



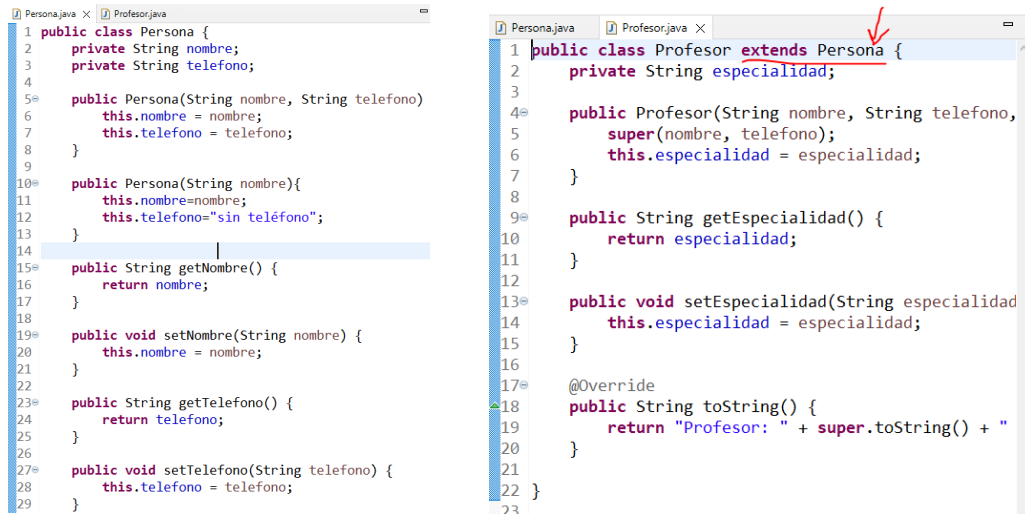
Hito Individual

Jonathan Hierro 1ºDAM

Campus FP Humanes

1. POO en Java y colecciones

1.1 Diseña un ejemplo de clases en donde puedas demostrar la herencia en Java. Explica si es posible realizar herencia múltiple en Java y por qué.



```
1 public class Persona {
2     private String nombre;
3     private String telefono;
4
5     public Persona(String nombre, String telefono) {
6         this.nombre = nombre;
7         this.telefono = telefono;
8     }
9
10    public Persona(String nombre) {
11        this.nombre = nombre;
12        this.telefono = "sin teléfono";
13    }
14
15    public String getNombre() {
16        return nombre;
17    }
18
19    public void setNombre(String nombre) {
20        this.nombre = nombre;
21    }
22
23    public String getTelefono() {
24        return telefono;
25    }
26
27    public void setTelefono(String telefono) {
28        this.telefono = telefono;
29    }
30 }

1 public class Profesor extends Persona {
2     private String especialidad;
3
4     public Profesor(String nombre, String telefono,
5                     String especialidad) {
6         super(nombre, telefono);
7         this.especialidad = especialidad;
8     }
9
10    public String getEspecialidad() {
11        return especialidad;
12    }
13
14    public void setEspecialidad(String especialidad) {
15        this.especialidad = especialidad;
16    }
17
18    @Override
19    public String toString() {
20        return "Profesor: " + super.toString() + "
21    }
22 }
```

¿Por qué Java no admite herencia múltiple?

- Conflictos de nombres y ambigüedad. Cuando los compiladores de los lenguajes de programación que soportan este tipo de superclases de sucesión múltiple contienen métodos con el mismo nombre, a veces no pueden determinar a qué miembro o método acceder o invocar.
- Además, un programador puede involuntariamente introducir un conflicto de nombres añadiendo un nuevo método a una superclase.

1.2 Siguiendo el ejercicio anterior, puedes proponer un ejemplo para diferenciar sobrecarga y sobrescritura. El ejemplo funcionará en consola.

```
21 //Esto es sobrescritura
22 @Override
23 public String andar() {
24     return "Está andando";
25 }
26 //Esto es sobrecarga
27 public String andar(int cuanto) {
28     return "El profesor ha andado "+cuanto+" metros";
29 }
30
31
32

40 // Métodos abstractos, deben ser implementados
41 // en una clase derivada.
42 public abstract String andar();
43 public abstract String correr();
44
45 }
46
```

```
1
2 public class Principal {
3
4     public static void main(String[] args) {
5         Profesor miProfesor = new Profesor("Alvaro", "66666666", "Matematicas");
6
7         System.out.println(miProfesor.andar(10));
8         System.out.println(miProfesor.andar(0));
9
10    }
11
12 }
13
```

Problems Javadoc Declaration Console X
<terminated> Principal (2) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (19 may 2022 13:09:56 - 13:09:56) [pid: 8904]
El profesor ha andado 10 metros ←
Está andando ←

1.3 En Java existen varias opciones para almacenar datos en colecciones. Explica con un ejemplo qué diferencia hay entre colecciones de tipo lista, pila y cola.

