

AUFGABE 14

```
import java.util.ResourceBundle;

public class Persistenz {

    public static void main(String[] args) throws IOException {

        ResourceBundle rb = ResourceBundle.getBundle("PersResource");

        Screen sc = new Screen();
        ...
        sc.println( rb.getString("inhalt") + pname);
        ...
        sc.println("'+(zeile+1)+ rb.getString("zeilen") + pname
                    + rb.getString("enthalten") + "\n");
        ...
        try {
            sc.println( rb.getString("beenden") );
            do {
                sc.print( rb.getString("eingabe") );
                ...
                sc.println( rb.getString("gegeben")+(zeile+1)+": " +ea);
                ...
            } while (true);
            sc.println();
            sc.println("'+ein+ rb.getString("gelesen"));
            sc.println("'+aus+ rb.getString("geschrieben"));
        }

    }
}
```

```
import java.util.ResourceBundle;
```

```
public class PersObjekt {
```

```
    ...
```

```
    private Screen sc = new Screen();
```

```
    private ResourceBundle rb = ResourceBundle.getBundle("PersResource");
```

```
    ...
```

```
    protected void store() throws IOException {
```

```
        raus.writeObject(data);
```

```
        raus.close();
```

```
        sc.println(fileName + rb.getString("gesichert"));
```

```
    }
```

```
    ...
```

```
}
```

```
}
```

```
import java.util.ListResourceBundle;
```

```
public class PersResource extends ListResourceBundle {
```

```
    public Object[][] getContents() { return texte; }
```

```
    static final Object[][] texte = {
```

```
        { "inhalt",      "current content of" },
```

```
        { "zeilen",      "lines contained in" },
```

```
        { "enthalten",    "" },
```

```
        { "beenden",      "terminate with 'ende'" },
```

```
        { "eingabe",      "please enter data: " },
```

```
        { "gegeben",      "entered: " },
```

```
        { "gelesen",      "line(s) read from key-board" },
```

```
        { "geschrieben", "line(s) written to screen" },
```

```
        { "gesichert",    "file written" },
```

```
        { "xxx",          "xxx" }
```

```
    };
```

```
}
```

```

import java.util.ListResourceBundle;

public class PersResource_de extends ListResourceBundle {

    public Object[][] getContents() { return texte; }

    static final Object[][] texte = {
        { "inhalt",      "Aktueller Inhalt von" },
        { "zeilen",      "Zeilen in" },
        { "enthalten",    "enthalten" },
        { "beenden",      "Beenden mit 'ende'" },
        { "eingabe",      "Bitte um Eingabe: " },
        { "gegeben",      "Eingabe: " },
        { "gelesen",      "Zeile(n) von der Tastatur gelesen" },
        { "geschrieben",  "Zeile(n) auf den Bildschirm geschrieben" },
        { "gesichert",     "Datei geschrieben" },
        { "xxx",          "xxx" }
    };
}

```

AUFGABE 8

BufferedReader *in Verbindung mit einem* InputStreamReader

StringBuffer

Methoden: **int** length() **char** charAt(**int**)
 void setCharAt(**int**, **char**) **void** reverse()

Character

Klassenmethoden: **boolean** isUpperCase(char)
 oder **boolean** isLowerCase(char)

```
import java.io.*;

class StringReverse {

    public static void main(String[] args) throws IOException {

        BufferedReader in = new BufferedReader(
            new InputStreamReader(System.in));

        for (;;) {
            String s = in.readLine();
            if (s.length() == 0) break;

            StringBuffer sb = new StringBuffer(s);

            char anfang = sb.charAt(0);
            char ende = sb.charAt(sb.length()-1);

            if (Character.isUpperCase(anfang)) {
                sb.setCharAt(0, Character.toLowerCase(anfang));
                sb.setCharAt(sb.length()-1,
                    Character.toUpperCase(ende));
            }

            System.out.println(sb.reverse());
        }
    }
}
```