Bets: Patterns

Josu Aguinaga Bengoetxea[1]

E-mail: [1]aginagajosu@gmail.com

2023.eko azaroaren 12

Gaien Aurkibidea

1 Factory Method patroia **2**
- 1.1 UML . . . 2
- 1.2 Kode aldaketak . . . 3
  - 1.2.1 BLFactory.java . . . 3
  - 1.2.2 ApplicationLauncher.java . . . 4

2 Iterator patroia **5**
- 2.1 UML . . . 6
- 2.2 Kode aldaketak . . . 6
  - 2.2.1 ExtendedIterator . . . 6
  - 2.2.2 ExtendedIteratorEvents . . . 7
  - 2.2.3 BLFacade . . . 8
  - 2.2.4 BLFacadeImplementation . . . 8
  - 2.2.5 ApplicationLauncher . . . 9
- 2.3 Irudia . . . 10

3 Adapter patroia **11**
- 3.1 UML . . . 12
- 3.2 Kode aldaketak . . . 12
  - 3.2.1 Etiquetas . . . 12
  - 3.2.2 BLFacadeImplementationModelAdapter . . . 12
  - 3.2.3 SeeMovementsTableGUI . . . 14
  - 3.2.4 UserGUI . . . 15
- 3.3 Irudia . . . 16