Listas

Una lista (list) es una colección de objetos donde cada uno de ellos lleva un índice asociado. Así, podríamos tener una lista con los nombres de las personas que han utilizado un servicio de acceso a internet que podría ser: usuarios --> (Juan R.R., Sara G.B., Rodolfo M.N., Pedro S.T., Claudio R.S., Juan R.R.). Donde cada contenido va asociado a un índice, usuario(0) sería Juan R.R., usuario(1) sería Sara G.B, usuario(2) sería Rodolfo M.N. y así sucesivamente. En una lista podemos insertar y eliminar objetos de posiciones intermedias. Ejemplos de listas son la clase ArrayList y LinkedList del API de Java.

Cola

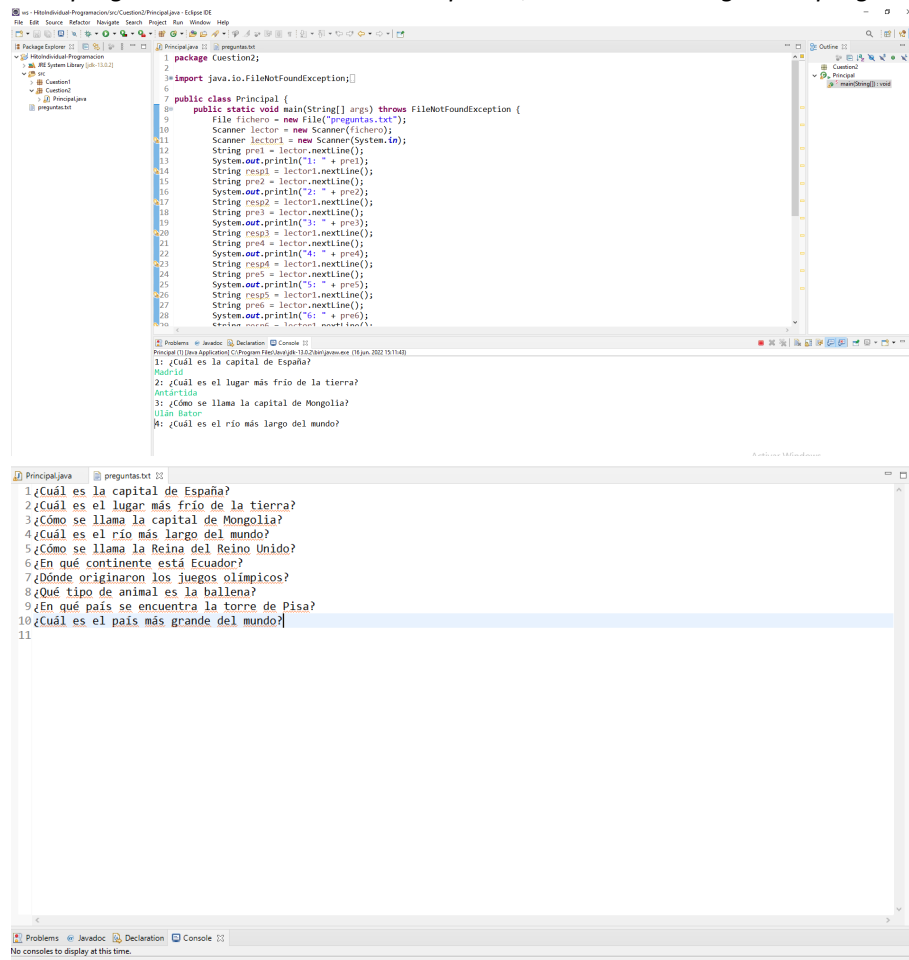
Una cola (queue) sería una colección de objetos que se comportan como lo haría un grupo de personas en la cola de una caja de un supermercado. Los objetos se van poniendo en cola y el primero en salir es el primero que llegó.

Pila

Una pila (stack) sería una colección de objetos que se comportan como lo haría un montón de objetos apilados, el último en añadirse queda en la cima de la pila y el primero en salir es el último que ha llegado. El primero en llegar está en el fondo de la pila.

2. Acceso a información con ficheros.

2.1 Crea un archivo de texto con el enunciado de 10 preguntas. Desde consola muestra cada pregunta. Cuando el usuario responde, se muestra la siguiente pregunta.



The image shows a screenshot of an IDE (Eclipse) with a Java project named 'H01Individual-Programacion'. The project contains a package 'Cuestion2' with a file 'Principal.java' and a text file 'preguntas.txt'.

