

# ACTIVIDADES DE LA UNIDAD DE TRABAJO 7 – PRO

**Autor: Derimán Tejera Fumero. 1ºDAW**

**Fecha: 26/01/2022**

## Índice

Actividades de comprobación .....	2
Actividades de aplicación .....	4

## Actividades de comprobación

7.1. Dos clases se consideran vecinas siempre y cuando:

- a) Sean visibles.
- b) Ambas dispongan del mismo número de constructores.
- c) Pertenezcan al mismo paquete.
- d) Todo lo anterior ha de cumplirse para que dos clases sean vecinas.

7.2. Un miembro cuyo modificador de acceso es `private` será visible desde:

- a) Todas las clases vecinas.
- b) Todas las clases externas.
- c) Es indistinto el paquete, pero será visible siempre que se importe la clase que lo contiene.
- d) Ninguna de las respuestas anteriores.

7.3. Si desde un constructor queremos invocar a otro constructor de la misma clase, tendremos que usar

- a) `set ()`.
- b) `get ()`.
- c) `this()`.
- d) `this`.

7.4. Si por error dejamos un objeto sin ninguna referencia, siempre podremos volver a referenciarlo mediante:

- a) La referencia `this`.
- b) La referencia `null`.
- c) Utilizando `new`.
- d) Es imposible.

7.5. ¿Qué hace el operador new?

- a) Construye un objeto, invoca al constructor y devuelve su referencia.
- b) Construye un objeto, comprueba que su clase esté importada y devuelve su referencia.
- c) Busca en la memoria un objeto del mismo tipo, invoca al constructor y devuelve su referencia.
- d) Busca en memoria un objeto del mismo tipo y devuelve su referencia.

7.6. Cuando hablamos de miembros de una clase, nos estamos refiriendo a:

- a) Todos los atributos.
- b) Todos los métodos.
- c) Todos los atributos y métodos, indistintamente de los modificadores de acceso utilizados.
- d) Todos los atributos y métodos que son visibles por sus clases vecinas.

7.7. En la definición de una clase, los únicos modificadores de acceso que se pueden utilizar son:

- a) public.
- b) public y el modificador de acceso por defecto.
- c) public, el modificador de acceso por defecto y private.
- d) El modificador class.

7.8. ¿Qué diferencia un atributo estático definido en una clase de otro que no lo es?

- a) El atributo estático es visible por todas las clases vecinas, mientras que el no estático solo será visible para las clases que usen importación.
- b) Solo existe una copia del atributo estático en la clase, mientras que el atributo no estático tendrá una copia en cada uno de los objetos.
- c) Existe una copia del atributo estático en todos y cada uno de los objetos, mientras que del atributo no estático solo existe una copia en la clase.
- d) Ambos disponen de copias en cada objeto, pero el atributo no estático es accesible mediante la clase y el no estático es accesible mediante los objetos.

7.9. ¿Qué efecto tiene las siguientes líneas de código?

Cliente c;

c.nombre = "Pepita";

- a) Inicializa el atributo nombre de Cliente con el valor «Pepita».
- b) Invoca al constructor y posteriormente asigna el valor «Pepita» al atributo nombre, siempre y cuando este sea público.
- c) Si el atributo nombre es público, se le asigna un valor, pero si el atributo es privado, producirá un error.
- d) Siempre produce un error.

7.10. La ocultación de atributos puede definirse como:

- a) El proceso en el que un atributo pasa de ser público a privado.
- b) El proceso en el que se define una variable local (en un método) con el mismo identificador que un atributo.
- c) El proceso en el que un atributo estático deja de serlo.
- d) Todas las respuestas anteriores son correctas.