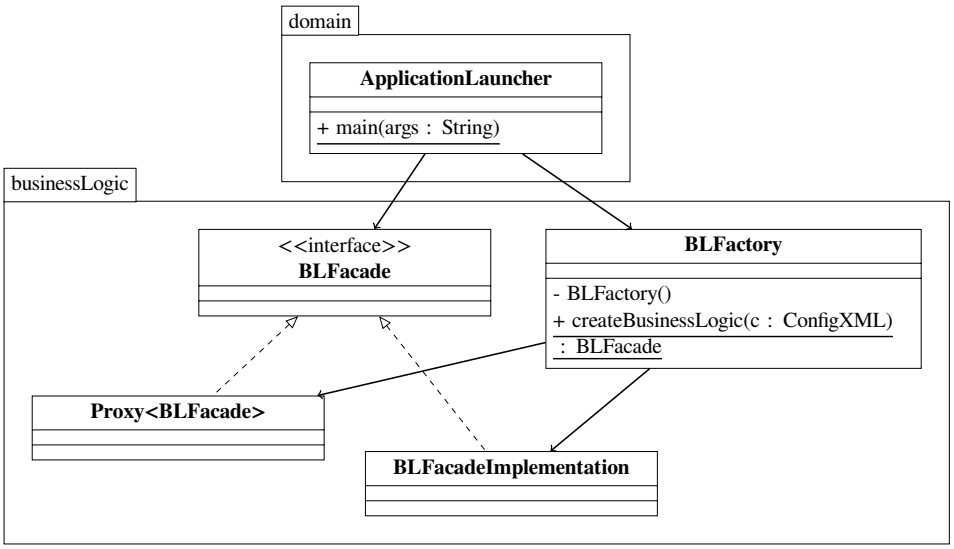
Kapitulua 1

Factory Method patroia

Kapituluaren edukia

1.1 UML 2
1.2 Kode aldaketak 3
1.2.1 BLFactory.java 3
1.2.2 ApplicationLauncher.java 4

ATALA 1 UML



Irudia 1.1: Factory Method patroia aplikatutako UML

ATALA 2 Kode aldaketak

AZPIATALA 1 BLFactory.java

```
1  package businessLogic;
2
3  import java.net.URL;
4
5  import javax.xml.namespace.QName;
6  import javax.xml.ws.Service;
7
8  import configuration.ConfigXML;
9  import dataAccess.DataAccess;
10
11 public final class BLFactory {
12
13     private BLFactory() {
14         throw new IllegalStateException("Utility
            ↪ class");
15     }
16
17     public static BLFacade
        ↪ createBusinessLogic(ConfigXML c) throws
        ↪ NullPointerException {
18         if (c.isBusinessLogicLocal()) {
19             DataAccess da= new DataAccess(c.getDataBase
                ↪ OpenMode().equals("initialize"));
20             return new BLFacadeImplementation(da);
21         }
22
23         String serviceName = "http://" +
            ↪ c.getBusinessLogicNode() + ":"
24                 + c.getBusinessLogicPort() + "/ws/" +
                    ↪ c.getBusinessLogicName()
25                 + "?wsdl";
26         try {
27             URL url = new URL(serviceName);
28             QName qname = new
                ↪ QName("http://businessLogic/",
                ↪ "BLFacadeImplementationService");
29             Service service = Service.create(url,
                ↪ qname);
30             return service.getPort(BLFacade.class);
31         }
32         catch(Exception e) {
```

```
33              e.printStackTrace();
34          }
35
36          throw new NullPointerException("Function
             ↪  returned a null value");
37      }
38
39  }
```

AZPIATALA 2 ApplicationLauncher.java

```
1   public static void main(String[] args) {
2       ConfigXML c = ConfigXML.getInstance();
3       System.out.println(c.getLocale());
4       Locale.setDefault(new Locale(c.getLocale()));
5       System.out.println("Locale: " +
         ↪  Locale.getDefault());
6       MainGUI a = new MainGUI();
7       a.setVisible(true);
8       try {
9           BLFacade appFacadeInterface;
10          UIManager.setLookAndFeel("javax.swing.plaf.meta
             ↪  l.MetalLookAndFeel");
11          appFacadeInterface =
             ↪  BLFactory.createBusinessLogic(c);
12          MainGUI.setBussinessLogic(appFacadeInterface);
13      }
14      catch (Exception e) {
15          a.jLabelSelectOption.setText("Error: " +
             ↪  e.toString());
16          a.jLabelSelectOption.setForeground(Color.RED);
17          System.out.println("Error in
             ↪  ApplicationLauncher: " + e.toString());
18      }
19  }
```

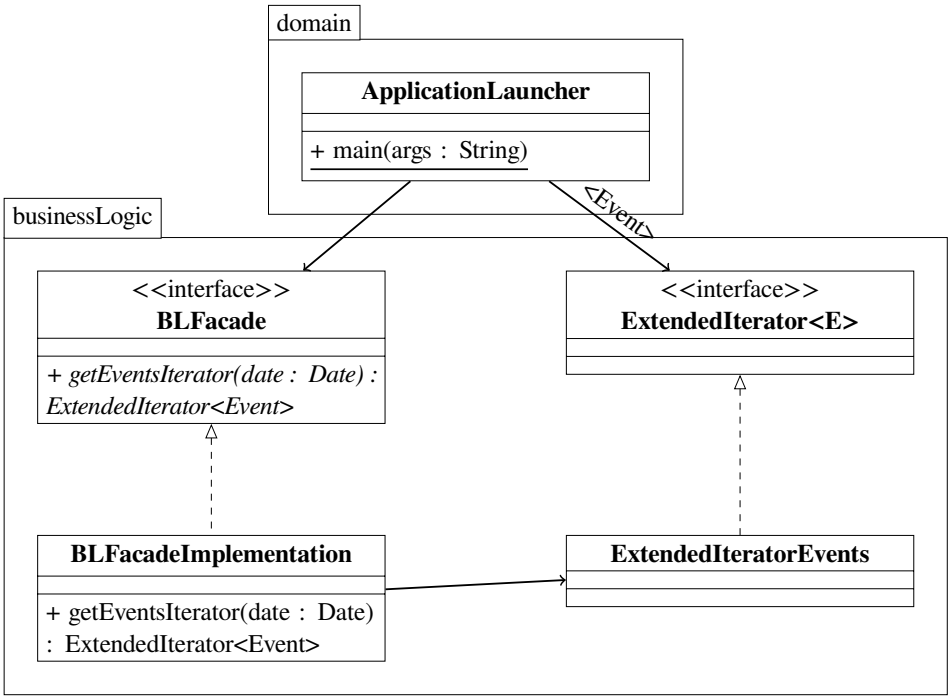
Kapitulua 2

Iterator patroia

Kapituluaren edukia

2.1 UML 6
2.2 Kode aldaketak 6
2.2.1 ExtendedIterator 6
2.2.2 ExtendedIteratorEvents 7
2.2.3 BLFacade 8
2.2.4 BLFacadeImplementation 8
2.2.5 ApplicationLauncher 9
2.3 Irudia 10

