

```
1 package Ejercicio05;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.File;
6 import java.io.FileInputStream;
7 import java.io.FileNotFoundException;
8 import java.io.FileOutputStream;
9 import java.io.FileReader;
10 import java.io.FileWriter;
11 import java.io.IOException;
12 import java.io.OutputStream;
13
14
15 public class U06E05 {
16
17     public static void main(String[] args) throws IOException {
18
19         //variables locales
20         //creamos variables finales que van a ser cada una de las letras del ADN
21         final int a = 0;
22         final int g = 1;
23         final int t = 2;
24         final int c = 3;
25
26         int contador = 0; //esta variable la uso para ir formando grupos de 4 letras
27         int binario = 0; //esta variable almacena el numero binario que vamos a ir cosntruyendo
28         int[] caracteres = new int[4]; //este array ira almacenando los caracteres leidos, hasta formar un bloque de 4
29         int ultimoByte = 0; //almacenamos el valor del ultimo byte cuando comprimimos.
30
31         BufferedReader br = null; //declaramos una variable BufferedReader
32         BufferedWriter bw = null; //declaramos una variable BufferedWriter
33
34         OutputStream salida = null; //declaramos una variable outputstream
35         FileInputStream st = null; //declaramos una variable fileinputstream
36         BufferedInputStream bis = null; //declaramos un bufferedInputStream
37         FileWriter salida2 = null;
38
39         String sinComprimir = "prueba.adn";
40         String comprimido = "prueba.adc";
41
42         //pendiente de pedir los ficheros de entrada y salida al usuario
43         //en principio vamos al grano y vamos a trabajar con ficheros especificos
44
45         //comprimir
46
47         try {
48             //crear un objeto BufferedReader al que se le pasa
49             //un objeto FileReader con el nombre del fichero
50             br = new BufferedReader(new FileReader(sinComprimir));
51             File ficheroBinario = new File(comprimido);
```

```

52     salida = new FileOutputStream(ficheroBinario);
53
54     int charRead = br.read();
55
56     //Ponemos el array a -1
57     for(int i = 0; i < 4; i++) {
58         caracteres[i] = -1;
59     }
60
61     while (charRead != -1) {
62
63
64         if(contador < caracteres.length) {
65             //System.out.println(charRead);
66             if(charRead == 65 || charRead == 97) caracteres[contador] = a;
67             if(charRead == 71 || charRead == 103) caracteres[contador] = g;
68             if(charRead == 84 || charRead == 116) caracteres[contador] = t;
69             if(charRead == 67 || charRead == 99) caracteres[contador] = c;
70             contador++;
71             if(contador < caracteres.length) charRead = br.read();
72
73         } else {
74             //primera letra
75             binario = caracteres[0];
76             //segunda letra
77             binario = binario << 2;
78             binario = binario + caracteres[1];
79             //tercera letra
80             binario = binario << 2;
81             binario = binario + caracteres[2];
82             //ultima letra
83             binario = binario << 2;
84             binario = binario + caracteres[3];
85             //sacamos la info
86             salida.write(binario);
87             //Ponemos el array a -1
88             for(int i = 0; i < 4; i++) {
89                 caracteres[i] = -1;
90             }
91             binario = 0;
92             contador=0;
93             charRead = br.read();
94         }
95
96         if (charRead == -1) {
97             binario = 0;
98             //primera letra
99             if(caracteres[0] != -1) binario = binario + caracteres[0];
100             else binario = binario + a; //
101
102             //segunda letra