## Actividades de aplicación

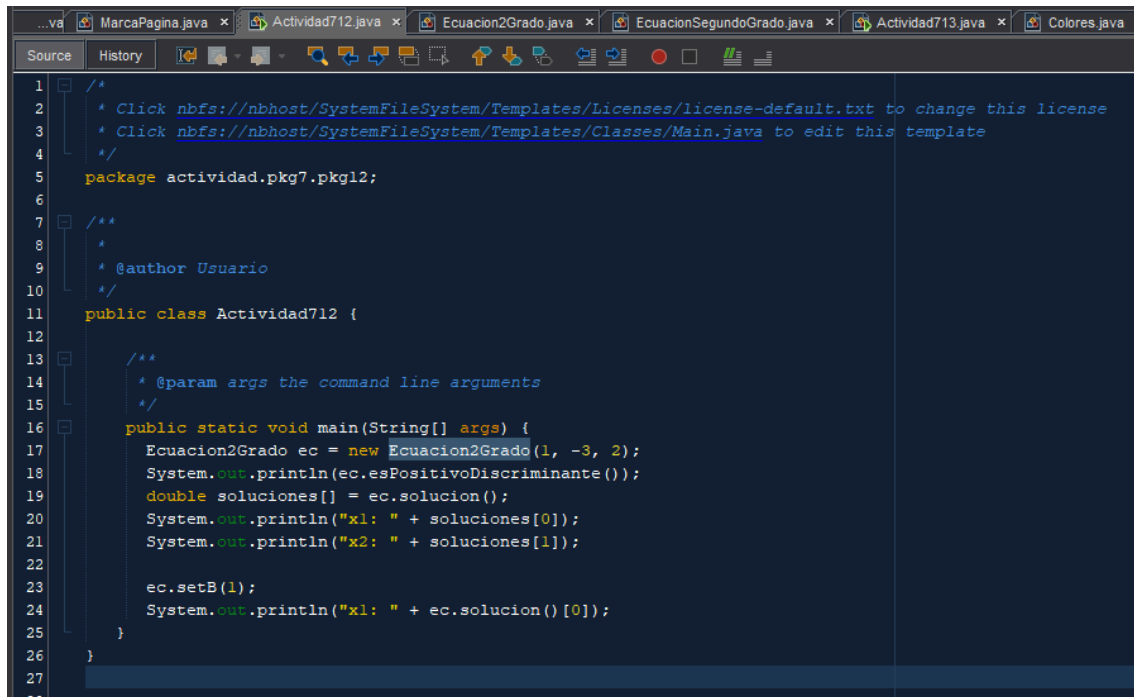
7.11. Escribe la clase MarcaPagina. que ayuda a llevar el control de la lectura de un libro. Deberá disponer de métodos para incrementar la página leída, para obtener información de la última página que se ha leído y para comenzar desde el principio una nueva lectura del mismo libro.

```
...va Punto.java x Actividad717.java x Cifrado.java x Actividad718.java x Cola.java x Actividad719.java x Pila.java x Actividad711Anterior.java x
Source History
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package actividad.pkg7.pkg11;
6
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author Usuario
12   */
13
14  public class Actividad711 {
15
16      /**
17       * @param args the command line arguments
18       */
19
20      public static void main(String[] args) {
21          Marcapagina marca;
22          marca = new Marcapagina();
23          marca.siguientePag();
24          marca.siguientePag();
25          marca.siguientePag();
26          System.out.println("Última página: " + marca.ultimaPaginaLeida());
27          marca.comenzar();
28          System.out.println("Última página: " + marca.ultimaPaginaLeida());
29      }
30  }
31
32  }
```

```
...va Punto.java x Actividad717.java x Cifrado.java x Actividad718.java x Cola.java x Actividad719.java x Pila.java x Actividad711Anterior.java x Marcapagina.java x
Source History
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package actividad.pkg7.pkg11;
6
7  /**
8   *
9   * @author Usuario
10   */
11
12  public class Marcapagina {
13
14      int paginaActual;
15
16      public Marcapagina() {
17          this.paginaActual = 0;
18      }
19
20      public void siguientePag() {
21          paginaActual++;
22          System.out.println("Página leída... ");
23      }
24
25      public int ultimaPaginaLeida() {
26          return paginaActual;
27      }
28
29      public void comenzar() {
30          paginaActual = 0;
31      }
32  }
33
34  //Autor: Derimán Tejera Fumero.
```

```
Output - Actividad 7.11 (run) x
run:
Página leída...
Página leída...
Página leída...
Última página: 3
Última página: 0
BUILD SUCCESSFUL (total time: 2 seconds)
```

