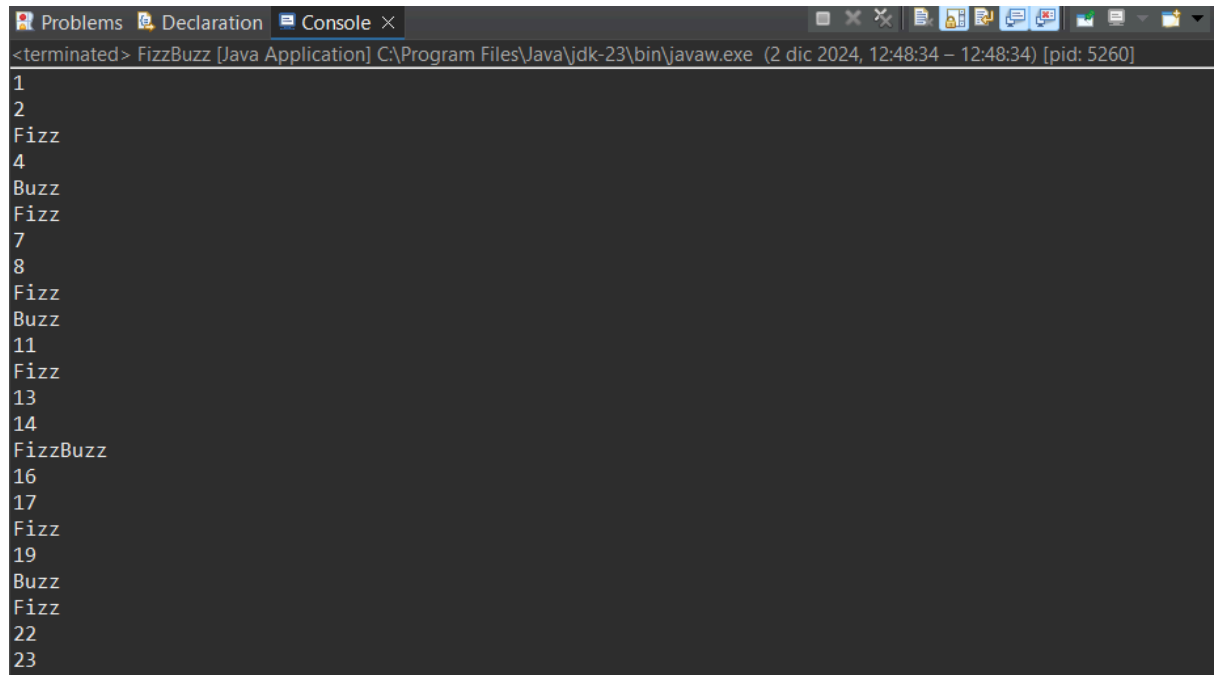


Capturas de pantalla del Código:

FizzBuzz:

```
1 package Entorno;
2
3 public class FizzBuzz {
4
5     //Clase main:
6
7     public static void main(String[] args) {
8
9         //Uso un bucle for para recorrer desde 1 hasta 100:
10
11         for (int i=1; i<=100; i++) {
12
13             //Si es múltiplo de 3 y de 5 imprime "FizzBuzz":
14
15             if (i%3==0 && i%5==0) {
16
17                 System.out.println("FizzBuzz");
18
19             //Si es múltiplo de 3 "Fizz":
20             } else if (i%3==0) {
21
22                 System.out.println("Fizz");
23
24             //Si es múltiplo de 5 imprime "Buzz":
25
26             } else if (i%5==0) {
27
28                 System.out.println("Buzz");
29
30             //Si no es múltiplo de 3 ni de 5, imprime el número:
31             } else {
32                 System.out.println(i);
33             }
34         }
35     }
36 }
37
```

Resultado en consola:



The screenshot shows an IDE's console window with the following content:

```
<terminated> FizzBuzz [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe (2 dic 2024, 12:48:34 - 12:48:34) [pid: 5260]  
1  
2  
Fizz  
4  
Buzz  
Fizz  
7  
8  
Fizz  
Buzz  
11  
Fizz  
13  
14  
FizzBuzz  
16  
17  
Fizz  
19  
Buzz  
Fizz  
22  
23
```

