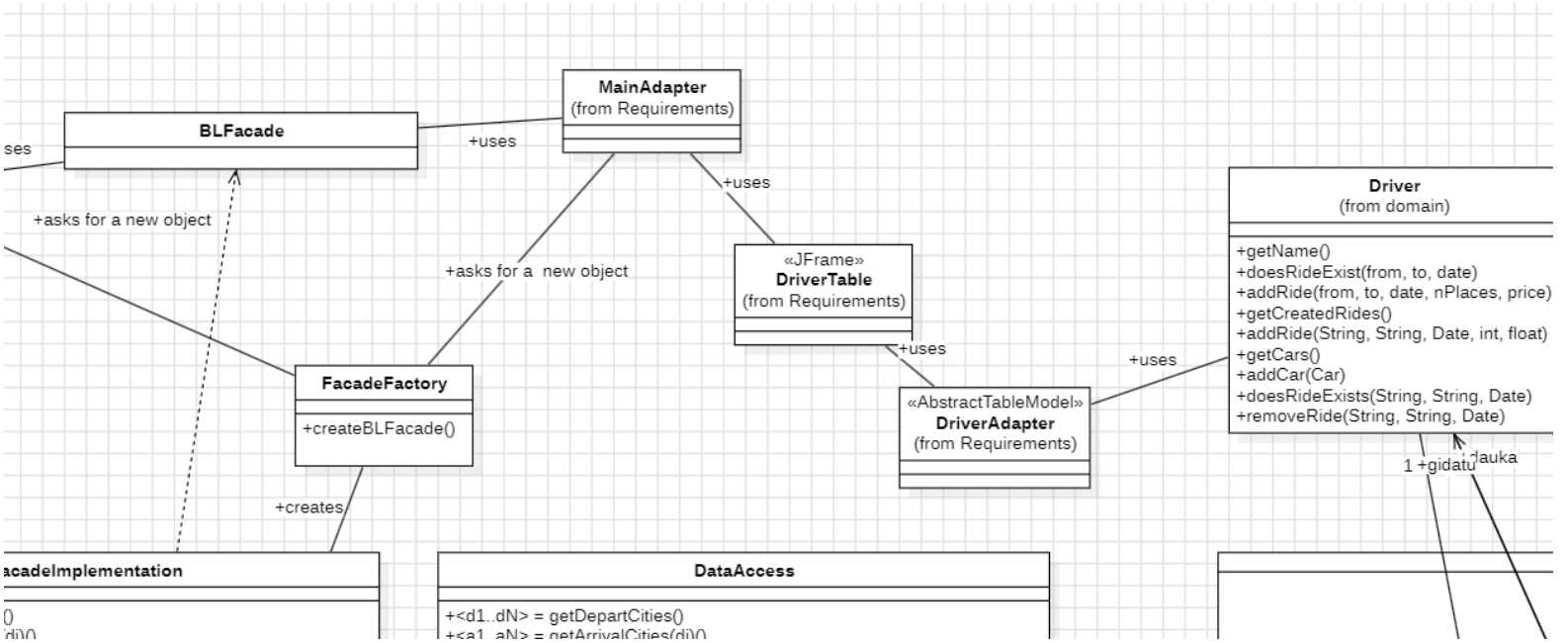
```java
import java.util.List;
public class CitiesIterator implements ExtendedIterator<String> {
    int i=0;
    List<String> cities;
    public CitiesIterator(List<String> cities) {
        this.cities=cities;
    }
    @Override
    public boolean hasNext() {
        return i < cities.size();
    }
    @Override
    public String next() {
        String c =cities.get(i);
        i = i + 1;
        return c;
    }
    @Override
    public String previous() {
        i = i - 1;
        String c =cities.get(i);
        return c;
    }
    @Override
    public boolean hasPrevious() {
        return i>=1;
    }
    @Override
    public void goFirst() {
        i=0;
    }
    @Override
    public void goLast() {
        i=cities.size();
    }
}
```

c) Main exekutatzean lortutakoa:

```
<terminated> Main [Java Application] C:\Users\Usuario\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.
Read from config.xml:    businessLogicLocal=true         databaseLocal=true
File deleted
DataAccess opened => isDatabaseLocal: true
Db initialized
DataAccess created => isDatabaseLocal: true isDatabaseInitialized: true
DataAcess closed
Creating BLFacadeImplementation instance with DataAccess parameter
DataAccess opened => isDatabaseLocal: true
DataAcess closed

_____________________
FROM    LAST    TO      FIRST
Madrid
Irun
Donostia
Barcelona


_____________________
FROM    FIRST   TO      LAST
Barcelona
Donostia
Irun
Madrid
```

Adapter patroia:

a) Hasteko MainAdapter-ek datu basea hartu eta DriverTableri deitzen dio gidaria eta datu basea pasata. Klase honek taula sortzeko, driverAdapter klaseari deitzen dio, AbstractTableModel klasea inplementatzen duela, dei hau egiterakoan gidari eta datu basea pasatzen dio, honek BLFacaderen bitartez metodo batzuk erabil behar dituelako.



b) Lehen esan bezala, DriverAdapter klasean DataAccess klase bat pasatzen da, honela, rides atributuan gidari honen bidaia guztiak gorde ditzakegulako. Atributu bezala, gidari bidaia eta BLFacadeaz gain columnNames da, honetan zutabe bakoitzean zer gordetzen ari garen jartzen da. Erabiltzen diren metodoak beharrezkoak dira taula ondo erakutsi al izateko. Hemen dago kodearen argazkia:

```java
public class DriverAdapter extends AbstractTableModel {
    protected Driver driver;
    protected String[] columnNames= new String [] {"from","to","date","places","price"};

    private BLFacade blFacade;
    //private static BLFacade appFacadeInterface;
    private List<Ride> rides;

    public DriverAdapter(Driver driver,DataAccess da) {
        this.driver=driver;
        blFacade= new FacadeFactory().createBLFacade(da);
        rides=blFacade.getRidesByDriver(driver.getUsername());

    }
    public int getColumnCount() {
        return columnNames.length ;
    }
    public int getRowCount() {
        return rides.size();
    }
    public String getColumnName(int i) {
        // Challenge!
          return columnNames[i];
      }
    public Object getValueAt(int row, int col) {
        switch (col) {
        case 0:
            return rides.get(row).getFrom();
        case 1:
            return rides.get(row).getTo();
        case 2:
            return rides.get(row).getDate();
        case 3:
            return rides.get(row).getnPlaces();
        case 4:
            return rides.get(row).getPrice();
        default:
            return "Error";
        }

    }
}
```

c) Hemen dago exekuzioaren irudia:

```java
package adapter;

import businessLogic.BLFacade;

public class AdapterMain {

    public static void main(String[] args) {
        DataAccess da = new DataAccess();
        BLFacade blFacade = new FacadeFactory().createBLFacade(da);
        Driver d= blFacade.getDriver("Urtzi");
        DriverTable dt=new DriverTable(d,da);
        dt.setVisible(true);

    }

}
```

Urtzi's rides

from	to	date	places	price
Donostia	Madrid	Thu May 30 00:00:00 ...	5	20.0
Irun	Donostia	Thu May 30 00:00:00 ...	5	2.0
Madrid	Donostia	Fri May 10 00:00:00 C...	5	5.0
Barcelona	Madrid	Sat Apr 20 00:00:00 C...	0	10.0