ATALA 1 UML



Irudia 2.1: Iterator patroia aplikatutako UML

ATALA 2 Kode aldaketak

AZPIATALA 1 ExtendedIterator

```
1  package businessLogic;
2
3  import java.util.Iterator;
4
5  public interface ExtendedIterator<E> extends
    ↪  Iterator<E> {
6      //uneko elementua itzultzen du eta aurrekora
        ↪  pasatzen da
7      public E previous();
8      //true aurreko elementua existitzen bada.
9      public boolean hasPrevious();
10     //Lehendabiziko elementuan kokatzen da.
```

```
11      public void goFirst();
12      //Azkeneko elementuan kokatzen da.
13      public void goLast();
14  }
```

AZPIATALA 2 **ExtendedIteratorEvents**

```
1   package businessLogic;
2
3   import java.util.NoSuchElementException;
4
5   import domain.Event;
6
7   public class ExtendedIteratorEvents implements
     ↪  ExtendedIterator<Event> {
8
9       private Event[] events;
10      private int current;
11
12      public ExtendedIteratorEvents(Event[] events) {
13          this.events = events;
14          this.current = 0;
15      }
16
17      @Override
18      public boolean hasNext() {
19          return this.events.length != 0 && this.current
             ↪  < this.events.length;
20      }
21
22      @Override
23      public Event next() {
24          if (!this.hasNext()) {
25              throw new NoSuchElementException();
26          }
27          return this.events[this.current++];
28      }
29
30      @Override
31      public Event previous() {
32          if (!this.hasPrevious()) {
33              throw new NoSuchElementException();
34          }
```

```
35          return this.events[this.current--];
36      }
37
38      @Override
39      public boolean hasPrevious() {
40          return this.events.length != 0 && this.current
             ↪  > -1;
41      }
42
43      @Override
44      public void goFirst() {
45          this.current = 0;
46      }
47
48      @Override
49      public void goLast() {
50          this.current = this.events.length > 0 ?
             ↪  this.events.length - 1 : 0;
51      }
52
53  }
```

AZPIATALA 3 **BLFacade**

```
1   ...
2   @WebMethod public ExtendedIterator<Event>
     ↪  getEventsIterator(Date date);
3   ...
```

AZPIATALA 4 **BLFacadeImplementation**

```
1   ...
2   @WebMethod
3   public ExtendedIterator<Event> getEventsIterator(Date
     ↪  date) {
4       dbManager.open(false);
5       Vector<Event> events = dbManager.getEvents(date);
6       dbManager.close();
```

```
7       return new
         ↪  ExtendedIteratorEvents(events.toArray(new
         ↪  Event[0]));
8   }
9   ...
```

AZPIATALA 5 ApplicationLauncher

```
1   ...
2   SimpleDateFormat sdf = new
     ↪  SimpleDateFormat("dd/MM/yyyy");
3   Date date = sdf.parse("17/06/2023");
4   ExtendedIterator<Event> i =
     ↪  appFacadeInterface.getEventsIterator(date);
5
6   Event e;
7
8   System.out.println("_________________________");
9   System.out.println("ATZETIK AURRERAKA");
10
11  i.goLast();
12  while (i.hasPrevious()) {
13      e = i.previous();
14      System.out.println(e.toString());
15  }
16
17  System.out.println("\n_________________________");
18  System.out.println("AURRETIK ATZERA");
19
20  i.goFirst();
21  while (i.hasNext()) {
22      e = i.next();
23      System.out.println(e.toString());
24  }
25  ...
```

