

BOLETÍN 1

1 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        int i = 42;  
        String s = (i<40)?"life":(i>50)?"universe":"everything";  
        System.out.println(s);  
    }  
}
```

¿Cuál es el resultado?

- a. null
 - b. life
 - c. **everything**
 - d. Fallo de compilación
 - e. Excepción en tiempo de ejecución
-

2 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        float f1 = 2.3f;  
        float[][] f2 = {{42.0f}, {1.7f, 2.3f}, {2.6f, 2.7f}};  
        float[] f3 = {2.7f};  
        long x = 42L;  
        // insertar código aquí  
        System.out.println("true");  
    }  
}
```

Y los siguientes fragmentos de código a insertar:

- 1. if (f1 == f2)
- 2. if (f1 == f2[2][1])
- 3. if(x == f2[0][0])
- 4. if(f1 == f2[1,1])
- 5. if(f2 == f2[2])

¿Cuál es verdadera?

- a. Compilará una única opción y se mostrará true.
- b. **Compilará con dos opciones y sólo con una se mostrará true.**
- c. Compilará con dos opciones y con ambas se mostrará true.
- d. Compilará con tres opciones y sólo con una se mostrará true.
- e. Compilará con tres opciones y con las tres se mostrará true
- f. Compilará con tres opciones y sólo con dos será true.

3 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        String s = "";  
        boolean b1 = true;  
        boolean b2 = false;  
        if (b2 = false | (21%5) > 2) s += "x";  
        if (b1 || (b2 = true)) s += "y";  
        if (b2 == true) s += "z";  
        System.out.println(s);  
    }  
}
```

¿Cuáles son verdaderas?

- a. Fallo de compilación
 - b. El caracter x será incluido en lo que se muestra por pantalla.
 - c. El caracter y será incluido en lo que se muestra por pantalla.**
 - d. El caracter z será incluido en lo que se muestra por pantalla.
 - e. Excepción en tiempo de ejecución
-

4 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        int mask = 0;  
        int count = 0;  
        if ((5<7) || (++count < 10) | mask++ < 10)  
            mask = mask + 1;  
        if ((6 > 8) ^ false) mask = mask + 10;  
        if ( !(mask > 1) && ++count > 1) mask = mask + 100;  
        System.out.println(mask + " " + count );  
    }  
}
```

¿Cuáles de las siguientes opciones son verdaderas?

- a. mask es 0
- b. mask es 1
- c. mask es 2**
- d. mask es 10
- e. mask es mayor que 10
- f. count es mayor que 0

5 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        int[] x = {7,6,5,4,3,2,1};  
        // inserta el código aquí  
        System.out.println(y + " ");  
    }  
}
```

¿Cuáles de las siguientes opciones insertada en la línea comentada compila correctamente?

- a. `for(int y:x)`
 - b. `for(x : int y)`
 - c. `int y = 0; for(y : x)`
 - d. `for(int y=0, z=0; z<x.length; z++) { y = x[z];`
 - e. `for(int y=0, int z=0; z<x.length; z++) { y = x[z];`
 - f. **`int y = 0; for(int z=0; z<x.length; z++) { y = x[z];`**
-

6 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        int[] ia = {1,3,5,7,9};  
        for(int x: ia) {  
            for(int j=0; j<3; j++) {  
                if (x>4 && x<8) continue;  
                System.out.println(" " + x);  
                if (j==1) break;  
                continue;  
            }  
            continue;  
        }  
    }  
}
```

¿Cuál será el resultado?

- a. 1 3 9
- b. 5 5 7 7
- c. **1 3 3 9 9**
- d. 1 1 3 3 9 9
- e. 1 1 1 3 3 3 9 9 9
- f. Fallo de compilación

7 Dado el siguiente código

```
public class Ejercicio {  
    public static void main(String[] args) {  
        foreach:  
        for(int j=0; j<5; j++) {  
            for(int k=0; k<3; k++) {  
                System.out.println(" " + j);  
                if (j==3 && k==1) break foreach;  
                if (j==0 || j==2) break;  
            }  
        }  
    }  
}
```

¿Cuál será el resultado?