7.12. Implemento una clase que permita resolver ecuaciones de segundo grado. Los coeficientes pueden indicarse en el constructor y modificarse a posteriori. Es fundamental que la clase disponga de un método que devuelva las distintas soluciones y de un método que nos informe si el discriminante es positivo.



```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package actividad.pkg7.pkg12;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class Actividad712 {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          Ecuacion2Grado ec = new Ecuacion2Grado(1, -3, 2);
18          System.out.println(ec.esPositivoDiscriminante());
19          double soluciones[] = ec.solucion();
20          System.out.println("x1: " + soluciones[0]);
21          System.out.println("x2: " + soluciones[1]);
22
23          ec.setB(1);
24          System.out.println("x1: " + ec.solucion()[0]);
25      }
26  }
27
28  }
```

```
...va  MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x EcuacionSegundoGrado.java x Actividad713.java x Colores.java
Source History
1  2  /*
3  3  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4  4  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
5  5  */
6  package actividad.pkg7.pkg12;
7
8  /**
9  9  *
10 10  * @author Usuario
11 11  */
12  public class Ecuacion2Grado {
13
14      double a;
15      double b;
16      double c;
17      double discriminante;
18
19      public Ecuacion2Grado(double a, double b, double c) {
20          this.a = a;
21          this.b = b;
22          this.c = c;
23      }
24
25      public boolean esPositivoDiscriminante() {
26          discriminante = b * b - 4 * a * c;
27          return discriminante > 0;
28      }
29
30      public double[] solucion() {
31          discriminante = b * b - 4 * a * c;
32          double solucion1 = (-b + Math.sqrt(discriminante)) / (2 * a);
33          double solucion2 = (-b - Math.sqrt(discriminante)) / (2 * a);
34          return new double[]{solucion1, solucion2};
35      }
36
37      public void setB(double b) {
38          this.b = b;
39      }
40
41      //Autor: Derimán Tejera Fumero.
```

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package actividad.pkg7.pkg12;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class Ecuacion2Grado {
12
13      double a;
14      double b;
15      double c;
16      double discriminante;
17
18      public Ecuacion2Grado(double a, double b, double c) {
19          this.a = a;
20          this.b = b;
21          this.c = c;
22      }
23
24      public boolean esPositivoDiscriminante(){
25          discriminante = b * b - 4 * a * c;
26          return discriminante > 0;
27      }
28
29      public double[] solucion(){
30          discriminante = b * b - 4 * a * c;
31          double solucion1 = (-b + Math.sqrt(discriminante)) / (2 * a);
32          double solucion2 = (-b - Math.sqrt(discriminante)) / (2 * a);
33          return new double[]{solucion1, solucion2};
34      }
35
36      public void setB(double b) {
37          this.b = b;
38      }
39  }
40  //Autor: Derimán Tejera Fumero.
41
```

Output - Actividad 7.12 (run) X

```
>> run:
>> true
x1: 2.0
x2: 1.0
x1: NaN
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.13. En el momento de decorar una casa, una habitación o cualquier objeto, se plantea el problema de elegir la paleta de colores que vamos a utilizar en nuestra decoración. Existe una solución, algo atrevida, que consiste en utilizar colores al azar.

Diseña la clase Colores, que alberga por defecto una serie de colores (mediante una cadena). aunque es posible añadir tantos como necesitemos. La clase tendrá un método que devuelve una tabla con los n colores que necesitemos elegidos al azar sin repeticiones.



```
...va Actividad711.java x MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x
Source History
1 1 /*
2 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this licens
3 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4 4  */
5 5 package actividad.pkg7.pkg13;
6 6
7 7 import java.util.Arrays;
8 8 /**
9 9  *
10 10  * @author Usuario
11 11  */
12 12 public class Actividad713 {
13 13
14 14     /**
15 15     * @param args the command line arguments
16 16     */
17 17     public static void main(String[] args) {
18 18
19 19         Colores colores = new Colores();
20 20         colores.addColor("Marrón");
21 21         colores.addColor("Azul");
22 22         colores.addColor("Amarillo");
23 23         colores.addColor("Celeste");
24 24         colores.addColor("Rosa");
25 25         colores.addColor("ColorPrueba1");
26 26         colores.addColor("ColorPrueba2");
27 27
28 28         System.out.println(Arrays.toString(colores.seleccionColores(20)));
29 29     }
30 30 }
31 31
32 32
```

```
...va Actividad711.java x MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x
Source History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package actividad.pkg7.pkg13;
6
7   import java.util.Arrays;
8   import java.util.Random;
9
10  /**
11   *
12   * @author Usuario
13   */
14  public class Colores {
15
16      String color[] = new String[0]; //Creo array color.
17
18      void addColor(String cl){
19
20          color = Arrays.copyOf(color, color.length+1);
21          color[color.length-1] = cl;
22      }
23
24      String[] seleccionColores(int cantidad){
25          String copiaArray[] = Arrays.copyOf(color, color.length);
26
27          desordenar(copiaArray);
28          int x = Math.min(cantidad, color.length);
29
30          return Arrays.copyOf(copiaArray, x);
31      }
32
33      private void desordenar(String copiaArray[]) {
34
35          Random aleatorio = new Random();
36
37          for (int i = 0; i < copiaArray.length; i++) {
38              int posAleatoria = aleatorio.nextInt(copiaArray.length);
39              String temp = copiaArray[i];
40              copiaArray[i] = copiaArray[posAleatoria];
41              copiaArray[posAleatoria] = temp;
42          }
43      }
44  }
45  //Autor: Derimán Tejera Fumero.
46
```

```
Output - Actividad 7.13 (run) x
run:
[Celeste, ColorPrueba2, Rosa, Marrón, Azul, ColorPrueba1, Amarillo]
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.14. Creo una clase que sea capaz de mostrar el importe de un cambio, por ejemplo, al realizar una compra, con el menor número de monedas y billetes posibles.