ATALA 3 **Irudia**

Irudia 2.2: ApplicationLauncher klasearen exekuzioa aldaketak ondoren

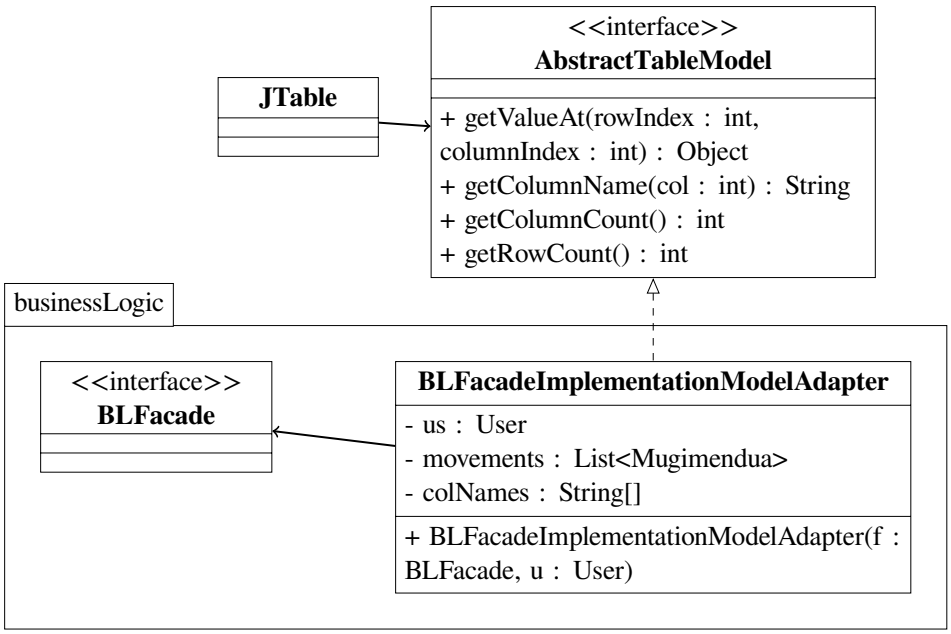
Kapitulua 3

Adapter patroia

Kapituluaren edukia

3.1 UML 12
3.2 Kode aldaketak 12
3.2.1 Etiquetas 12
3.2.2 BLFacadeImplementationModelAdapter 12
3.2.3 SeeMovementsTableGUI 14
3.2.4 UserGUI 15
3.3 Irudia 16

ATALA 1 UML



Irudia 3.1: Adapter patroia aplikatutako UML

ATALA 2 Kode aldaketak

AZPIATALA 1 Etiquetas

```
...
MovementTableOf=Table of
...
```

AZPIATALA 2 BLFacadeImplementationModelAdapter

```
package businessLogic;

import domain.Event;
```

```
4  import domain.Mugimendua;
5  import domain.Question;
6  import domain.User;
7
8  import java.util.ArrayList;
9  import java.util.List;
10
11 import javax.swing.table.AbstractTableModel;
12
13 public class BLFacadeImplementationModelAdapter extends
    ↪ AbstractTableModel {
14     private User us;
15     private final List<Mugimendua> movements;
16     private String[] colNames = new String[] {"Event",
        ↪ "Question", "Event date", "Bet (€)"};
17
18     public BLFacadeImplementationModelAdapter(BLFacade
        ↪ f, User u) {
19         this.us = f.getUser(u);
20         this.movements = us.getMugimenduak();
21     }
22
23     @Override
24     public Object getValueAt(int rowIndex, int
        ↪ columnIndex) {
25         Event ev;
26         Question q;
27         Mugimendua move = movements.get(rowIndex);
28         if (move == null) {
29             return "";
30         }
31         switch(columnIndex) {
32             case 0:
33                 ev = move.getGertaera();
34                 return ev != null ? ev.getDescription()
                    ↪ : "";
35             case 1:
36                 q = move.getGaldera();
37                 return q != null ? q.getQuestion() :
                    ↪ "";
38             case 2:
39                 ev = move.getGertaera();
40                 return ev != null ? ev.getEventDate() :
                    ↪ "";
41             case 3:
42                 ev = move.getGertaera();
43                 if (ev != null) {
44                     String desc = ev.getDescription();
45                     if (desc != null &&
                        ↪ !desc.equals("")) {
```

```
46                              return move.getDiruKop();
47                          }
48                      }
49                      return "";
50                  default:
51                      return null;
52              }
53          }
54
55          @Override
56          public String getColumnName(int col) {
57              return colNames[col];
58          }
59
60          @Override
61          public int getColumnCount() {
62              return 4;
63          }
64
65          @Override
66          public int getRowCount() {
67              return this.movements.size();
68          }
69      }
```

