Ejercicios repaso

1. Defina con sus propias palabras los conceptos de: Entidad, atributo, comportamiento, abstracción.
2. Defina en sus propias palabras los conceptos de Clase y Objeto.
3. Qué es y para qué sirve un diagrama de clases.
4. ¿Qué es herencia? De un ejemplo.
5. ¿Qué tipo de relación es agregación y composición?
6. ¿Para qué sirven los modificadores de acceso? Nombre algunos modificadores de acceso de Java.
7. ¿Qué es una interface? De un ejemplo.
8. ¿Qué es una clase abstracta y una clase concreta?
9. ¿Cómo debe ser la sintaxis y las buenas prácticas de declaración de clases, métodos, atributos, interfaces y constantes?
10. ¿Qué tipo de comentarios con respecto a documentación conocen?
11. ¿Para qué sirven los constructores de una clase? ¿Qué reglas contemplan?
12. ¿Qué significa sobrecarga y sobreescritura de métodos? Dé un ejemplo.
13. Nombre los tipos de datos que conozca de Java.
14. Nombre y explique los principales tipos de operadores.
15. Dada la siguiente expresión, realice los pasos y construya el resultado de:

```
public static void main(String[] args) {

    int a = 9, b = 5, c = 8;

    int num = -(a++) - (c - ++b) * c + --c;

    System.out.println("Num: "+num);
    System.out.println("a: "+a);
    System.out.println("b: "+b);
    System.out.println("c: "+c);
}
```

```
int num = -(a++) - (c - ++b) * c + --c;
num = -(9) - (8 - 6) * 8 + 7;
num = -(9) - (2) * 8 + 7;
num = -(9) - 16 + 7;
num = -(2) - 16;
num = -18;
```

16. ¿Qué sentencias condicionales conoce? Dé un ejemplo de su uso.
    if, switch y else

```
if(esDeManana) {
    levantarme();
}
```

17. ¿Qué sentencias repetitivas conoce? Dé un ejemplo de su uso.

for, do-while, while

```
while(sigaDurmiendo) {
        soñar();
    }
```

18. ¿Qué es JUnit, cómo se realizan las pruebas unitarias en Java?
JUnit es una herramienta para realizar pruebas o también llamados tests para observar el comportamiento de un proyecto del lado del cliente; para usar las pruebas unitarias se deben de incluir como si fueran dependencias.

19. Escriba un método que sume los bordes de una matriz nxn.

```
   private static int sumarBordesMatrizNxN(int[][] matriz) {
        int sum = 0;
        for (int i = 0; i < matriz.length; i++) {
            for (int j = 0; j < matriz[i].length; j++) {
                    if (i == 0 || i == matriz.length - 1 || j == 0 || j ==
matriz[i].length - 1) {
                    sum += matriz[i][j];
                }
            }
        }
        return sum;

    }
```

```
0 1 1 1 1 1 0
1 1 0 0 1 1 0
1 1 0 0 1 1 1
1 0 1 0 1 1 1
0 1 0 0 0 1 0
0 0 0 0 0 1 0
1 1 0 1 0 1 0

14
```

20. Escriba un método que, dado un ArrayList de String halle la palabra que tiene más consonantes.

```
    public static void main(String[] args) {

        ArrayList<String> nombres = new ArrayList<String>();

        nombres.add("Luis");

        nombres.add("Ana");
```

```
          nombres.add("Carlos");

          nombres.add("María");

          nombres.add("Juan");

          nombres.add("Sofía");

          nombres.add("Andrés");

          nombres.add("Camila");

          nombres.add("Felipe");

          nombres.add("Valentina");

          System.out.println(imprimirMatriz(nombres));

          System.out.println("La palabra con más consonantes
es:" + buscarPalabraMasConsonantes(nombres));

     }

     private                    static                    String
buscarPalabraMasConsonantes(ArrayList<String> nombres) {

          int indiceMayor = -1;

          int cantMayor = 0;

          for (int i = 0; i < nombres.size(); i++) {

               String cadena = nombres.get(i);

               int              cantConsonantes              =
buscarCantConsonantes(cadena);

               if (cantConsonantes >= cantMayor) {

                    indiceMayor = i;

                    cantMayor = cantConsonantes;

               }

          }

          return nombres.get(indiceMayor);

     }
```

```
    private static int buscarCantConsonantes(String cadena) {
        int cant = 0;
        for (int i = 0; i < cadena.length(); i++)
            if (esConsonante(cadena.charAt(i)))
                cant++;
        return cant;
    }
    private static boolean esConsonante(char caracter) {
        final String vocales = "AEIOUaeiou";
        for (int i = 0; i < vocales.length(); i++)
            if (vocales.charAt(i) == caracter)
                return false;
        return caracter >= 'B' && caracter <= 'Z' ||
caracter >= 'b' && caracter <= 'z' || caracter == 'ñ'
                || caracter == 'Ñ';
    }
```

21. Escriba un método que Invierta un array de float.

```
private static void invertirArray(float[] numeros) {
        final float[] numeros2 = Arrays.copyOf(numeros,
numeros.length);
        for (int i = 0; i < numeros.length; i++)
            numeros[i] = numeros2[numeros.length - i - 1];
    }
    public static void main(String[] args) {
        float[] numeros = { 1.642f, 2.63f, 3.14f, 2.718f,
1.618f, 0.707f, 4.669f, 1.414f, 2.236f, 0.577f };
```

```
        invertirArray(numeros);

        System.out.println(Arrays.toString(numeros));

    }
```