```
...va Actividad711.java x MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package actividad.pkg7.pkg14;
6
7   /**
8    *
9    * @author Usuario
10   */
11   public class Actividad714 {
12
13       /**
14        * @param args the command line arguments
15        */
16       public static void main(String[] args) {
17           Cambio c = new Cambio(12.30);
18           c.mostrarCambio();
19           c.setImporte(1234.56);
20           c.mostrarCambio();
21       }
22   }
23
```

```
...va Actividad916.java x Actividad617.java x Actividad918.java x Actividad920.java x ActividadDePruebaCapitulo7Clases.java x ClasePrueba1.java x Actividad711.java x Marcapagina.java x Actividad
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package actividad.pkg7.pkg14;
6
7   /**
8    *
9    * @author Usuario
10   */
11   public class Cambio {
12
13       final double [] monedas={500, 200, 100, 50, 20, 10, 5, 2, 1, 0.5, 0.20, 0.10, 0.05, 0.02, 0.01}; //Propiedad de la clase.
14       double importe;
15
16       public Cambio(double importe) {
17           this.importe = importe;
18       }
19
20       public void setImporte(double importe){
21           this.importe = importe;
22       }
23
24       public void mostrarCambio(){
25
26           System.out.println();
27           System.out.println("== NUEVO IMPORTE ==");
28           System.out.println("Importe: " + importe);
29           System.out.println();
30
31           double [] devolver={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
32
33           for(int i=0;i<monedas.length;i++)
34           {
35               if(importe>=monedas[i])
36               {
37                   devolver[i]=Math.floor(importe/monedas[i]);
38                   importe=importe-(devolver[i]*monedas[i]);
39               }
40           }
41
42           for(int i = 0; i<monedas.length; i++){
43
44               System.out.println("Valor [" + monedas[i] + "] * " + " = " + devolver[i] + " unidad/es");
45           }
46
47           System.out.println("=====");
48       }
49   }
50   //Autor: Derinán Tejera Fumero.
51
```

```
Output - Actividad 7.14 (run) X
run:
== NUEVO IMPORTE ==
Importe: 12.3

Valor [500.0] = 0.0 unidad/es
Valor [200.0] = 0.0 unidad/es
Valor [100.0] = 0.0 unidad/es
Valor [50.0] = 0.0 unidad/es
Valor [20.0] = 0.0 unidad/es
Valor [10.0] = 1.0 unidad/es
Valor [5.0] = 0.0 unidad/es
Valor [2.0] = 1.0 unidad/es
Valor [1.0] = 0.0 unidad/es
Valor [0.5] = 0.0 unidad/es
Valor [0.2] = 1.0 unidad/es
Valor [0.1] = 1.0 unidad/es
Valor [0.05] = 0.0 unidad/es
Valor [0.02] = 0.0 unidad/es
Valor [0.01] = 0.0 unidad/es
=====

== NUEVO IMPORTE ==
Importe: 1234.56

Valor [500.0] = 2.0 unidad/es
Valor [200.0] = 1.0 unidad/es
Valor [100.0] = 0.0 unidad/es
Valor [50.0] = 0.0 unidad/es
Valor [20.0] = 1.0 unidad/es
Valor [10.0] = 1.0 unidad/es
Valor [5.0] = 0.0 unidad/es
Valor [2.0] = 2.0 unidad/es
Valor [1.0] = 0.0 unidad/es
Valor [0.5] = 1.0 unidad/es
Valor [0.2] = 0.0 unidad/es
Valor [0.1] = 0.0 unidad/es
Valor [0.05] = 1.0 unidad/es
Valor [0.02] = 0.0 unidad/es
Valor [0.01] = 0.0 unidad/es
=====