The 'Principal.java' file contains the following code:

```
1 package Cuestion2;
2
3 import java.io.FileNotFoundException;
4
5
6
7 public class Principal {
8     public static void main(String[] args) throws FileNotFoundException {
9         File fichero = new File("preguntas.txt");
10        Scanner lector = new Scanner(fichero);
11        Scanner lector1 = new Scanner(System.in);
12        String pre1 = lector.nextLine();
13        System.out.println("1: " + pre1);
14        String resp1 = lector1.nextLine();
15        String pre2 = lector.nextLine();
16        System.out.println("2: " + pre2);
17        String resp2 = lector1.nextLine();
18        String pre3 = lector.nextLine();
19        System.out.println("3: " + pre3);
20        String resp3 = lector1.nextLine();
21        String pre4 = lector.nextLine();
22        System.out.println("4: " + pre4);
23        String resp4 = lector1.nextLine();
24        String pre5 = lector.nextLine();
25        System.out.println("5: " + pre5);
26        String resp5 = lector1.nextLine();
27        String pre6 = lector.nextLine();
28        System.out.println("6: " + pre6);
29    }
30 }
```

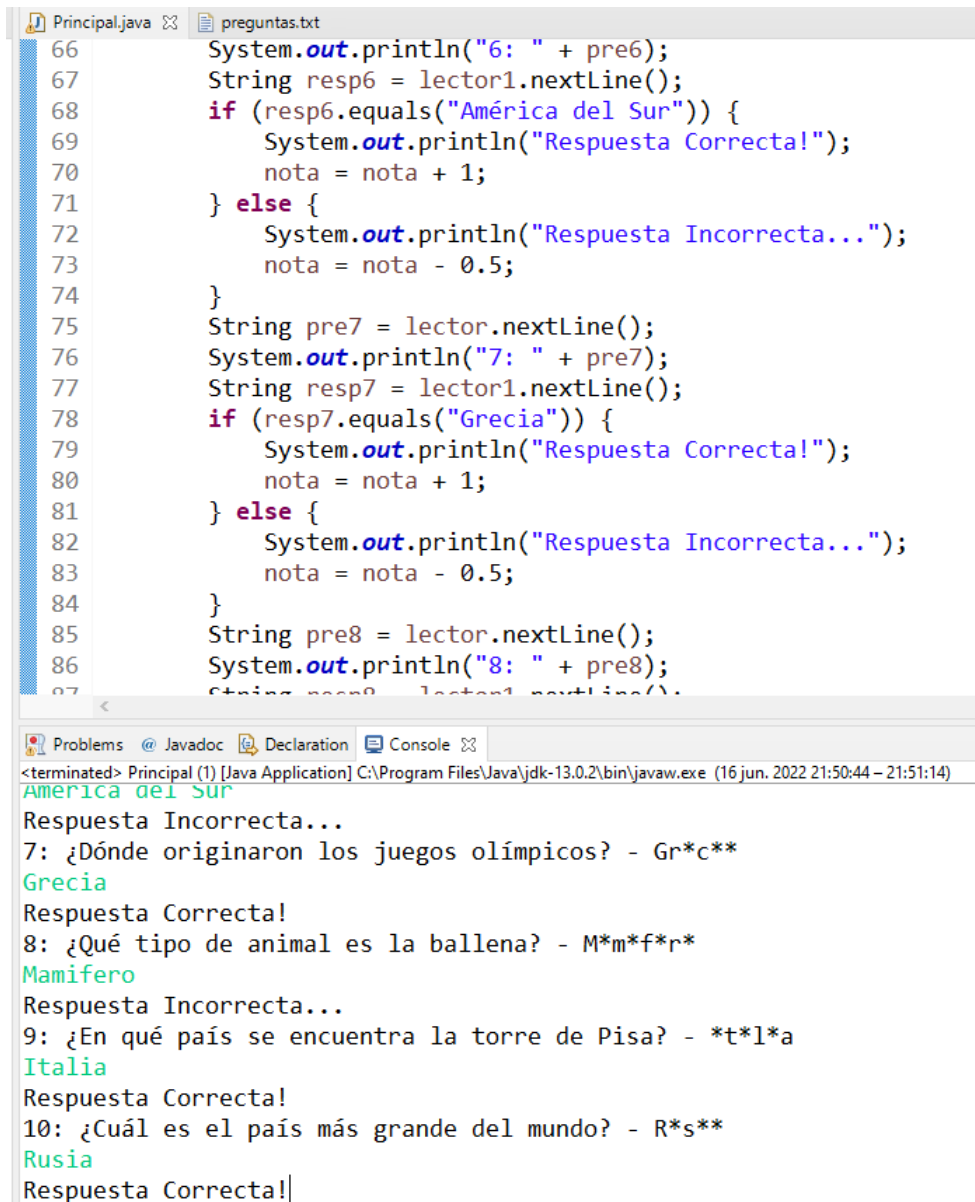
The console output shows the program running and displaying the first six questions from the file:

```
1: ¿Cuál es la capital de España?
Madrid
2: ¿Cuál es el lugar más frío de la tierra?
Antártida
3: ¿Cómo se llama la capital de Mongolia?
Ulan Bator
4: ¿Cuál es el río más largo del mundo?
5: ¿Cómo se llama la Reina del Reino Unido?
6: En qué continente está Ecuador?
```