## Tamagotchi:

```
1 package Entorno;
2
3 public class Tamagotchi {
4
5     //Atributos (los declaro estáticos para usarlos en todo el programa, ya que me harán falta):
6
7     private static int hanger;
8     private static int energy;
9     private static int mood;
10
11     //Constructor (por defecto sus atributos serán 4):
12
13     public Tamagotchi() {
14
15         this.hanger=4;
16         this.energy=4;
17         this.mood=4;
18
19     }
20
21     //Método Play:
22
23     public String Play() {
24
25         hanger+=1;
26         mood+=1;
27         energy-=1;
28
29         //Comprueba su estado:
30
31         if(mood>8) {
32
33             return ":-)";
34
35         }
36
37         else if(energy<3) {
38
39             return"(-_-)";
40
41         }
42
43         else if(energy==0) {
```

```
44         return"(-_-) zZZ";
45     }
46
47     }
48
49     else if(mood<2) {
50
51         return"ಠ_ಠ";
52     }
53
54
55     //Si no se cumple lo anterior, su estado será "Normal":
56
57     else {
58
59         return":-|";
60
61     }
62 }
63
64 //Método Eat:
65
66 public String Eat() {
67
68     hanger-=2;
69     energy-=1;
70
71     //Comprueba su estado:
72
73     if(mood>8) {
74
75         return":-)";
76
77     }
78
79     else if(energy<3) {
80
81         return"(-_-)";
82
83     }
84
85     else if(energy==0) {
86
```

```

87         return"(-_-) zZZ";
88     }
89 }
90
91 else if(mood<2) {
92     return"ಠ_ಠ";
93 }
94
95 }
96
97 //Si no se cumple lo anterior, su estado será "Normal":
98
99 else {
100     return":-|";
101 }
102 }
103 }
104 }
105
106 //Método Sleep:
107
108 public String Sleep() {
109     energy+=2;
110     //Al dormir, pasa a estado "dormido", no hace falta ver su estado:
111     return"(-_-) zZZ";
112 }
113
114 }
115 }
116
117 //Creo una instancia de la clase para probarla:
118
119 public static void main(String[] args) {
120     Tamagotchi tamagotchi1=new Tamagotchi();
121
122     System.out.println("El tamagotchi está jugando.");
123     System.out.println("Estado del tamagotchi tras jugar: "+tamagotchi1.Play());
124     System.out.println("");
125
126     System.out.println("El tamagotchi está comiendo.");
127     System.out.println("Estado del tamagotchi tras comer: "+tamagotchi1.Eat());
128
129     System.out.println("");
130
131     System.out.println("El tamagotchi está durmiendo.");
132     System.out.println("Estado del tamagotchi tras dormir: "+tamagotchi1.Sleep());
133     System.out.println("");
134 }
135 }
136 }
137 }
138

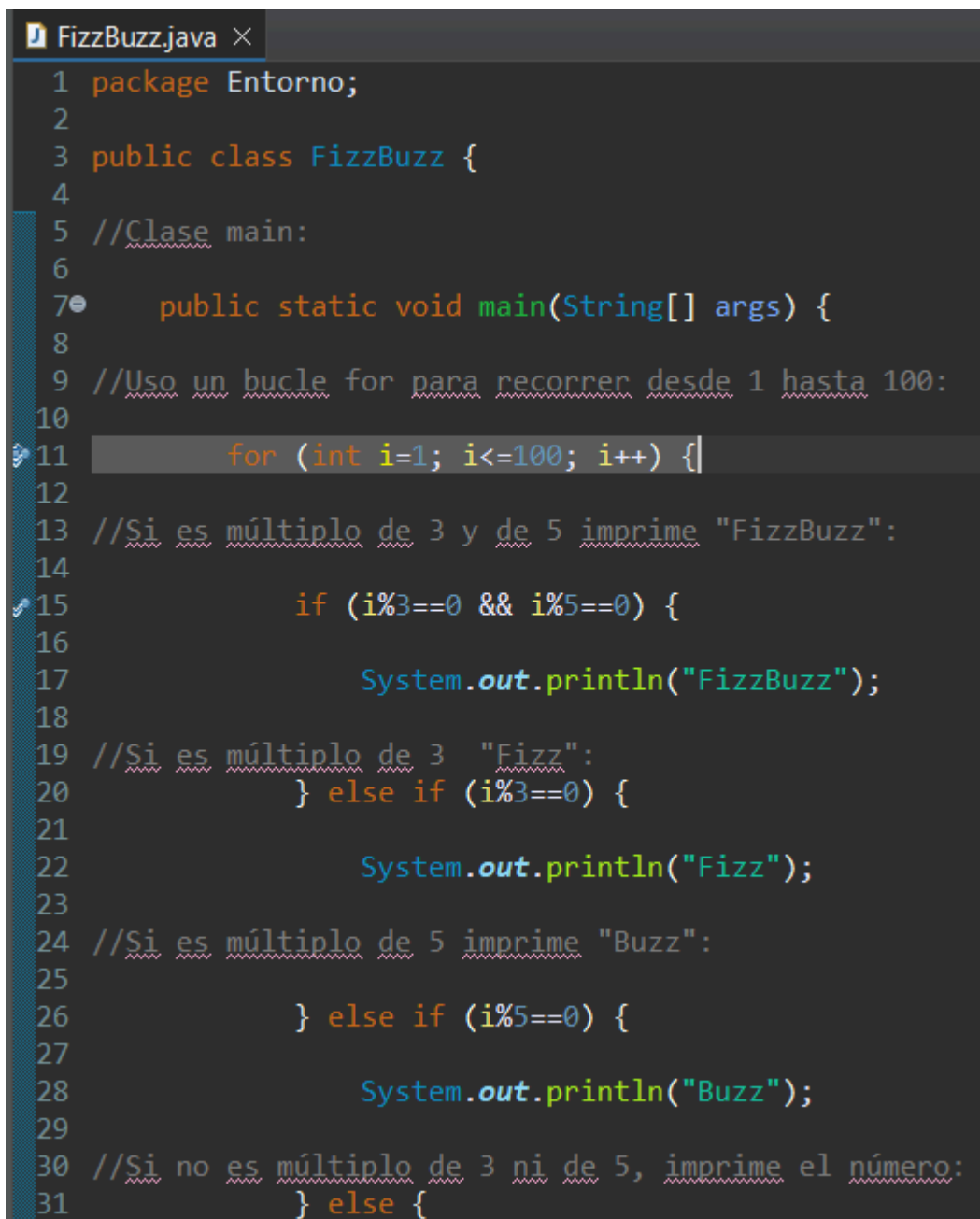
```