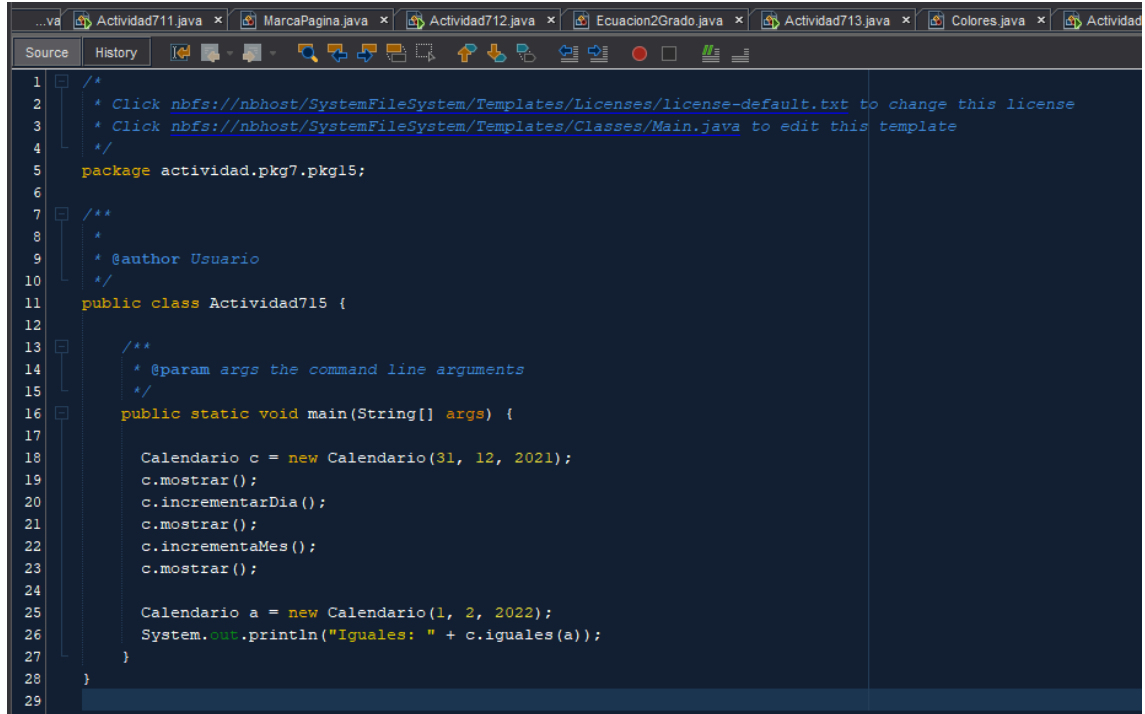
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.15. Diserta la dase Calendario que representa una fecha concreta (arto, mes y día). La clase debe disponer de los métodos:

- Calendario(int año, int mes, int dia): que crea un objeto con los datos pasados como parámetros, siempre y cuando, la fecha que representen sea correcta.
- void incrementarDia (): que incrementa en un día la fecha del calendario.
- void incrementarMesO: que incrementa en un mes la fecha del calendario.
- void incrementarlo (int cantidad): que incrementa la fecha del calendario en el número de artos especificados. Ten en cuenta que el arto 0 no existió.
- void mostrar (): muestra la fecha por consola.

■ boolean iguales {Calendario otraFecha): que determina si la fecha invocante y la que se pasa como parámetro son iguales o distintas.

Por simplicidad, solo tendremos en consideración que existen meses con distinto número de días, pero no tendremos en cuenta los años bisiestos.



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package actividad.pkg7.pkg15;
6
7   /**
8    *
9    * @author Usuario
10   */
11   public class Actividad715 {
12
13       /**
14        * @param args the command line arguments
15        */
16       public static void main(String[] args) {
17
18           Calendario c = new Calendario(31, 12, 2021);
19           c.mostrar();
20           c.incrementarDia();
21           c.mostrar();
22           c.incrementaMes();
23           c.mostrar();
24
25           Calendario a = new Calendario(1, 2, 2022);
26           System.out.println("Iguales: " + c.iguales(a));
27       }
28   }
29
```

```

58     mes++;
59     dia=1;
60 }
61
62     if (dia==28 && (mes==2)) {
63         mes++;
64         dia=1;
65     }
66
67     if (dia==31 && (mes==12)) {
68         mes=1;
69         dia=1;
70         año++;
71     }
72 }
73
74 public void incrementaMes() {
75
76     if (mes==12) {
77         mes=1;
78         dia=1;
79         año++;
80     }
81     else{
82         mes++;
83     }
84 }
85
86 public void incrementarAño(int cantidad){
87
88     System.out.println("Introduce el valor para año: ");
89
90     do{
91         System.out.print("Introduce el año: ");
92         año = sc.nextInt();
93         if (año <= 0){
94             System.out.println("El año no es válido, recuerda que tiene que ser mayor que 0.");
95         }
96     } while (año <= 0);
97 }
98
99 public void mostrar(){
100     System.out.println();
101     System.out.println("Mostrando fecha...");
102     System.out.println(dia + "/" + mes + "/" + año);
103     System.out.println();
104 }
105
106 public boolean iguales(Calendar otraFecha){
107     return (otraFecha.getDia() == dia && otraFecha.getMes() == mes && otraFecha.getAño() == año);
108 }
109
110 public int getDia(){
111     return dia;
112 }
113
114 public int getMes(){
115     return mes;
116 }
117
118 public int getAño(){
119     return año;
120 }
121 }
122 //Autor: Derimán Tejeras Fumero.
123

```

