

Oracle Certified Professional Java SE 17 Developer

Chapter 4: Core APIs (pdf - 276)

1. F: El código no compila

En la línea 5 trata de asignar un int a un String

2. C: `String beans = new beans [6];`

E: `int[][] types = new int[];`

F: `int[][] java = new int[][];`

En la opción C no debería estar el nombre de la variable después de new. En la opción E, no se puede declarar una matriz de un lado y un arreglo del otro. En la opción F, el primer nivel del arreglo no está inicializado

3. A: `LocalDate.of(2022, 3, 13);`

C: `LocalDate.of(2022, 11, 6);`

D: `LocalDate.of(2022, 11, 7);`

La segunda opción no funciona ya que ningún mes tiene 40 días. La quinta opción no funciona porque el 2023 no es bisiesto, por lo que febrero tiene 28 días. La sexta opción no funciona porque ese no es el nombre correcto del enum, sería `Month.MARCH`

4. A: one

C: three

D: four

```

var s = "Hello";
var t = new String(s);
//El método equals de String compara el contenido de la cadena -> true
if("Hello".equals(s)) System.out.println("one");
//Se compara si las variables apuntan al mismo objeto, t apunta a un
objeto y s punta al pool de Strings -> false
if(t == s) System.out.println("two");
//t.intern() apunta a "Hello" dentro del pool de Strings -> true
if(t.intern() == s) System.out.println("three");
//"Hello" está en el pool de Strings, a donde apunta s -> true
if("Hello" == s) System.out.println("four");
//"Hello" esta en el pool de Strings y t apunta a un objeto -> false
if("Hello".intern() == t) System.out.println("five");

```

5. B: abbacca

```

sb.append("aaa") //sb = "aaa"
  .insert(1, "bb") //sb = "abbaa"
  .insert(4, "ccc"); //sb = "abbacca"

```

6. C: 2

Para convertir un double en float se requiere un casteo implícito (línea 25). Los elementos dentro del arreglo tendrían que ser double (línea 26)

7. A: La primer fecha/ hora es antes

E: Hay seis horas de diferencia entre ambas

```
2022-08-28T05:00 GMT-04:00 //Hora central -> 09:00
```

```
2022-08-28T09:00 GMT-06:00 //Hora central -> 15:00
```

8. A: builder.charAt(4)

```
B: builder.replace(2, 4, "6").charAt(3);
```

```
F: string.replace("123", "1").charAt(2)

builder.replace(2, 4, "6").charAt(3) //builder = "1265", el
carácter en la posición 3 es 5

builder.replace(2, 5, "6").charAt(3) //builder = "126", el
carácter en la posición 2 es 6

string.charAt(5) //Se sale de los límites de la cadena

string.length //No compila porque faltan los paréntesis
```

9. A: El primer elemento es el índice 0

C: Tienen un tamaño fijo

F: Llamar a equals() con dos arreglos diferentes que contienen los mismos números primitivos siempre retorna false

10. A: 0

11. E: El código no compila

```
var date = LocalDate.of(2022, 4, 3); //2022-04-03
date.plusDays(2); //2022-04-05
date.plusHours(3); //Error: no hay dónde agregar las horas
```

12. A: 12

D: 78

E: Una línea en blanco

```
//Agrega un espacio al inicio
var numbers = "012345678".indent(1);
numbers = numbers.stripLeading(); //Quita el espacio del inicio
System.out.println(numbers.substring(1, 3)); //12
```

```
//No imprime nada
System.out.println(numbers.substring(7, 7));
System.out.print(numbers.substring(7)); //78
```

13. B: roar roar!!!

El primer parámetro es un String, que es inmutable, por lo que los cambios no se ven reflejados en la variable. El segundo parámetro es un StringBuilder, que sí es mutable, entonces el valor de la variable sí cambia

14. A: Instant.now()

F: zonedDateTime.toInstant()

15. C: [123, PIG, pig]

E: -3

La primera línea que se imprime muestra el arreglo ordenado, que se hace de esta forma: números, mayúsculas y minúsculas. La segunda línea imprime las veces que "Pippa" fue encontrado en el arreglo, es -3 porque hay 3 elementos y ninguno coincide

16. A: 10

B: 11

G: 16

```
var base = "ewe\nsheep\\t";
//11, \n (salto de línea) cuenta como 1 y se elimina un \
int length = base.length();
//16, suma a 11 dos espacios en cada línea, también un salto de línea
int indent = base.indent(2).length();
```

```
//10, solo cuenta "ewe", "sheep", un salto de línea (\n) y un espacio  
largo (\t)  
int translate = base.translateEscapes().length //8  
var formatted = "%s %s %s".formatted(length, indent,  
translate); //11, 16, 10
```

17. A: `letter.substring(1, 2)` retorna una cadena de un carácter
- G: `letter.substring(6, 5)` arroja una exception
- `letter.substring` toma la cadena desde el índice 1 hasta el índice 1
- `letter.substring` se sale de los límites de la cadena, los parámetros indican el inicio y el fin de la subcadena

18. C: 7

F: `equals`

Las líneas 17, 18 y 19 no afectan a `s1`. En la línea 20, `s1 = "purrrtwo"`. Línea 22: `s2 = "2"`. Línea 23: `s2 = "2c"`. Línea 24: `s2 = "2cfalse"`. A pesar de que no se creó un nuevo objeto para `s2`, de las operaciones con cadenas se genera un objeto `String`, por lo que la línea 26 es `false`. De la misma manera, la línea 27 imprime `"equals"`. La línea 28 imprime 7

19. A: `compare(s1, s2)`
- B: `mismatch(s1, s2)`
- D: `mismatch(s3, s4)`

La primera opción imprime 4 ya que el método `compare` devuelve un entero positivo si `s1` es lexicográficamente mayor que `s2`. La segunda opción imprime 1 ya que `mismatch` devuelve el

índice del primer elemento diferente entre los arreglos. La tercera opción imprime -1 ya que null es mayor que nada. La cuarta opción imprime 1. La quinta opción imprime 0. La sexta opción imprime -1 porque todos los elementos son iguales

20. A: diff = 1

 D: hour = 3

```
long diff = ChronoUnit.HOURS.between(dateTime1, dateTime2)
//diff es 1 porque dateTime 2 tomo dateTime1 y le sumo 1 hora
int hour = dateTime2.getHour();
//hour es 3 por el cambio de horario, entonces, si era la 1, dateTime
sumo 1 y el cambio e horario sumo 1 = 3
Boolean offset = dateTime1.getOffset() ==
dateTime2.getOffset();
//offset es falso ya que el horario de verano cambia de -05 a -04
```

21. A: reverse()

```
C: append("vaJ$").delete(0,
3).deleteCharAt(puzzle.length() - 1)
```

La tercera opción primero agrega vaJ\$ = JavavaJ\$, luego borra 3 caracteres iniciando en el índice 0 = avaJ\$, luego borra el carácter en el índice 4 = avaJ. La cuarta opción se sale de los límites de la cadena

22. A: 2022 APRIL 30

LocalDate es immutable, por lo que las operaciones no afectan la variable