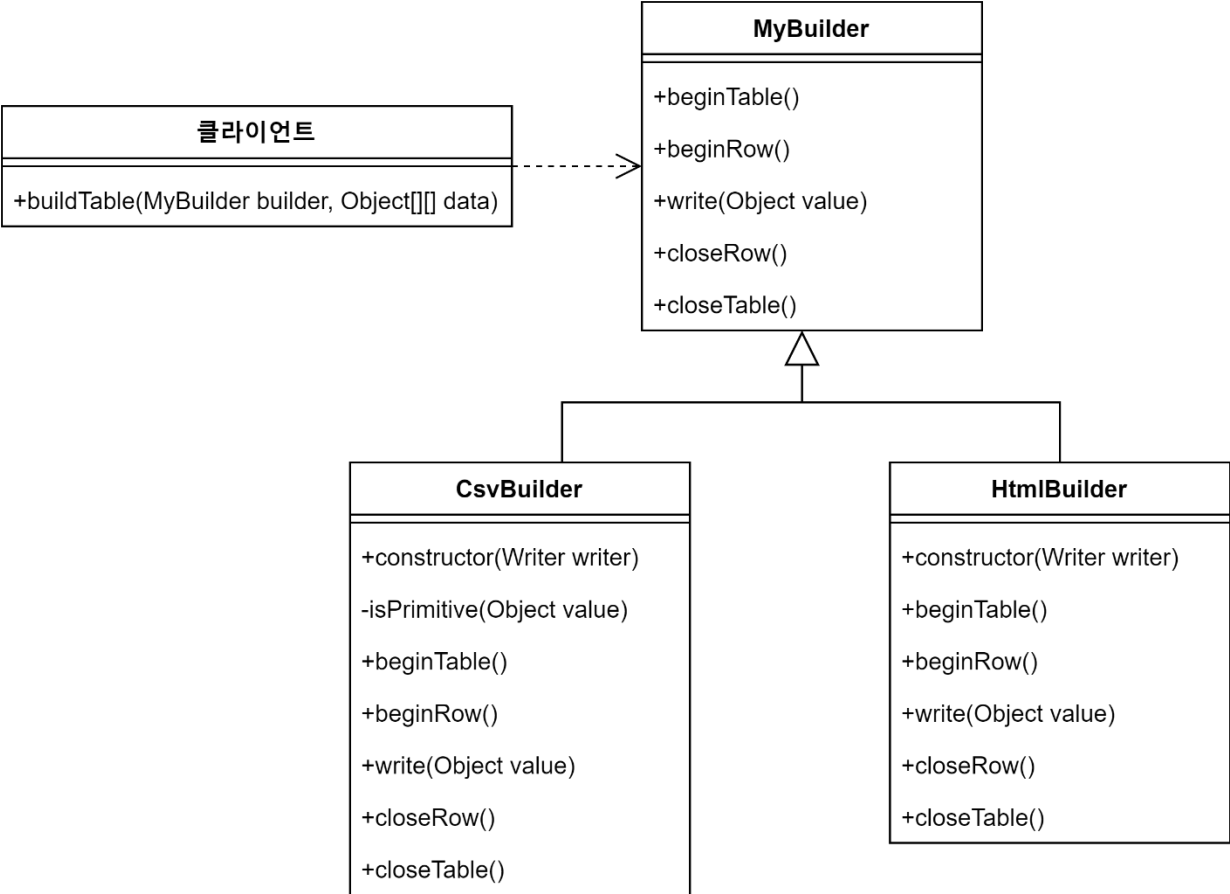


1) 개요



## 2) MyBuilder.java

```
1 package builder.e4a;
2
3 import java.io.IOException;
4
5 public interface MyBuilder {
6     void beginTable() throws IOException;
7     void beginRow() throws IOException;
8     void write(Object value) throws IOException;
9     void closeRow() throws IOException;
10    void closeTable() throws IOException;
11 }
```

