

Navegación por el cuestionario

1	2	3	4	5	6	7	8	9	10
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Finalizar revisión

Comenzado el	martes, 11 de diciembre de 2018, 19:07
Estado	Finalizado
Finalizado en	miércoles, 12 de diciembre de 2018, 19:06
Tiempo empleado	23 horas 58 minutos
Calificación	9,07 de 10,00 (91%)

Pregunta 1
Correcta
Puntuó 0,67 sobre 1,00
Marcar pregunta

Indica cuáles son características del objeto System.out:

Seleccione una o más de una:

a. Solo permite usar los métodos print y println

b. No funciona con números

c. Permite escribir texto por consola ✓

d. Es un objeto de la clase PrintStream ✓

e. Maneja la salida estándar del sistema ✓

f. Permite leer datos del teclado

Respuesta correcta

La respuesta correcta es: Es un objeto de la clase PrintStream, Maneja la salida estándar del sistema, Permite escribir texto por consola

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,67/1,00.

Pregunta 2
Correcta
Puntuó 1,00 sobre 1,00
Marcar pregunta

Completa el siguiente código para que pida el nombre de un fichero hasta que éste exista y se pueda leer:

```
import java.io.*;
[import java.util.*]

public class Programa {
    public static void main(String[] args) [throws FileNotFoundException] {
        Scanner teclado = new Scanner(System.in);

        System.out.print("Nombre del fichero: ");
        String nombre = teclado.nextLine();
        [File fichero = new File(nombre);]

        while ([fichero.canRead()]) {
            System.out.println("No se puede leer este fichero. Inténtalo otra vez.");
            System.out.print("Nombre del fichero: ");
            nombre = teclado.nextLine();
            [fichero = new File(nombre);]
        }

        Scanner leerFichero = new Scanner(fichero);
    }
}
```

nada

File fichero = new File(nombre);

fichero.canRead()

String nombre = teclado.nextLine();

Respuesta correcta

La respuesta correcta es:

Completa el siguiente código para que pida el nombre de un fichero hasta que éste exista y se pueda leer:

```
import java.io.*;
[import java.util.*]

public class Programa {
    public static void main(String[] args) [throws FileNotFoundException] {
        Scanner teclado = new Scanner(System.in);

        System.out.print("Nombre del fichero: ");
        [String nombre = teclado.nextLine();]
        [File fichero = new File(nombre);}

        while (!fichero.canRead()) {
            System.out.println("No se puede leer este fichero. Inténtalo otra vez.");
            System.out.print("Nombre del fichero: ");
            [nombre = teclado.nextLine();]
            [fichero = new File(nombre);]

        }

        [Scanner leerFichero = new Scanner(fichero);

        // Aquí procesamos el fichero

        leerFichero.close();
    }
}
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 3
Correcta
Puntuó 1,00 sobre 1,00
Marcar pregunta

Completa el siguiente código para que lea el fichero entrada.txt y escriba su contenido en salida.txt.

```
import java.io.*;
import java.util.*;

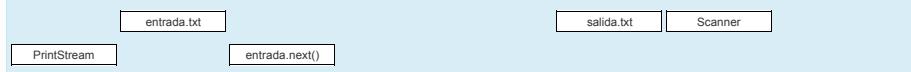
public class Programa {
    public static void main(String[] args) throws FileNotFoundException {
        File fichEntrada = new File ["entrada.txt"];
        Scanner [entrada = new Scanner fichEntrada];
        File [fichSalida = new File "salida.txt"];
        PrintStream [salida = new PrintStream fichSalida];

        while (entrada.hasNextLine()) {
            String linea = entrada.nextLine();
        }
    }
}
```

```

        salida.println(linea);
    }
    entrada.close();
    salida.close();
}

```



Respuesta correcta

La respuesta correcta es:

Completa el siguiente código para que lea el fichero entrada.txt y escriba su contenido en salida.txt.

```

import java.io.*;
import java.util.*;

public class Programa {
    public static void main(String[] args) throws FileNotFoundException {
        File fichEntrada = new File(["entrada.txt"]);
        [Scanner] entrada = new [Scanner](fichEntrada);

        File [fichSalida] = new File(["salida.txt"]);
        [PrintStream] salida = new [PrintStream](fichSalida);

        while (entrada.hasNextLine()) {
            String linea = [entrada.nextLine()];
            salida.println([linea]);
        }
        entrada.close();
        salida.close();
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4

Correcta

Puntuá 1,00 sobre
1,00

▼ Marcar
pregunta

Tenemos el siguiente método:

```

public static void escribir(PrintStream escribirTexto) {
    escribirTexto.print("Empezamos");
    escribirTexto.println(" a trabajar");
    escribirTexto.printf("Entero: %d", 2345);
    escribirTexto.println();
    escribirTexto.printf("Decimal: %.1f\n", 15.67823);
}