The text file 'preguntas.txt' contains the following 10 questions:

```
1 ¿Cuál es la capital de España?
2 ¿Cuál es el lugar más frío de la tierra?
3 ¿Cómo se llama la capital de Mongolia?
4 ¿Cuál es el río más largo del mundo?
5 ¿Cómo se llama la Reina del Reino Unido?
6 En qué continente está Ecuador?
7 ¿Dónde originaron los juegos olímpicos?
8 ¿Qué tipo de animal es la ballena?
9 ¿En qué país se encuentra la torre de Pisa?
10 ¿Cuál es el país más grande del mundo?
11
```

2.2 En otro fichero de texto, o en el mismo anterior, añadimos las respuestas a las preguntas. Ahora, cuando el usuario responde, indica si la respuesta es correcta o no.



The screenshot shows an IDE with two tabs: 'Principal.java' and 'preguntas.txt'. The 'Principal.java' tab is active, displaying Java code from line 66 to 87. The code uses `System.out.println` to print questions and `lector1.nextLine()` to read user input. It includes `if` and `else` blocks to check if the input matches the expected answer (e.g., "América del Sur", "Grecia") and update a `nota` variable accordingly. The console window at the bottom shows the execution output, including the questions, user input, and the program's feedback.

```
66 System.out.println("6: " + pre6);
67 String resp6 = lector1.nextLine();
68 if (resp6.equals("América del Sur")) {
69     System.out.println("Respuesta Correcta!");
70     nota = nota + 1;
71 } else {
72     System.out.println("Respuesta Incorrecta...");
73     nota = nota - 0.5;
74 }
75 String pre7 = lector1.nextLine();
76 System.out.println("7: " + pre7);
77 String resp7 = lector1.nextLine();
78 if (resp7.equals("Grecia")) {
79     System.out.println("Respuesta Correcta!");
80     nota = nota + 1;
81 } else {
82     System.out.println("Respuesta Incorrecta...");
83     nota = nota - 0.5;
84 }
85 String pre8 = lector1.nextLine();
86 System.out.println("8: " + pre8);
87 String resp8 = lector1.nextLine();
```

Problems @ Javadoc Declaration Console

<terminated> Principal (1) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (16 jun. 2022 21:50:44 - 21:51:14)

América del Sur
Respuesta Incorrecta...
7: ¿Dónde originaron los juegos olímpicos? - Gr*c**
Grecia
Respuesta Correcta!
8: ¿Qué tipo de animal es la ballena? - M*m*f*r*
Mamifero
Respuesta Incorrecta...
9: ¿En qué país se encuentra la torre de Pisa? - *t*l*a
Italia
Respuesta Correcta!
10: ¿Cuál es el país más grande del mundo? - R*s**
Rusia
Respuesta Correcta!

2.3 Muestra la puntuación obtenida por el usuario. Cada respuesta acertada suma un punto, y cada errónea resta 0.5. La nota necesaria para aprobar es un 5.

```
Principal.java  preguntas.txt
100     nota = nota + 1;
101     } else {
102         System.out.println("Respuesta Incorrecta...");
103         nota = nota - 0.5;
104     }
105     String pre10 = lector.nextLine();
106     System.out.println("10: " + pre10);
107     String resp10 = lector1.nextLine();
108     if (resp10.equals("Rusia")) {
109         System.out.println("Respuesta Correcta!");
110         nota = nota + 1;
111     } else {
112         System.out.println("Respuesta Incorrecta...");
113         nota = nota - 0.5;
114     }
115     System.out.println("-----FIN DEL TEST-----");
116     System.out.println("-----NOTA TEST: " + nota + "----");
117     lector.close();
118 }
119
120 }
```

<terminated> Principal (1) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (16 jun. 2022 21:55:49 – 21:56:16)

America del Sur
Respuesta Incorrecta...
7: ¿Dónde originaron los juegos olímpicos? - Gr*c**
Grecia
Respuesta Correcta!
8: ¿Qué tipo de animal es la ballena? - M*m*f*r*
Mamifero
Respuesta Incorrecta...
9: ¿En qué país se encuentra la torre de Pisa? - *t*l*a
Italia
Respuesta Correcta!
10: ¿Cuál es el país más grande del mundo? - R*s**
Rusia
Respuesta Correcta!
-----FIN DEL TEST-----
-----NOTA TEST: 4.0-----