```

Output - Actividad 7.15 (run) X
run:

Mostrando fecha...
31/12/2021

Mostrando fecha...
1/1/2022

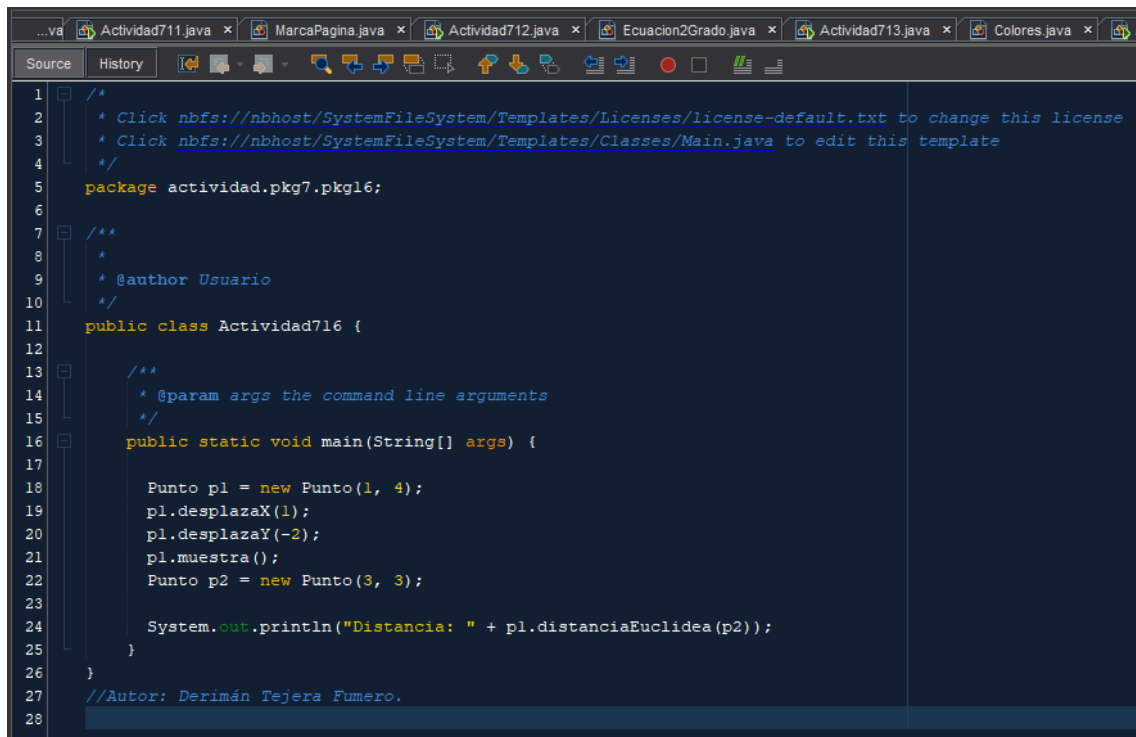
Mostrando fecha...
1/2/2022

Iguales: true
BUILD SUCCESSFUL (total time: 0 seconds)

```

7.16. Escribe la clase Punto que representa un punto en el plano (con un componente x y un componente y), con los métodos:

- `Punto(double x, double y)`: construye un objeto con los datos pasados como parámetros.
  - `void desplazaX {double dx}`: incrementa el componente x en la cantidad dx.
  - `void desplazaY {double dy}`: incrementa el componente y en la cantidad dy.
  - `void desplaza (double dx, double dy)`: desplaza ambos componentes según las cantidades dx (en el eje x) y dy (en el componente y).
- a double `distanciaEuclidea (Punto otro)`: calcula y devuelve la distancia euclidea entre el punto invocante y el punto otro.
- `void muestra ()`: muestra por consola la información relativa al punto.