```

¿Qué veremos en el fichero escribir.txt después de ejecutar el siguiente código?

```

PrintStream escribirFichero = new PrintStream(new File("escribir.txt"));
escribir(escribirFichero);

```

Respuesta: (penalty regime: 10, 20, ... %)

1 Empezamos a trabajar	2 Entero: 2345	3 Decimal: 15,7
--------------------------	------------------	-------------------

Todas las pruebas superadas. ✓

Question author's solution:

```

Empezamos a trabajar
Entero: 2345
Decimal: 15,7

```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 5

Correcta

Puntuá 1,00 sobre
1,00

▼ Marcar
pregunta

Tenemos el siguiente método:

```

public static void escribir(PrintStream escribirTexto) {
    escribirTexto.println("Linea 1");
    escribirTexto.print("Kaixo");
    escribirTexto.printf("Numero: %d", 45);
    escribirTexto.println();
    escribirTexto.printf("Numero: %.2f\n", 145.67823);
}

```

¿Qué mostrará por consola si hacemos la siguiente llamada?

```

escribir(System.out);

```

Respuesta: (penalty regime: 10, 20, ... %)

1 Linea 1	2 KaixoNumero: 45	3 Numero: 145,68
-------------	---------------------	--------------------

Expected	Got	Comment	Mark
✓ Línea 1 KaixoNumero: 45 Número: 145,68	Línea 1 KaixoNumero: 45 Número: 145,68	Line 0 right 1 Line 1 right Line 2 right	✓

Todas las pruebas superadas. ✓

Question author's solution:

```
Línea 1
KaixoNumero: 45
Número: 145,68
```

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6

Correcta

Puntuá 0,50 sobre
1,00

▼ Marcar
pregunta

Elige las afirmaciones que describen a las excepciones InputMismatchException .

Seleccione una o más de una:

- a. Pertenecen al grupo de excepciones denominadas "unchecked" ✓
- b. Provocan errores de compilación
- c. Se producen cuando tratamos de leer con Scanner un dato del tipo equivocado ✓
- d. Se produce cuando usamos el método next() de la clase Scanner

Respuesta correcta

La respuesta correcta es: Pertenecen al grupo de excepciones denominadas "unchecked", Se producen cuando tratamos de leer con Scanner un dato del tipo equivocado

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,50/1,00.

Pregunta 7

Correcta

Puntuá 1,00 sobre
1,00

▼ Marcar
pregunta

Tenemos el siguiente main:

```
public static void main(String[] args) throws FileNotFoundException {
    System.out.println("Mensaje 1");
    Scanner leerFichero = new Scanner(new File("datos.txt"));
    System.out.println("Mensaje 2");
}
```

Indica qué líneas se mostrarán por consola, si se produce una excepción al intentar conectar con el fichero "datos.txt"

Seleccione una o más de una:

- a. Nada
- b. Mensaje 1 ✓
- c. Mensaje 2

Respuesta correcta

La respuesta correcta es: Mensaje 1

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 8

Correcta

Puntuá 1,00 sobre
1,00

▼ Marcar
pregunta

Elige las afirmaciones que describen a la sentencia catch

Seleccione una o más de una:

- a. Entre paréntesis debemos identificar el tipo de excepción que queremos detectar ✓
- b. Incluye el código que queremos que se ejecute cuando se produce una excepción ✓
- c. Por cada sentencia try solo podemos tener un catch
- d. Siempre debe ir acompañado de finally

Respuesta correcta

La respuesta correcta es: Incluye el código que queremos que se ejecute cuando se produce una excepción, Entre paréntesis debemos identificar el tipo de excepción que queremos detectar

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 9

Correcta

Puntuá 0,90 sobre
1,00

▼ Marcar
pregunta

Completa el programa con el método buscarEdad() al que se le pasarán un objeto de tipo Scanner conectado a un fichero y un nombre. Leerá el fichero y devolverá la edad de esa persona. Si el nombre no está en el fichero devolverá -1.

El fichero sigue el siguiente formato:

```
Marta 25 Nerea 30 Miren 27 Aitor 41 Javier 36
```

El resultado de las siguientes llamadas debería ser:

```
resultado = buscarEdad(leerFich, "marta");
devolverá 25

resultado = buscarEdad(leerFich, "Ramon");
devolverá -1
```

Respuesta: (penalty regime: 10, 20, ... %)

Reiniciar respuesta

```
1 import java.util.*;
2 import java.io.*;
3
4 public class Ficheros {
5     public static void main(String[] args) throws FileNotFoundException {
6         Scanner leerFich = new Scanner(new File("edades.dat"));
7         Scanner leerTeclado = new Scanner(System.in);
8
9         System.out.print("Nombre: ");
10        String buscar = leerTeclado.next();
11
12        int edad = buscarEdad(leerFich, buscar);
13
14        if (edad == -1) {
15            System.out.println("Ese nombre no esta en la lista");
16        } else {
17            System.out.println("La edad de " + buscar + " es " + edad);
18        }
19
20    leerFich.close();
```

