## UT1\_PD1

## Parte 1

## Ejercicio 1

Dado el siguiente programa:

public static void zoop () {

baffle ();

System.out.print ("Vos zacata ");

baffle ();

}

public static void main (String[] args) {

System.out.print ("No, yo ");

zoop ();

System.out.print ("Yo ");

baffle ();

}

public static void baffle () {

System.out.print ("pac");

ping ();

}

public static void ping () {

System.out.println (".");

}

¿Cuál es la salida? Sé preciso acerca de dónde hay espacios y dónde hay nuevas líneas.

Indicar cuál es la respuesta más correcta: (***\n*** denota nueva línea)

La salida del programa es:

No, yo pac.

Vos zacata pac.

Yo pac.

Esto se debe a que inicia su ejecución en la sexta línea (System.out.print ("No, yo ");)), luego entra en el método zoop(), el cual dentro de él llama al método baffle(), el cual imprime “pac” y luego llama al método ping para imprimir “.”. Después de esto, en una nueva línea, imprime “Vos zacata”, y al llamar nuevamente a baffle(), se imprime “pac” a su lado (además del “.” de ping() ). Por último, en main se imprime “Yo ”, siguiéndole nuevamente el funcionamiento del método baffle, quedando así la salida del programa presentada anteriormente.

## Ejercicio 2

Dado el siguiente código fuente:

public class Zumbido {

public static void desconcertar (String dirigible) {

System.out.println (dirigible);

sipo ("ping", -5);

}

public static void sipo (String membrillo, int flag) {

if (flag < 0) {

System.out.println (membrillo + " sup");

} else {

System.out.println ("ik");

desconcertar (membrillo);

System.out.println ("muaa-ja-ja-ja");

}

}

public static void main (String[] args) {

sipo ("traqueteo", 13);

}

}

a) ¿Cuál es la primera sentencia que se **ejecuta**?

b) Escribir número 2 al lado de la segunda sentencia, un 3 al lado de la que se ejecuta en

tercer lugar, y así siguiendo hasta el final del programa. Si una sentencia se ejecuta más

de una vez, puede que termine con más de un número al lado.

1. La primer sentencia que se ejecuta proviene del método sipo(String membrillo, int flag), que, como “flag” provisto desde main es positivo (flag = 13), el primer paso del programa será imprimir “ik”. En otras palabras, la primera sentencia ejecutada es la llamada al método sipo(“traqueteo”, 13)

public class Zumbido {

public static void desconcertar (String dirigible) {

System.out.println (dirigible); //4

sipo ("ping", -5); // 5

}

public static void sipo (String membrillo, int flag) {

if (flag < 0) {

System.out.println (membrillo + " sup"); //6

} else {

System.out.println ("ik"); //2

desconcertar (membrillo); // 3

System.out.println ("muaa-ja-ja-ja"); //7

}

}

public static void main (String[] args) {

sipo ("traqueteo", 13); // 1

}

}

## Ejercicio 4

1. El error se debe de que el nombre es null y más adelante se trata de usar el getNombreAdmiracion() con la función de concatenar. Para eso, modifiqué la inicialización de la variable a “ ”, y se evalúa si la cadena está vacía para poder avanzar.
2. El error en “recorrer” se debe a un problema con los índices. El primer índice de todos es el 0, el último índice en el programa será cadena.length()-1. Para eso entonces, se inicializa la iteración en 0, y se recorre a cada una unidad hasta que el índice sea menor que el largo (sale de la iteración cuando es igual al largo, lo que significa que ya se sobrepasó el último índice).
3. El error en este método es similar al anterior. En este caso particular, al ser 8 elementos del vector, el último índice será “7”, no “8”, ya que los índices irán de 0 a 8.
4. Lo mismo sucede en este caso, el primer índice es el 0, no 1. Además, se debe verificar en primer instancia si la cadena está vacía para poder retornar el contenido de dicho índice.
5. Para solucionar ese error utilicé un método de la clase adaptadora Integer (la cual convierte directamente una variable de tipo int en tipo String).