

```

1 import java.io.*;
2 public class file_ops {
3
4     public static void createFile() throws IOException
5     {
6         try {
7             Scanner crf = new Scanner(System.in);
8             retrieve_files obj = new retrieve_files();
9
10            System.out.println("Enter the file name you want to create");
11            String str1 = crf.nextLine();
12            StringBuilder ff = new StringBuilder(str1.toLowerCase());
13            ff.append(".txt");
14            String str = ff.toString();
15            StringBuilder stb = new StringBuilder("E:\\Virtual Key for Your Repositories\\src\\
16            \\files\\");
17            stb.append(str);
18            obj.updateAllFiles(str);
19            File nfile = new File(stb.toString());
20            if (nfile.createNewFile()) {
21                System.out.println("*****");
22                System.out.println("FILE CREATION :-> SUCCESSFUL");
23                System.out.println("*****");
24            }
25            System.out.println("-----");
26            System.out.println("press 1 for writing");
27            System.out.println("Else press 0 ");
28            int b = crf.nextInt();
29            System.out.println("-----");
30            if (b == 1) {
31                WriteFile(stb.toString());
32            }
33        }
34        catch (Exception e)
35        {
36            System.out.println(e);
37            System.out.println("*****");
38            System.out.println("FILE CREATION :-> FAILED");
39            System.out.println("*****");
40        }
41    }
42
43    public static void ReadFile(String filename)
44    {
45        try{
46            FileInputStream fin=new FileInputStream(filename);
47            int i=0;
48            while((i=fin.read())!=-1){
49                System.out.print((char)i);
50            }
51            fin.close();
52            System.out.println(" ");
53            System.out.println("*****");
54            System.out.println("READING :-> SUCCESSFUL");
55            System.out.println("*****");
56        }catch(Exception e){
57            System.out.println(e);
58        }
59    }
60

```

```
61         System.out.println("*****");
62         System.out.println("READING :-> FAILED");
63         System.out.println("*****");
64     }
65 }
66
67 public static void WriteFile(String filename)
68 {
69     try{
70         Scanner wf = new Scanner(System.in);
71         System.out.println("-----");
72         System.out.println("***** what you want to write *****");
73         String Str = wf.nextLine();
74         FileOutputStream fout = new FileOutputStream(filename);
75         byte b[] = Str.getBytes();
76         fout.write(b);
77         System.out.println("-----");
78         fout.close();
79         System.out.println("*****");
80         System.out.println("WRITING :-> SUCCESSFUL");
81         System.out.println("*****");
82     }
83     catch(Exception e)
84     {
85         System.out.println(e);
86         System.out.println("*****");
87         System.out.println("WRITING :-> FAILED");
88         System.out.println("*****");
89     }
90 }
91
92
93 public static void appendFile(String filename) throws IOException {
94     try
95     {
96         String app ;
97         Scanner ap = new Scanner(System.in);
98         System.out.println("----- Enter what you want to append -----");
99         System.out.println("-----");
100        app = ap.nextLine();
101        BufferedWriter b = new BufferedWriter(new FileWriter(filename , true));
102        b.write("\n");
103        b.write(app);
104        System.out.println("-----");
105        b.close();
106        System.out.println("*****");
107        System.out.println("WRITING :-> SUCCESSFUL");
108        System.out.println("*****");
109    }
110    catch (Exception e)
111    {
112        System.out.println(e);
113        System.out.println("*****");
114        System.out.println("WRITING :-> FAILED");
115        System.out.println("*****");
116    }
117 }
```

```
118     }
119
120     public static void DeleteFile(String filename) throws IOException {
121         Path path = Paths.get(filename);
122         try {
123             if (Files.deleteIfExists(path)) {
124                 System.out.println("*****");
125                 System.out.println("DELETION  :-> SUCCESSFUL");
126                 System.out.println("*****");
127             } else {
128                 System.out.println("*****");
129                 System.out.println("DELETION  :-> FAILED");
130                 System.out.println("*****");
131             }
132         }
133         catch (Exception e)
134         {
135             System.out.println(e);
136         }
137     }
138 }
139
140
141 public static void main() {
142     try {
143         Scanner sc = new Scanner(System.in);
144         int num = 1;
145         retrieve_files rf = new retrieve_files();
146         while(num < 3)
147         {
148             System.out.println("*****");
149             System.out.println("\t PRESS");
150             System.out.println("1 :-> To Create File");
151             System.out.println("2 :-> To Perform operation");
152             System.out.println("3 :-> BACK ");
153             System.out.print("Enter your choice : ");
154             num = sc.nextInt();
155             sc.nextLine();
156             System.out.println("*****");
157             switch (num)
158             {
159                 case 1 -> createFile();
160                 case 2 -> {
161                     System.out.println("*****");
162                     System.out.println("\t PRESS");
163                     System.out.println("1. To Read File");
164                     System.out.println("2. To Delete File");
165                     System.out.println("3. To Append File");
166                     System.out.println("4. BACK ");
167                     System.out.print("Enter your choice : ");
168                     int op_num = sc.nextInt();
169                     sc.nextLine();
170                     System.out.println("*****");
171
172                     if(op_num >= 4)
173                     {
174                         break;
```

```
175         }
176
177         System.out.println("Search the file on which you want to do
operation");
178         String filenameStr1 = sc.nextLine();
179         System.out.println("-----");
180         StringBuilder ff = new StringBuilder(filenameStr1.toLowerCase());
181         ff.append(".txt");
182         String filenameStr = ff.toString();
183         StringBuilder filename = new StringBuilder("E:\\Virtual Key for Your
Repositories\\src\\files\\");
184         filename.append(filenameStr.toLowerCase());
185         if(rf.search(filenameStr)) {
186
187             switch (op_num)
188             {
189                 case 1 -> ReadFile(filename.toString());
190                 case 2 -> {
191                     DeleteFile(filename.toString());
192                     rf.RemoveFileFromAllFiles(filenameStr);
193                 }
194                 case 3 -> appendFile(filename.toString());
195                 default -> System.out.println(" MOVING BACK ");
196             }
197         }
198         else{
199             System.out.println("file doesn't exist or wrong file name");
200             continue;
201         }
202     }
203     default -> System.out.println(" MOVING BACK ");
204 }
205 }
206 } catch (Exception e) {
207     System.out.println("Error: "+e);
208 }
209 }
210 }
211 }
```