### 3) CsvBuilder.java

```
1 package builder.e4a;
2
3 import java.io.IOException;
4 import java.io.Writer;
5 import java.util.HashSet;
6 import java.util.Set;
7
8 public class CsvBuilder implements MyBuilder {
9
10     private static Set<Class<?>> primitiveTypes = new HashSet<Class<?>>();
11     private boolean newLine = true;
12     private Writer writer;
13
14     public CsvBuilder(Writer writer) {
15         this.writer = writer;
16     }
17
18     static {
19         primitiveTypes.add(Boolean.class);
20         primitiveTypes.add(Character.class);
21         primitiveTypes.add(Byte.class);
22         primitiveTypes.add(Short.class);
23         primitiveTypes.add(Integer.class);
24         primitiveTypes.add(Long.class);
25         primitiveTypes.add(Float.class);
26         primitiveTypes.add(Double.class);
27     }
28
29     private static boolean isPrimitive(Object value) {
30         return primitiveTypes.contains(value.getClass());
31     }
32
33     @Override
34     public void write(Object value) throws IOException {
35         if (newLine)
36             newLine = false;
37         else
38             writer.write(",");
39
40         if (isPrimitive(value))
41             writer.write(value.toString());
42         else {
43             String s = value.toString();
44             s = s.replaceAll("([0-9]{6})-[0-9]{7}", "$1-*****");
45
46             writer.write("W");
47             writer.write(s.replace("W", "WWW"));
48             writer.write("W");
49         }
50     }
51
52     @Override
53     public void closeRow() throws IOException {
54         writer.write("\n");
55         newLine = true;
56     }
57
58     @Override
59     public void beginTable() throws IOException {
60     }
61
62     @Override
63     public void beginRow() throws IOException {
64     }
65
66     @Override
67     public void closeTable() throws IOException {
68     }
```

69 70	}
----------	---

## 4) HtmlBuilder.java

```
1 package builder.e4a;
2
3 import java.io.IOException;
4 import java.io.Writer;
5
6 public class HtmlBuilder implements MyBuilder {
7
8     private Writer writer;
9
10    public HtmlBuilder(Writer writer) {
11        this.writer = writer;
12    }
13
14    @Override
15    public void beginTable() throws IOException {
16        writer.write("<table>\\n");
17    }
18
19    @Override
20    public void beginRow() throws IOException {
21        writer.write("<tr>");
22    }
23
24    @Override
25    public void write(Object value) throws IOException {
26        String s = value.toString();
27        s = s.replaceAll("([0-9]{6})-([0-9]{7})", "$1-*****");
28        writer.write("<td>");
29        writer.write(s);
30        writer.write("</td>");
31    }
32
33    @Override
34    public void closeRow() throws IOException {
35        writer.write("</tr>\\n");
36    }
37
38    @Override
39    public void closeTable() throws IOException {
40        writer.write("</table>\\n");
41    }
42 }
```

## 5) Example4a.java

```
1 package builder.e4a;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5 import java.io.Writer;
6
7 public class Example4a {
8
9     static void buildTable(MyBuilder builder, Object[][] data) throws IOException {
10         builder.beginTable();
11         for (Object[] record : data) {
12             builder.beginRow();
13             for (Object field : record)
14                 builder.write(field);
15             builder.closeRow();
16         }
17         builder.closeTable();
18     }
19
20     public static void main(String[] args) throws IOException {
21         Object[][] data = {
22             { "tom", 24, "910614-1488111", "02-333-4444" },
23             { "john", 28, "870320-1434123", "02-444-5555" },
24             { "smith", 25, "901105-1300234", "02-555-6666" }
25         };
26         Writer writer = new PrintWriter(System.out, true);
27         buildTable(new CsvBuilder(writer), data);
28         buildTable(new HtmlBuilder(writer), data);
29         writer.close();
30     }
31 }
32 }
```

출력

```
"tom",24,"910614-*****","02-333-4444"
"john",28,"870320-*****","02-444-5555"
"smith",25,"901105-*****","02-555-6666"
<table>
<tr><td>tom</td><td>24</td><td>910614-*****</td><td>02-333-4444</td></tr>
<tr><td>john</td><td>28</td><td>870320-*****</td><td>02-444-5555</td></tr>
<tr><td>smith</td><td>25</td><td>901105-*****</td><td>02-555-6666</td></tr>
</table>
```