```

1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package actividad.pkg7.pkg16;
6
7   /**
8    *
9    * @author Usuario
10   */
11  public class Actividad716 {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17
18          Punto p1 = new Punto(1, 4);
19          p1.desplazaX(1);
20          p1.desplazaY(-2);
21          p1.muestra();
22          Punto p2 = new Punto(3, 3);
23
24          System.out.println("Distancia: " + p1.distanciaEuclidea(p2));
25      }
26  }
27  //Autor: Derimán Tejera Fumero.
28

```

```
...va Actividad711.java x MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x A
Source History
1 1
2 2
3 3
4 4
5 package actividad.pkg7.pkg16;
6
7 /**
8  *
9  * @author Usuario
10 */
11 public class Punto {
12
13     double x;
14     double y;
15
16     public Punto(double x, double y) {
17         this.x = x;
18         this.y = y;
19     }
20
21     void desplazaX(double dx){
22         x = x+dx;
23     }
24
25     void desplazaY(double dy){
26         y = y+dy;
27     }
28
29     void desplaza(double dx, double dy){
30         x = x+dx;
31         y = y+dy;
32     }
33
34     double distanciaEuclidea(Punto otro){
35         return Math.sqrt(Math.pow((otro.getX() - x), 2) + Math.pow((otro.getY() - y), 2));
36     }
37
38     void muestra(){
39         System.out.println();
40         System.out.println("Mostrando datos del punto...");
41         System.out.println("(" + x + "," + y + ")");
42         System.out.println();
43     }
44
45     public double getX(){
46         return x;
47     }
48
49     public double getY(){
50         return y;
51     }
52 }
53 //Autor: Derimán Tejera Fumero.
54
```

```
Output - Actividad 7.16 (run) x
run:
Mostrando datos del punto...
(2.0,2.0)
Distancia: 1.4142135623730951
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.17. El cifrado César es una forma sencilla de modificar un texto para que no sea entencible a quienes no conocen el código. Este cifrado consiste en modificar cada letra de un texto por otra que se encuentra en el alfabeto n posiciones detrás.



Por ejemplo, para un valor de n igual a 3. la letra a se codifica con la d. y la letra q se codifica con la x. En el caso de que una letra exceda a la z. seguiremos de forma circular utilizando la a. Solo se cifrarán las letras, mayúsculas o minúsculas.

Realiza una clase que. mediante un método estático, devuelva cifrado el texto que se le pasa con un paso de n letras.

```
...va Actividad711.java x MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package actividad.pkg7.pkg17;
6
7   /**
8    *
9    * @author Usuario
10   */
11  public class Actividad717 {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          System.out.println(Cifrado.cesar("abcdefghijklmnpqrstuvwxyz", 3));
18          System.out.println(Cifrado.cesar("abcdef ... vwxyz", 3));
19          System.out.println(Cifrado.cesar("Hola mundo", 5));
20      }
21  }
22
23  }
```

```
...va Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x Actividad715.java x Calendario.java x
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package actividad.pkg7.pkg17;
6
7   /**
8    *
9    * @author Usuario
10   */
11  public class Cifrado {
12
13      private static final char[] Alfabeto = "abcdefghijklmnopqrstuvwxyz".toCharArray();
14
15
16
17      public static String cesar(String texto, int n) {
18          char cifradoTexto[] = new char[texto.length()];
19
20          for (int i = 0; i < texto.length(); i++) {
21              char charActual = texto.charAt(i);
22
23              if (Character.isLetter(charActual)) {
24                  int indiceAlfabeto;
25                  if (Character.isLowerCase(charActual)) {
26                      indiceAlfabeto = charActual - 'a';
27                  } else {
28                      indiceAlfabeto = charActual - 'A';
29                  }
30                  int indiceAlfabetoNuevo = (indiceAlfabeto + n) % Alfabeto.length;
31                  char charNuevo = Alfabeto[indiceAlfabetoNuevo];
32                  if (Character.isUpperCase(charActual)) {
33                      charNuevo = Character.toUpperCase(charNuevo);
34                  }
35                  cifradoTexto[i] = charNuevo;
36              } else {
37                  cifradoTexto[i] = charActual;
38              }
39          }
40
41          return new String(cifradoTexto);
42      }
43  }
44  //Autor: Derimán Tejera Fumero.
45  }
```

```
Output - Actividad 7.17 (run) X
run:
defghijklmnopqrstuvwxyzabc
defghi ... yzabc
Mtqf rzsit
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.18. Una cola es otra estructura dinámica como la pila, donde tos elementos, en vez de apilarse y desapilarse, se encolan y desencolan. La diferencia con las pilas es que se desencola el primer elemento encolado, ya que así es como funcionan las colas del autobús o del cine. El primero que llega es el primero que sale de la cola (vamos a suponer que nadie se cuela). Por tanto, los elementos se encolan y desencolan en extremos opuestos de la estructura, llamados primero (el que está primero y será el próximo en abandonar la cola) y último (el que llegó último). Implemento la clase Cola donde los elementos Integer encolados se guardan en una tabla.

