



```

1 package Bridge_pattern_2;
2
3 public interface CodeLanguage {
4     public void useLanguage(String fileName);
5 }
6
1 package Bridge_pattern_2;
2
3 public class JavaLanguage implements CodeLanguage {
4
5     @Override
6     public void useLanguage(String fileName) {
7         // TODO 自动生成的方法存根
8         System.out.println("日志"+fileName+"使用Java语言");
9     }
10
11 }
  
```

```

1 package Bridge_pattern_2;
2
3 public class PythonLanguage implements CodeLanguage {
4
5     @Override
6     public void useLanguage(String fileName) {
7         // TODO 自动生成的方法存根
8         System.out.println("日志"+fileName+"使用Python语言");
9     }
10
11 }
  
```

```
1 package Bridge_pattern_2;
2
3 public interface OutputFileForm {
4     public void outputForm(String fileName);
5 }
6
```

```
1 package Bridge_pattern_2;
2
3 public class TextForm implements OutputFileForm {
4
5     @Override
6     public void outputForm(String fileName) {
7         // TODO 自动生成的方法存根
8         System.out.println("设置日志"+fileName+"的输出格式为.txt");
9     }
10 }
11
12
```

```
1 package Bridge_pattern_2;
2
3 public class JSONForm implements OutputFileForm {
4
5     @Override
6     public void outputForm(String fileName) {
7         // TODO 自动生成的方法存根
8         System.out.println("设置日志"+fileName+"的输出格式为.json");
9     }
10 }
11
12
```

```
1 package Bridge_pattern_2;
2
3 public abstract class Logger {
4     OutputFileForm outputForm;
5     public void setOutputForm(OutputFileForm outputForm) {
6         this.outputForm = outputForm;
7     }
8     public void setLanguage(CodeLanguage language) {
9         this.language = language;
10    }
11    CodeLanguage language;
12
13    public Logger(OutputFileForm o, CodeLanguage l) {
14        this.outputForm = o;
15        this.language = l;
16    }
17    public abstract void outputLog(String fileName);
18
19 }
```

```

1 package Bridge_pattern_2;
2
3 public class LoggerLinux extends Logger {
4
5     public LoggerLinux(OutputFileForm o, CodeLanguage l) {
6         super(o, l);
7         // TODO 自动生成的构造函数存根
8     }
9
10    @Override
11    public void outputLog(String fileName) {
12        // TODO 自动生成的方法存根
13        this.language.useLanguage(fileName);
14        this.outputForm.outputForm(fileName);
15        System.out.println("在Window平台上生成了日志"+fileName);
16    }
17 }
18 }
19

```

```

1 package Bridge_pattern_2;
2
3 public class LoggerWindows extends Logger {
4
5
6     public LoggerWindows(OutputFileForm o, CodeLanguage l){
7         super(o,l);
8     }
9     @Override
10    public void outputLog(String fileName) {
11        // TODO 自动生成的方法存根
12        this.language.useLanguage(fileName);
13        this.outputForm.outputForm(fileName);
14        System.out.println("在Window平台上生成了日志"+fileName);
15    }
16
17 }

```

```

1 package Bridge_pattern_2;
2
3 public class test {
4
5     public static void main(String[] args) {
6         // TODO 自动生成的方法存根
7         //在windows Java Text
8         JavaLanguage jl = new JavaLanguage();
9         TextForm tf = new TextForm();
10        LoggerWindows wlogger = new LoggerWindows(tf,jl);
11        wlogger.outputLog("日志1");
12    }
13 }
14
15 }
16

```

日志日志1使用Java语言
 设置日志日志1的输出格式为.txt
 在Window平台上生成了日志日志1