```

```
103         if(caracteres[1] != -1) {
104             binario = binario << 2;
105             binario = binario + caracteres[1];
106         }
107         else {
108             binario = binario << 2;
109             binario = binario + a;
110         }
111         //tercera letra
112         if(caracteres[2] != -1) {
113             binario = binario << 2;
114             binario = binario + caracteres[2];
115         }
116         else {
117             binario = binario << 2;
118             binario = binario + a;
119         }
120         //ultima letra
121         if(caracteres[3] != -1) {
122             binario = binario << 2;
123             binario = binario + caracteres[3];
124         }
125         else {
126             binario = binario << 2;
127             binario = binario + a;
128         }
129         //sacamos la info
130         //System.out.println(binario);
131         salida.write(binario);
132
133         //añadimos el ultimo byte
134         if (contador==0) {
135             salida.write(4);
136             ultimoByte = 4;
137         }
138         if (contador==1) {
139             salida.write(1);
140             ultimoByte = 1;
141         }
142         if (contador==2) {
143             salida.write(2);
144             ultimoByte = 2;
145         }
146         if (contador==3) {
147             salida.write(3);
148             ultimoByte = 3;
149         }
150     }
151 }
152 //tratamiento de errores
153
```

```

154     } catch (FileNotFoundException e) {
155         System.out.println("Error: Fichero no encontrado");
156         System.out.println(e.getMessage());
157     } catch (Exception e) {
158         System.out.println("Error de lectura del fichero");
159         System.out.println(e.getMessage());
160     }
161     finally {
162         try {
163             if(br != null)
164                 br.close();
165         } catch (Exception e) {
166             System.out.println("Error al cerrar el fichero");
167             System.out.println(e.getMessage());
168         }
169     }
170
171     //descomprimir
172
173     try {
174
175         //creamos el FileInputStream
176         st = new FileInputStream(comprimido);
177         //lo encapsulamos en un bufferedInputStream
178         bis = new BufferedInputStream(st);
179
180         //creamos el fichero de salida
181         File archivo2 = new File("descomprimido.adn");
182         salida2 = new FileWriter(archivo2,true);
183
184         //miramos cuantos bytes tiene el fichero comprimido
185         int totalBytes = st.available();
186         int cuentaBytes = 0; //para contar las veces que leemos 1 byte en el fichero comprimido
187
188         //leemos el primer byte
189         int valor=bis.read();
190
191         //hemos leído un byte, incrementamos el contador
192         cuentaBytes++;
193
194         //mientras hay bytes que leer almacenamos en cada posición del array las veces que aparece
195         while(valor != -1) {
196             //vemos si hemos leído el penúltimo byte
197             if(cuentaBytes != totalBytes-1) {
198                 //pasamos el primer byte a binario
199                 String numBinario = Integer.toBinaryString(valor);
200                 //como los primeros ceros se ignoran, los ponemos nosotros
201                 while(numBinario.length()<8) {
202                     numBinario = "0" + numBinario;
203                 }
204                 //vamos leyendo los bits y transformándolos a carácter

```

```

205         for (int i = 0; i < 8; i=i+2) {
206             String trozo = numBinario.substring(i, i+2);
207             //System.out.println(trozo);
208             if(trozo.equals("00")) salida2.write("A");
209             if(trozo.equals("01")) salida2.write("G");
210             if(trozo.equals("10")) salida2.write("T");
211             if(trozo.equals("11")) salida2.write("C");
212         }
213
214         valor=bis.read();
215         cuentaBytes++;
216     } else {
217         //pasamos el byte a binario
218         String numBinario = Integer.toBinaryString(valor);
219         //como los primeros ceros se ignoran, los ponemos nosotros
220         while(numBinario.length()<8) {
221             numBinario = "0" + numBinario;
222         }
223         //vamos leyendo los bits y trasformandolos a caracter, pero solo los que nos marque el ultimo byte
224         ultimoByte = ultimoByte * 2;
225         if(ultimoByte == 8) ultimoByte = 0;
226         for (int i = 0; i < ultimoByte; i=i+2) {
227             String trozo = numBinario.substring(i, i+2);
228             //System.out.println(trozo);
229             if(trozo.equals("00")) salida2.write("A");
230             if(trozo.equals("01")) salida2.write("G");
231             if(trozo.equals("10")) salida2.write("T");
232             if(trozo.equals("11")) salida2.write("C");
233         }
234         valor=bis.read(); //leemos el ultimo byte, de control
235         valor=bis.read(); //volvemos a leer para forzar la salida
236         cuentaBytes++;
237     }
238 }
239
240 } catch (Exception e) {
241     System.out.println(e.getMessage());
242 } finally {
243     st.close();
244     bis.close();
245     salida2.close();
246 }
247
248 }
249
250 }
251

```