```
...va MarcaPagina.java x Actividad712.java x Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x Actividad715.java
Source History
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package actividad.pkg7.pkg18;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class Actividad718 {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17
18          Cola c = new Cola(10);
19          for (int i = 1; i <= 10; i++) {
20              c.encola(i);
21          }
22
23          System.out.println("Primero: " + c.primer());
24          System.out.println("Vacía: " + c.vacia());
25          while(!c.vacia()) {
26              System.out.println(c.desencola());
27          }
28      }
29  }
30
```

```
..va  MarcaPagina.java x  Actividad712.java x  Ecuacion2Grado.java x  Actividad713.java x  Colores.java x  Actividad714.java x  Cambio.java x
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package actividad.pkg18;
6
7   /**
8    *
9    * @author Usuario
10   */
11   public class Cola {
12
13       int cola[];
14       int primero;
15       int ultimo;
16       int tamaño;
17
18       public Cola(int tamaño) {
19           this.cola = new int[tamaño];
20           primero = -1;
21           ultimo = -1;
22           this.tamaño = tamaño;
23       }
24
25       public void encola(int valor) {
26           if (ultimo == tamaño - 1) {
27               System.out.println("La cola está llena");
28           } else if (primero == -1 && ultimo == -1) {
29               primero++;
30               ultimo++;
31               cola[ultimo] = valor;
32           } else {
33               ultimo++;
34               cola[ultimo] = valor;
35           }
36       }
37
38       public int desencola() {
39           int valor = -1;
40           if (primero == -1 && ultimo == -1) {
41               System.out.println("La cola está vacía");
42           } else if (primero == ultimo) {
43               valor = cola[primero];
44               primero = -1;
45               ultimo = -1;
46           } else {
47               valor = cola[primero];
48               primero++;
49           }
50           return valor;
51       }
52
53       public int primero() {
54           if (primero == -1 && ultimo == -1) {
55               System.out.println("La cola está vacía");
56               return -1;
57           }
58           return cola[primero];
59       }
60
61       public boolean vacia() {
62           return (primero == -1 && ultimo == -1);
63       }
64   }
65   //Autor: Derimán Tejera Fumero.
66
```

```
Output - Actividad 7.18 (run) X
run:
Primero: 1
Vacía: false
1
2
3
4
5
6
7
8
9
10
BUILD SUCCESSFUL (total time: 0 seconds)
```

7.19. Implemento la clase Pila para números Integer. usando directamente una tabla para guardar los elementos apilados.

```
...va Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x Actividad715.java x Calendario.java x Actividad716.java x
Source History
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package actividad.pkg7.pkg19;
6
7  /**
8   *
9   * @author Usuario
10  */
11  public class Actividad719 {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          Pila p = new Pila();
18          for (int i = 1; i <= 10; i++) {
19              p.apila(i);
20          }
21
22          System.out.println("Cima: " + p.cima());
23          while(!p.vacia()) {
24              System.out.println(p.desapila());
25          }
26      }
27  }
28
```

```
...va x Ecuacion2Grado.java x Actividad713.java x Colores.java x Actividad714.java x Cambio.java x Actividad715.java x Calendario.java x Acti
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package actividad.pkg7.pkg19;
6
7   /**
8    *
9    * @author Usuario
10   */
11   public class Pila {
12
13       private int pila[];
14       private int cima;
15       private int tamaño;
16
17       public Pila() {
18           tamaño = 10;
19           cima = -1;
20           pila = new int[tamaño];
21       }
22
23       public void apila(int valor) {
24           if (cima == tamaño - 1) {
25               System.out.println("La pila está llena");
26           } else {
27               cima++;
28               pila[cima] = valor;
29           }
30       }
31
32       public int desapila() {
33           if (cima == -1) {
34               System.out.println("La pila está vacía");
35               return -1;
36           } else {
37               int valor = pila[cima];
38               cima--;
39               return valor;
40           }
41       }
42
43       public int cima() {
44           if (cima == -1) {
45               System.out.println("La pila está vacía");
46               return -1;
47           } else {
48               return pila[cima];
49           }
50       }
51
52       public boolean vacia() {
53           return cima == -1;
54       }
55   }
56   //Autor: Derimán Tejera Fumero.
```

Output - Actividad 7.19 (run) X



run:



Cima: 10



10



9

8

7

6

5

4

3

2

1

BUILD SUCCESSFUL (total time: 0 seconds)