- a. 0 1 2 3
- b. 1 1 1 3 3
- c. **0 1 1 1 2 3 3**
- d. 1 1 1 3 3 4 4 4
- e. 0 1 1 1 2 3 3 4 4 4
- f. Fallo de compilación

8 Dado el siguiente código:

```
public class Ejercicio {  
    public static void main(String[] args) {  
        String s1 = "abc";  
        String s2 = s1;  
        s1 += "d";  
        System.out.println(s1 + " " + s2 + " " + (s1==s2));  
        StringBuffer sb1 = new StringBuffer("abc");  
        StringBuffer sb2 = sb1;  
        sb1.append("d");  
        System.out.println(sb1 + " " + sb2 + " " + (sb1 ==  
sb2));  
    }  
}
```

¿Cuáles de las siguientes opciones son verdaderas?

- a. Fallo de compilación
- b. La primera línea de la salida es abc abc true
- c. **La primera línea de la salida es abc abc false**
- d. La primera línea de la salida es abcd abc false
- e. **La segunda línea de la salida es abcd abcd true**
- f. La segunda línea de la salida es abcd abcd false

9 Dado el siguiente fragmento de código:

```
public class Ejercicio {
    public static void main(String[] args) {
        for(char a = ' '; Math.random() < 0.9; System.gc()) {
            System.out.println("Hey!");
        }
    }
}
```

¿Cuáles de las siguientes opciones son verdaderas?

- a. No compila
- b. Al menos se mostrará un Hey! por pantalla
- c. **Compila y puede ser que se muestre o no se muestre un Hey! por pantalla**
- d. **El uso de la memoria es eficiente**
- e. Excepción en tiempo de ejecución

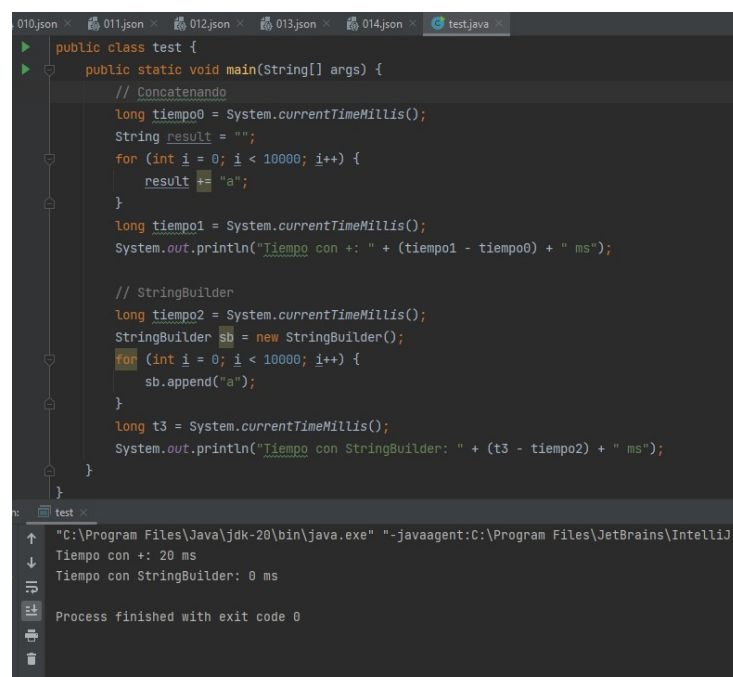
10 Escribir una clase con un método *main* donde se concatene un *String* 10000 veces de dos formas:

- 1. Usando el operador +
- 2. Usando *StringBuilder*

Medir, de manera aproximada los tiempos de ejecución en ambos casos y mostrarlos por pantalla.

Tip: si quiero medir el tiempo de ejecución de un fragmento de código, puedo hacer:

```
long t0 = System.currentTimeMillis();
/*
 * FRAGMENTO DE CÓDIGO QUE DESEO MEDIR
 */
long t1 = System.currentTimeMillis();
System.out.println("Tiempo: " + (t1-t0));
```



```
public class test {
    public static void main(String[] args) {
        // Concatenando
        long tiempo0 = System.currentTimeMillis();
        String result = "";
        for (int i = 0; i < 10000; i++) {
            result += "a";
        }
        long tiempo1 = System.currentTimeMillis();
        System.out.println("Tiempo con +: " + (tiempo1 - tiempo0) + " ms");

        // StringBuilder
        long tiempo2 = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 10000; i++) {
            sb.append("a");
        }
        long t3 = System.currentTimeMillis();
        System.out.println("Tiempo con StringBuilder: " + (t3 - tiempo2) + " ms");
    }
}
```

test

"C:\Program Files\Java\jdk-20\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ..."

Tiempo con +: 20 ms

Tiempo con StringBuilder: 0 ms

Process finished with exit code 0