Resultado en consola:

```
El tamagotchi está jugando.  
Estado del tamagotchi tras jugar: :-|  
  
El tamagotchi está comiendo.  
Estado del tamagotchi tras comer: (-_-)  
  
El tamagotchi está durmiendo.  
Estado del tamagotchi tras dormir: (-_-) zZZ
```

### Capturas de pantalla del proceso de depuración

Captura de pantalla del punto de interrupción de **FizzBuzz** (línea 11):



The screenshot shows an IDE window titled "FizzBuzz.java". The code is as follows:

```
1 package Entorno;  
2  
3 public class FizzBuzz {  
4  
5     //Clase main:  
6  
7     public static void main(String[] args) {  
8  
9         //Uso un bucle for para recorrer desde 1 hasta 100:  
10  
11         for (int i=1; i<=100; i++) {  
12  
13             //Si es múltiplo de 3 y de 5 imprime "FizzBuzz":  
14  
15             if (i%3==0 && i%5==0) {  
16  
17                 System.out.println("FizzBuzz");  
18  
19             //Si es múltiplo de 3 "Fizz":  
20             } else if (i%3==0) {  
21  
22                 System.out.println("Fizz");  
23  
24             //Si es múltiplo de 5 imprime "Buzz":  
25  
26             } else if (i%5==0) {  
27  
28                 System.out.println("Buzz");  
29  
30             //Si no es múltiplo de 3 ni de 5, imprime el número:  
31             } else {
```

A blue vertical bar on the left side of the editor indicates a breakpoint is set at line 11, which is highlighted with a grey background.

Captura de pantalla del punto de interrupción de **FizzBuzz** (línea 15):

```
FizzBuzz.java ×
1 package Entorno;
2
3 public class FizzBuzz {
4
5     //Clase main:
6
7     public static void main(String[] args) {
8
9         //Uso un bucle for para recorrer desde 1 hasta 100:
10
11         for (int i=1; i<=100; i++) {
12
13             //Si es múltiplo de 3 y de 5 imprime "FizzBuzz":
14
15             if (i%3==0 && i%5==0) {
16
17                 System.out.println("FizzBuzz");
18
19             //Si es múltiplo de 3 "Fizz":
20             } else if (i%3==0) {
21
22                 System.out.println("Fizz");
23
24             //Si es múltiplo de 5 imprime "Buzz":
25
26             } else if (i%5==0) {
27
28                 System.out.println("Buzz");
29
30             //Si no es múltiplo de 3 ni de 5, imprime el número:
31             } else {
```

Estado de las variables:

(x) Variables × Breakpoints Expressions

Name	Value
no method return value	
args	String[0] (id=20)
i	1

<Choose a previously entered expression>

1

Captura de pantalla del punto de interrupción de **TamaGotchi** (línea 13):

```
1 package Entorno;
2
3 public class Tamagotchi {
4
5 //Atributos (los declaro estáticos para usarlos en todo el programa, ya que me harán falta):
6
7     private static int hanger;
8     private static int energy;
9     private static int mood;
10
11 //Constructor (por defecto sus atributos serán 4):
12
13 public Tamagotchi() {
14
15     this.hanger=4;
16     this.energy=4;
17     this.mood=4;
18
19 }
20
21 //Método Play:
22
23 public String Play() {
24
25     hanger+=1;
26     mood+=1;
27     energy-=1;
28
29 //Comprueba su estado:
30
31     if(mood>8) {
```

Captura de pantalla del punto de interrupción de **TamaGotchi** (línea 23):

```
1 package Entorno;
2
3 public class Tamagotchi {
4
5 //Atributos (los declaro estáticos para usarlos en todo el programa, ya que me harán falta):
6
7     private static int hanger;
8     private static int energy;
9     private static int mood;
10
11 //Constructor (por defecto sus atributos serán 4):
12
13 public Tamagotchi() {
14
15     this.hanger=4;
16     this.energy=4;
17     this.mood=4;
18
19 }
20
21 //Método Play:
22
23 public String Play() {
24
25     hanger+=1;
26     mood+=1;
27     energy-=1;
28
29 //Comprueba su estado:
30
31     if(mood>8) {
```



Estado de las variables:

(x)= Variables × Breakpoints Expressions	
Name	Value
no method return value	
> • this	Tamagotchi (id=20)