AZPIATALA 3 SeeMovementsTableGUI

```
1   package gui;
2
3   import javax.swing.JFrame;
4   import javax.swing.JPanel;
5   import javax.swing.JScrollPane;
6   import javax.swing.JTable;
7   import javax.swing.border.EmptyBorder;
8   import javax.swing.table.AbstractTableModel;
9
10  import businessLogic.BLFacade;
11  import
     ↪ businessLogic.BLFacadeImplementationModelAdapter;
12  import domain.User;
13  import java.awt.BorderLayout;
14  import javax.swing.JTextField;
15
```

```
16 public class SeeMovementsTableGUI extends JFrame {
17     private JTable table;
18
19     public SeeMovementsTableGUI(User u) {
20         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         setTitle("Mugimenduak");
22
23         JScrollPane scrollPane = new JScrollPane();
24         getContentPane().add(scrollPane,
            ↪  BorderLayout.CENTER);
25
26         BLFacade facade = MainGUI.getBusinessLogic();
27         AbstractTableModel model = new
            ↪  BLFacadeImplementationModelAdapter(facade,
            ↪  u);
28         table = new JTable();
29         scrollPane.setViewportView(table);
30         table.setModel(model);
31     }
32 }
```

AZPIATALA 4 **UserGUI**

```
1  private JButton jButtonMugTable=null;
2
3  private JPanel getJContentPane() {
4      if (jContentPane == null) {
5          ...
6          jContentPane.add(getBotonMovTable());
7          ...
8      }
9      return jContentPane;
10 }
11
12 private JButton getBotonMovTable() {
13     if (jButtonMugTable == null) {
14         User u = user;
15         jButtonMugTable = new JButton();
16         jButtonMugTable.setText(ResourceBundle.getBundl
            ↪  e("Etiquetas").getString("MovementTableOf")
            ↪  + u.getIzena() + u.getAbizena());
17         jButtonMugTable.addActionListener(new
            ↪  java.awt.event.ActionListener() {
```

```
18              public void actionPerformed(java.awt.event.
                ↪  ActionEvent e)
                ↪  {
19                  JFrame a = new SeeMovementsTableGUI(u);
20                  a.setVisible(true);
21              }
22          });
23      }
24      return jButtonMugTable;
25  }
```

ATALA 3 Irudia

Event	Question	Event date	Bet (€)
Atlético-Athletic		Sat Jun 17 00:00:00 CEST 2023	100.0
Getafe-Celta		Thu Jun 01 00:00:00 CEST 2023	1.215752192E9
Eibar-Barcelona		Thu Jun 01 00:00:00 CEST 2023	1.215752142E9
Eibar-Barcelona		Thu Jun 01 00:00:00 CEST 2023	1.215752092E9
Alavés-Deportivo		Thu Jun 01 00:00:00 CEST 2023	1.215752042E9
Getafe-Celta		Sat Jun 17 00:00:00 CEST 2023	1.215752642E9
Eibar-Barcelona		Sun Jun 04 00:00:00 CEST 2023	1.215752592E9
c-d		Wed May 31 00:00:00 CEST 2023	1.215752802E9
Alavés-Deportivo		Sat Jun 17 00:00:00 CEST 2023	1.215752752E9
Español-Villareal		Sat Jun 17 00:00:00 CEST 2023	1.215752652E9
Malaga-Valencia		Sat Jun 17 00:00:00 CEST 2023	1.215742208E9
Girona-Leganés		Sat Jun 17 00:00:00 CEST 2023	1.215742199E9
a-b		Tue May 30 00:00:00 CEST 2023	1.215741533E9
c-d		Tue May 30 00:00:00 CEST 2023	1.2157415E9

Irudia 3.2: SeeMovementsTableGUI gui-ren exekuzioa