Input	Expected	Got
✓ marta	Nombre: La edad de marta es 25	Nombre: La edad de marta es 25 ✓
✓ Raul	Nombre: Ese nombre no esta en la lista	Nombre: Ese nombre no esta en la lista ✓

Todas las pruebas superadas. ✓

Question author's solution:

```
import java.util.*;
import java.io.*;
```

```

public class Ficheros {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner leerFich = new Scanner(new File("edad.dat"));
        Scanner leerTeclado = new Scanner(System.in);

        System.out.print("Nombre: ");
        String buscar = leerTeclado.next();

        int edad = buscarEdad(leerFich, buscar);

        if (edad == -1) {
            System.out.println("Ese nombre no esta en la lista");
        } else {
            System.out.println("La edad de " + buscar + " es " + edad);
        }

        leerFich.close();
    }

    public static int buscarEdad(Scanner leer, String buscar) {
        while (leer.hasNext()) {
            String nombre = leer.next();
            int edad = leer.nextInt();
            if (nombre.equalsIgnoreCase(buscar)) {
                return edad;
            }
        }
        return -1;
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,90/1,00.

Pregunta 10

Correcta

Puntúa 1,00 sobre
1,00

▼ Marcar
pregunta

Crea el método conectarFichero que recibirá como parámetro un objeto de la clase Scanner para leer por teclado y devolverá otro objeto de la clase Scanner conectado al fichero introducido por el usuario. El método pedirá el nombre del fichero por teclado hasta que el usuario o la usuaria introduzca uno válido, que pueda ser leído. A continuación, conectará con ese fichero y devolverá el Scanner conectado.

Cuando se llame al método mostrará algo parecido a:

```

Nombre del fichero:  
nombres.dat  
Este fichero no se puede leer. Nombre:  
entrada.txt  
Fichero valido.  
  
Devuelve el Scanner conectado a ese fichero

```

Respuesta: (penalty regime: 10, 20, ... %)

[Reiniciar respuesta](#)

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Programa {
5     public static void main(String[] args) throws FileNotFoundException {
6         Scanner teclado = new Scanner(System.in);
7         Scanner fichero = conectarFichero(teclado);
8         while (fichero.hasNextLine()) {
9             System.out.println(fichero.nextLine());
10        }
11        fichero.close();
12    }
13    public static Scanner conectarFichero(Scanner teclado) throws FileNotFoundException{
14        System.out.println("Nombre del fichero: ");
15        String nombreFich = teclado.nextLine();
16        File fichero = new File(nombreFich);
17        while (!fichero.canRead()){
18            System.out.println("Este fichero no se puede leer. Nombre: ");
19            nombreFich = teclado.nextLine();
20            fichero = new File(nombreFich);
21        }
22    }
}

```

Input	Expected	Got
✓ nombres.dat entrada.txt	Nombre del fichero: Este fichero no se puede leer. Nombre: Fichero valido. rojo 56 78,45 901 amarillo 34,33	Nombre del fichero: Este fichero no se puede leer. Nombre: Fichero valido. rojo 56 78,45 901 amarillo 34,33
	Todas las pruebas superadas. ✓	

Question author's solution:

```

import java.io.*;
import java.util.*;

public class Programa {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner teclado = new Scanner(System.in);
        Scanner fichero = conectarFichero(teclado);
        while (fichero.hasNextLine()) {
            System.out.println(fichero.nextLine());
        }
        fichero.close();
    }

    public static Scanner conectarFichero(Scanner leerTeclado) throws FileNotFoundException {
        System.out.println("Nombre del fichero: ");
        String nombre = leerTeclado.nextLine();
        File fichero = new File(nombre);
        while(!fichero.canRead()){
            System.out.println("Este fichero no se puede leer. Nombre: ");
            nombre = leerTeclado.nextLine();
            fichero = new File(nombre);
        }
        System.out.println("Fichero valido.");
        return new Scanner(fichero);
    }
}

```

Correcta

Puntos para este envío: 1,00/1,00.

[Finalizar revisión](#)



Urrutiko Lanbide Heziketako Institutua
Instituto de Formación Profesional a Distancia

Contacta con nosotros

- ❖ Dirección: Calle Álava 41, interior - Vitoria-Gasteiz
- ❖ Teléfono: 945 567 953
- ✉ E-mail: ulhi@ulhi.es
- ❖ Twitter: [@UrrutikoLH](https://twitter.com/UrrutikoLH)



EUSKAL
JAUARIARIA
GOBIERNO VASCO

HEZKUNTZA SAIA
Lanbide Heziketako Sailburuostea

DEPARTAMENTO DE EDUCACIÓN

Ministerio de Formación Profesional